تم تحميل وعرض المادة من :

موقع واجباتي

www.wajibati.net

موقع واجباتي منصة تعليمية تساهم بنشر
حل المناهج الدراسية بشكل متميز لترتقي بمجال التعليم
على الإنترنت ويستطيع الطلاب تصفح حلول الكتب مباشرة
لجميع المراحل التعليمية المختلفة

حمل التطبيق من هنا

المملكة العربية السعودية

وزارة التعليم
Ministry of Education

قررت وزارة التعليم تدريس
هذا الكتاب وطبعه على نفقتها

هندسة البرمجيات

التعليم الثانوي - نظام المسارات

السنة الثالثة

طبعة 1445-2023

يوزع مجاناً ولا يباع
وزارة التعليم
Ministry of Education
2023 - 1445

ⓒ وزارة التعليم، ١٤٤٤ هـ

فهرسة مكتبة الملك فهد الوطنية أثناء النشر
وزارة التعليم
هندسة البرمجيات. / وزارة التعليم.- الرياض، ١٤٤٤ هـ
٢٢٤ ص ؛ ٢١ x ٢٥,٥ سم
ردمك : ٣-٤٤٩-٥١١-٦٠٣-٩٧٨
١ - البرمجيات (حاسبات إلكترونية) أ . العنوان
ديوي ٠٠٥,١٣٣ ٩٢٧٨ / ١٤٤٤

رقم الإيداع : ٩٢٧٨ / ١٤٤٤
ردمك :٣-٤٤٩-٥١١-٦٠٣-٩٧٨

مواد إثرائية وداعمة على "منصة عين الإثرائية"

ien.edu.sa

أعزاءنا المعلمين والمعلمات، والطلاب والطالبات، وأولياء الأمور، وكل مهتم بالتربية والتعليم:
يسعدنا تواصلكم؛ لتطوير الكتاب المدرسي، ومقترحاتكم محل اهتمامنا.

fb.ien.edu.sa

أخي المعلم/أختي المعلمة، أخي المشرف التربوي/أختي المشرفة التربوية:
نقدر لك مشاركتك التي ستسهم في تطوير الكتب المدرسية الجديدة، وسيكون لها الأثر الملموس في دعم العملية التعليمية، وتجويد ما يقدم لأبنائنا وبناتنا الطلبة.

fb.ien.edu.sa/BE

وزارة التعليم
Ministry of Education
2023 - 1445

الناشر: شركة تطوير للخدمات التعليمية

تم النشر بموجب اتفاقية خاصة بين شركة Binary Logic SA وشركة تطوير للخدمات التعليمية (عقد رقم 2023/0003) للاستخدام في المملكة العربية السعودية

حقوق النشر © Binary Logic SA 2023

جميع الحقوق محفوظة. لا يجوز نسخ أي جزء من هذا المنشور أو تخزينه في أنظمة استرجاع البيانات أو نقله بأي شكل أو بأي وسيلة إلكترونية أو ميكانيكية أو بالنسخ الضوئي أو التسجيل أو غير ذلك دون إذن كتابي من الناشرين.

يُرجى ملاحظة ما يلي: يحتوي هذا الكتاب على روابط إلى مواقع إلكترونية لا تُدار من قبل شركة Binary Logic. ورغم أنَّ شركة Binary Logic تبذل قصارى جهدها لضمان دقة هذه الروابط وحداثتها وملاءمتها، إلا أنها لا تتحمل المسؤولية عن محتوى أي مواقع إلكترونية خارجية.

إشعار بالعلامات التجارية: أسماء المنتجات أو الشركات المذكورة هنا قد تكون علامات تجارية أو علامات تجارية مُسجَّلة وتُستخدم فقط بغرض التعريف والتوضيح وليس هناك أي نية لانتهاك الحقوق. تنفي شركة Binary Logic وجود أي ارتباط أو رعاية أو تأييد من جانب مالكي العلامات التجارية المعنيين. تُعد Excel علامة تجارية مُسجَّلة لشركة Microsoft Corporation. تُعد Tinkercad علامة تجارية مُسجَّلة لشركة Autodesk Inc. تُعد "Python" وشعارات Python علامات تجارية مسجلة لشركة Python Software Foundation. تُعد Jupyter علامة تجارية مُسجَّلة لشركة Project Jupyter. تُعد PyCharm علامة تجارية مُسجَّلة لشركة JetBrains s.r.o. تُعد Multisim Live علامة تجارية مُسجَّلة لشركة National Instruments Corporation. تُعد CupCarbon علامة تجارية مُسجَّلة لشركة CupCarbon. تُعد Arduino علامة تجارية مُسجَّلة لشركة Arduino SA. تُعد Micro:bit علامة تجارية مُسجَّلة لشركة Micro:bit Educational Foundation.
ولا ترعى الشركات أو المنظمات المذكورة أعلاه هذا الكتاب أو تصرح به أو تصادق عليه.

حاول الناشر جاهدا تتبع ملاك الحقوق الفكرية كافة، وإذا كان قد سقط اسم أيٍّ منهم سهوًا فسيكون من دواعي سرور الناشر اتخاذ التدابير اللازمة في أقرب فرصة.

binarylogic

وزارة التعليم
Ministry of Education
2023 - 1445

مقدمة

إن تقدم الدول وتطورها يقاس بمدى قدرتها على الاستثمار في التعليم، ومدى استجابة نظامها التعليمي لمتطلبات العصر ومتغيراته. وحرصًا من وزارة التعليم على ديمومة تطوير أنظمتها التعليمية، واستجابة لرؤية المملكة العربية السعودية 2030 فقد بادرت الوزارة إلى اعتماد نظام «مسارات التعليم الثانوي» بهدف إحداث تغيير فاعل وشامل في المرحلة الثانوية.

إن نظام مسارات التعليم الثانوي يقدم أنموذجًا تعليميًا متميزًا وحديثًا للتعليم الثانوي بالمملكة العربية السعودية يسهم بكفاءة في:

- تعزيز قيم الانتماء لوطننا المملكة العربية السعودية، والولاء لقيادته الرشيدة حفظهم الله، انطلاقًا من عقيدة صافية مستندة على التعاليم الإسلامية السمحة.
- تعزيز قيم المواطنة من خلال التركيز عليها في المواد الدراسية والأنشطة، اتساقًا مع مطالب التنمية المستدامة، والخطط التنموية في المملكة العربية السعودية التي تؤكد على ترسيخ ثنائية القيم والهوية، والقائمة على تعاليم الإسلام والوسطية.
- تأهيل الطلبة بما يتوافق مع التخصصات المستقبلية في الجامعات والكليات أو المهن المطلوبة؛ لضمان اتساق مخرجات التعليم مع متطلبات سوق العمل.
- تمكين الطلبة من متابعة التعليم في المسار المفضل لديهم في مراحل مبكرة، وفق ميولهم وقدراتهم.
- تمكين الطلبة من الالتحاق بالتخصصات العلمية والإدارية النوعية المرتبطة بسوق العمل، ووظائف المستقبل.
- دمج الطلبة في بيئة تعليمية ممتعة ومحفزة داخل المدرسة قائمة على فلسفة بنائية، وممارسات تطبيقية ضمن مناخ تعليمي نشط.
- نقل الطلبة عبر رحلة تعليمية متكاملة بدءًا من المرحلة الابتدائية حتى نهاية المرحلة الثانوية، وتُسهّل عملية انتقالهم إلى مرحلة ما بعد التعليم العام.
- تزويد الطلبة بالمهارات التقنية والشخصية التي تساعدهم على التعامل مع الحياة، والتجاوب مع متطلبات المرحلة.
- توسيع الفرص أمام الطلبة الخريجين عبر خيارات متنوعة إضافة إلى الجامعات مثل: الحصول على شهادات مهنية، والالتحاق بالكليات التطبيقية، والحصول على دبلومات وظيفية.

ويتكون نظام المسارات من تسعة فصول دراسية تُدرّس في ثلاث سنوات، تتضمن سنة أولى مشتركة يتلقى فيها الطلبة الدروس في مجالات علمية وإنسانية متنوعة، تليها سنتان تخصصيتان، يُسكّن الطلبة بها في مسار عام وأربعة مسارات تخصصية تتسق مع ميولهم وقدراتهم، وهي: المسار الشرعي، مسار إدارة الأعمال، مسار علوم الحاسب والهندسة، مسار الصحة والحياة، وهو ما يجعل هذا النظام هو الأفضل للطلبة من حيث:

- وجود مواد دراسية جديدة تتوافق مع متطلبات الثورة الصناعية الرابعة والخطط التنموية، ورؤية المملكة 2030، تهدف لتنمية مهارات التفكير العليا وحل المشكلات، والمهارات البحثية.
- برامج المجال الاختياري التي تتسق مع احتياجات سوق العمل وميول الطلبة، حيث يُمكّن الطلبة من الالتحاق بمجال اختياري محدد وفق مصفوفة مهارات وظيفية محددة.
- مقياس ميول يضمن تحقيق كفاءة الطلبة وفاعليتهم، ويساعدهم في تحديد اتجاهاتهم وميولهم، وكشف مكامن القوة لديهم، مما يعزز من فرص نجاحهم في المستقبل.
- العمل التطوعي المصمم للطلبة خصيصًا بما يتسق مع فلسفة النشاط في المدارس، ويعد أحد متطلبات التخرج؛ مما يساعد على تعزيز القيم الإنسانية، وبناء المجتمع وتنميته وتماسكه.
- التجسير الذي يمكن الطلبة من الانتقال من مسار إلى آخر وفق آليات محددة.
- حصص الإتقان التي يتم من خلالها تطوير المهارات وتحسين المستوى التحصيلي، من خلال تقديم حصص إتقان إثرائية وعلاجية.

- خيارات التعليم المدمج، والتعلم عن بعد، والذي بُني في نظام المسارات على أسس من المرونة، والملاءمة والتفاعل والفعالية.
- مشروع التخرج الذي يساعد الطلبة على دمج الخبرات النظرية مع الممارسات التطبيقية.
- شهادات مهنية ومهارية تمنح للطلبة بعد إنجازهم مهامَّ محددة، واختبارات معينة بالشراكة مع جهات تخصصية.

وبالتالي فإن مسار علوم الحاسب والهندسة كأحد المسارات المستحدثة في المرحلة الثانوية يسهم في تحقيق أفضل الممارسات عبر الاستثمار في رأس المال البشري، وتحويل الطالب إلى فرد مشارك ومنتج للعلوم والمعارف، مع إكسابه المهارات والخبرات اللازمة لاستكمال دراسته في تخصصات تتناسب مع ميوله وقدراته أو الالتحاق بسوق العمل.

وتعد مادة هندسة البرمجيات أحد المواد الرئيسة في مسار علوم الحاسب والهندسة. وتساعد المادة على تعلم أساسيات هندسة البرمجيات من خلال الانخراط والمشاركة في اكتشاف مجموعة متنوعة من الموضوعات في هذا المجال. ويقدِّم هذا الكتاب لمحة عامة عن دورة حياة تطوير البرمجيات ويناقش المفاهيم الرئيسة للتفاعل بين الإنسان والحاسب والنماذج الأولية، بالإضافة إلى ذلك يتعلم الطالب تصميم تطبيق الهاتف المحمول وتطويره مع أخذ ميزة قابلية الوصول في الاعتبار. كما تشتمل هذه المادة على مشاريع وتمارين تطبيقية لما يتعلمه الطالب؛ لحل مشاكل واقعية تحاكي مستوياته المعرفية، بتوجيه وإشراف من المعلم.

ويتميز كتاب هندسة البرمجيات بأساليب حديثة، تتوافر فيه عناصر الجذب والتشويق، والتي تجعل الطلبة يقبلون على تعلمه والتفاعل معه، من خلال ما يقدمه من تدريبات وأنشطة متنوعة، كما يؤكد هذا الكتاب على جوانب مهمة في تعليم هندسة البرمجيات وتعلمه، تتمثل في:

- الترابط الوثيق بين المحتويات والمواقف والمشكلات الحياتية.
- تنوع طرائق عرض المحتوى بصورة جذابة ومشوقة.
- إبراز دور المتعلم في عمليات التعليم والتعلم.
- الاهتمام بترابط محتوياته مما يجعل منه كلًّا متكاملًا.
- الاهتمام بتوظيف التقنيات المناسبة في المواقف المختلفة.
- الاهتمام بتوظيف أساليب متنوعة في تقويم الطلبة بما يتناسب مع الفروق الفردية بينهم.

ولمواكبة التطورات العالمية في هذا المجال، فإن كتاب مادة هندسة البرمجيات سوف يوفر للمعلم مجموعة متكاملة من المواد التعليمية المتنوعة التي تراعي الفروق الفردية بين الطلبة، بالإضافة إلى البرمجيات والمواقع التعليمية، التي توفر للطلبة فرصة توظيف التقنيات الحديثة والتواصل المبني على الممارسة؛ مما يؤكد دوره في عملية التعليم والتعلم.

ونحن إذ نقدم هذا الكتاب لأعزائنا الطلبة، نأمل أن يستحوذ على اهتمامهم، ويُلبي متطلباتهم، ويجعل تعلّمهم لهذه المادة أكثر متعة وفائدة.

والله ولي التوفيق

وزارة التعليم
Ministry of Education
2023 - 1445

وزارة التعليم
Ministry of Education
2023 - 1445

الفهرس

1. هندسة البرمجيات 8

الدرس الأول

مبادئ هندسة البرمجيات 9

تمرينات 21

الدرس الثاني

لغات البرمجة ومُعالجاتها 28

تمرينات 39

الدرس الثالث

أدوات تطوير البرمجيّات 42

تمرينات 55

المشروع 60

2. النمذجة الأولية 62

الدرس الأول

التحليل 63

تمرينات 79

الدرس الثاني

التفاعل بين الإنسان والحاسب 84

تمرينات 91

الدرس الثالث

إنشاء نموذج أولي 95

تمرينات 109

المشروع 112

3. تطوير التطبيقات باستخدام مخترع التطبيقات 114

الدرس الأول

مقدمة إلى مخترع التطبيقات MIT 115

تمرينات 133

الدرس الثاني

إضافة المزيد من العناصر إلى التطبيق ... 135

تمرينات 151

الدرس الثالث

برمجة تطبيق الهاتف المحمول 153

تمرينات 180

المشروع 182

4. قابلية الوصول إلى البرمجيات والتضمين الرقمي 184

الدرس الأول

اختبار التطبيقات ونشرها 185

تمرينات 190

الدرس الثاني

التضمين الرقمي 192

تمرينات 200

الدرس الثالث

ميزات قابلية الوصول في التطبيق 203

تمرينات 219

المشروع 220

وزارة التعليم
Ministry of Education
2023 - 1445

1. هندسة البرمجيات

سيتعرّف الطالب في هذه الوحدة على منهجيات تطوير البرمجيات الأكثر شيوعًا، وعلى المقصود بدورة حياة النظام، والمراحل المختلفة لهذه الدورة. وسيتعرّف أيضًا على أهمية تحويل لغات البرمجة عالية المستوى إلى تعليمات برمجية قابلة للتنفيذ بلغة الآلة، وعلى البرامج المستخدمة للقيام بذلك.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يُميِّز منهجيات تطوير البرمجيات الأكثر شيوعًا.
- يُعرِّف المراحل المختلفة لدورة حياة تطوير البرمجيات.
- يُنشئ دورة حياة تطوير برمجيات لأحد التطبيقات.
- يصف مرحلة التحليل لدورة حياة تطوير البرمجيات.
- يُعرِّف الطرق المختلفة لجمع المتطلّبات للأنظمة.
- يُصنِّف لغات البرمجة وخصائصها.
- يصف أهمية مُترجِم ومُفسِّر لغة البرمجة.
- يُصنِّف أدوات تطوير البرمجيات المختلفة.
- يُعرِّف محرر البرمجة ومزاياه وتحديات استخدامه.
- يُعرِّف بيئة التطوير المتكاملة ويشرح مزاياها وتحديات استخدامها.
- يستخدِم أدوات تطوير البرمجيات لتطوير حلول برمجية مختلفة.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

مبادئ هندسة البرمجيات

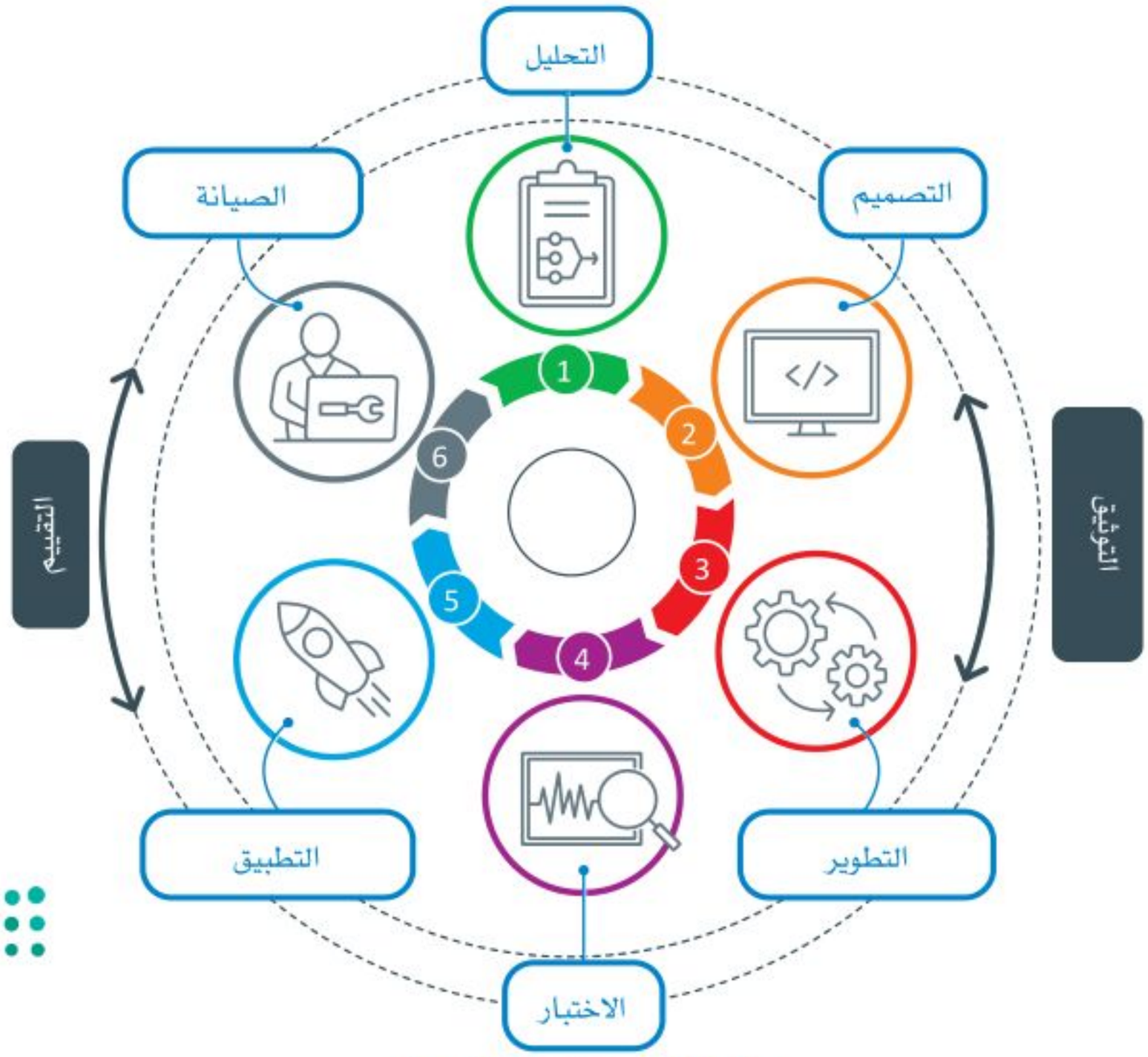
تُعدُّ **هندسة البرمجيات** (Software Engineering) أحد فروع علم الحاسب التي تختص بتطوير أنظمة البرمجيات وصيانتها، وتتضمن هندسة البرمجيات تطبيق المبادئ والأسس الهندسية لتصميم البرامج، وتطويرها، واختبارها، وصيانتها. تهدف هندسة البرمجيات إلى إنتاج برمجيات موثوقة وفعالة وذات كفاءة عالية تلبي متطلّبات المُستخدِمين والأهداف التي تم تطويرها لتحقيقها، ويتم تحقيق ذلك من خلال عملياتٍ وأدوات وتقنيات منهجية محدَّدة. تتضمن هندسة البرمجيات أيضًا إدارة عملية تطوير البرمجيات بما فيها عملية التخطيط للمشاريع والتقييم، وإدارة المخاطر، وضمان الجودة.

دورة حياة تطوير البرمجيات (SDLC) Software Development Lifecycle

تَصِفُ **دورة حياة تطوير البرمجيات** (SDLC) كيفية تنظيم عمليات إنتاج أنظمة المعلومات في شتى المجالات، ولا تقتصر أهداف دورة حياة تطوير البرمجيات على تحسين المنتج النهائي (نظام المعلومات)، بل تشمل أيضًا إدارة عمليات الإنتاج والتطوير وتنظيم وترشيد استخدام الموارد خلال هذه العمليات. سيتم في هذا الدرس مناقشة مراحل دورة حياة تطوير البرمجيات ضمن سياق تطوير أنظمة **تقنية المعلومات والاتصالات** (Information and Communication Technology –ICT).

تتكون دورة حياة تطوير البرمجيات من عدة مراحل متسلسلة كما يتضح في الشكل أدناه.

ستستكشف جميع هذه المراحل من خلال مثال تطبيقي برمجي لنظام مصرفي.



شكل 1.1: دورة حياة تطوير البرمجيات

التحليل Analysis

تتمثل الخطوة الأولى في دورة حياة تطوير البرمجيات في تحديد المشكلة التي يجب حلها، ثم تحديد متطلّبات الحل بأكبر قَدر مُمكن من الدقة من خلال عمليات جمع المتطلّبات. تتضمن عملية جمع المتطلّبات تحليل احتياجات وتوقُّعات أصحاب المصلحة (عملاء أو مُستخدِمي النظام البرمجي)، وتحديدها، والتحقق من صحتها، وإدارتها. يتضمن التحليل كذلك فهم مجال المشكلة وتحديد المتطلّبات الوظيفيّة وغير الوظيفيّة للبرنامج، ويتضح الاختلاف بين المتطلّبات الوظيفيّة وغير الوظيفيّة فيما يلي:

- **المتطلّبات الوظيفيّة** (Functional Requirements): هي القدرات أو الميزات المُحدَّدة التي يجب أن يمتلكها البرنامج لتلبية احتياجات أصحاب المصلحة. على سبيل المثال، يُعدُّ توفير وظيفة البحث عن المنتجات وإضافتها إلى عربة التسوق وإتمام عملية الدفع بأمان لمستخدمي موقع تسوّق إلكتروني وتمكينهم من ذلك، من المتطلّبات الوظيفيّة الشائعة لذلك الموقع.
- **المتطلّبات غير الوظيفيّة** (Non-functional Requirements): تتمثل في القيود وخصائص الجودة التي يجب أن يفي بها البرنامج ليكون مقبولًا لدى أصحاب المصلحة. ومن الأمثلة المميّزة على المتطلّبات غير الوظيفيّة للبرامج: توفير متطلّبات الأداء، مثل زمن الاستجابة، والإنتاجية، وتوفير متطلّبات الأمان، مثل: خصوصية البيانات، والمصادقة، ومتطلّبات قابلية الاستخدام مثل: وجود واجهة سهلة الاستخدام.

يجب أخذ جميع الموارد بالاعتبار خلال عملية التحليل، بما فيها الموارد البشرية والمادية، والتكلفة والميزانية والوقت المتاح للمشروع، وكل ما يتعلق بالمشروع، ويجب تحديد جميع وظائف النظام الجديد المطلوبة بالتفصيل مع الإشارة إلى القيود والعقبات التي تتعلق بهذا النظام.

تتضمن عملية التحليل تحديد المُستخدِمين واحتياجاتهم ومتطلّباتهم، وتُعدُّ الأدوات التالية الأكثر استخدامًا لجمع البيانات المطلوبة:

- الاستِبانات.
- المقابلات.
- الملاحظة.

يسعى أحد المصارف لإنشاء نظام إلكتروني لتقديم الخدمات المصرفية عبر الإنترنت، وستشمل مرحلة التحليل في هذا المشروع جمع المعلومات الخاصة بمتطلّبات الإدارة والعملاء من أجل تحديد الخدمات المصرفية التي يجب أن يتم توفيرها إلكترونيًا عبر الإنترنت، وتصميم واجهات المُستخدِم المطلوبة، وتحديد متطلّبات الأمان والأذونات الرقمية المخصصة لموظفي المصرف والعملاء في النظام.

التصميم Design

مرحلة التصميم هي المرحلة الثانية من دورة حياة تطوير البرمجيات، حيث يوفر مُحللو الأنظمة في هذه المرحلة الخبرات والمهارات اللازمة لتخطيط وتصميم حل المشكلة المحدَّدة.

تحدد مرحلة التصميم أنواع البيانات المطلوبة والواجهات المختلفة للنظام، وبشكل أكثر تحديدًا، تتضمن مرحلة التصميم ما يلي:

1. تحديد تدفق البيانات والمعلومات في جميع أجزاء النظام الجديد.
2. تحديد البيانات الرئيسة المراد معالجتها، والتي يتم تحديد هياكل البيانات المُستخدَمة من قِبَلْ النظام بناءً عليها.

3. تحديد مكان تخزين البيانات وطريقة تخزينها ومعايير الأمان الخاصة بها.
4. تصميم التقارير وغيرها من مُخرَجات البيانات والمعلومات.
5. تصميم واجهات المُستخدِم وتحديد وظائف كافة العناصر الموجودة فيها.
6. تصميم واجهات تكاملية لتبادل البيانات مع أنظمة المعلومات والاتصالات الأخرى.
7. تحديد طريقة اختبار النظام، والبيانات المستخدمة للاختبار، وكيفية استخدامها في ضمان الجودة.

خلال عملية تصميم النظام المصرفي الإلكتروني، يجب تحديد مسارات تدفق البيانات بين كل من النظام والمُستخدِمين وقواعد البيانات المختلفة وأنظمة المعلومات المتكاملة الأخرى. تعتمد هذه المسارات جميعها على أنواع البيانات المختلفة التي يجب تخزينها وحمايتها ونقلها. يجب تحديد متطلّبات النظام لعملية إدخال البيانات وإخراجها، كما يجب تصميم واجهات مُستخدِم خاصة بالموظفين والعملاء، وذلك بالإضافة إلى واجهات لتبادل البيانات والمعلومات المالية مع المؤسسات الأخرى. وفي الختام يجب تحديد الاختبارات التي يجب إجراؤها لضمان عمل النظام كما هو متوقع.

التطوير والاختبار Development and Testing

المرحلتان التاليتان في دورة حياة تطوير البرمجيات هما التطوير والاختبار. بعد إجراء عمليتي التحليل والتصميم بشكلٍ مفصّل، يتعين على المبرمجين وأولئك المسؤولون عن اختبار النظام تحويل المتطلّبات والمواصفات إلى مقاطع برمجية باستخدام البرامج المناسبة ولغات البرمجة المختلفة. لا يمكن الفصل بين مرحلتي التطوير والاختبار، حيث يجب اختبار النظام بشكل شامل أثناء التطوير وبعده لضمان معالجة جميع المشاكل، وللتأكد بشكلٍ مطلق بأن النظام يلبي متطلّبات المُستخدِمين، وتتضمن عناصر النظام التي تتطلب اختبارًا مُستقلًا ما يلي:

1. اختبار صحة البيانات المدخلة:

قد يتسبب إدخال بيانات غير صحيحة في حدوث مشاكل داخل النظام المصرفي، ولذلك من المهم اختبار صحة البيانات المُدخَلة. في مثال الخدمات المصرفية الإلكترونية، يتطلب ذلك تطوير قواعد الأمان لاستقبال كلمات المرور من المُستخدِمين، وتحديد عدد المحاولات غير الصحيحة المسموح بها عند إدخال كلمات المرور، وقواعد التحقق من الأرقام المدخلة في النظام كأرقام الهواتف المحمولة وأرقام الهوية الشخصية، وكذلك القواعد التي تحدد الحد الأقصى للمبالغ التي يمكن سحبها أو تحويلها من خلال النظام.

2. اختبار وظائف النظام وقابليته للاستخدام:

يتضمن ذلك اختبار واجهة المُستخدِم وتجربة المُستخدِم، على سبيل المثال في النظام المصرفي الإلكتروني يُمكن تشكيل مجموعة من أصحاب المصلحة لاختبار عمل النظام ووظائفه المختلفة كما هو مطلوب، وذلك من خلال القيام بالعمليات التي يقدمها النظام مثل: التحقق من الأرصدة أو إجراء المُعامَلات.

3. اختبار أخطاء التشغيل:

يتضمن ذلك اختبار الأخطاء المنطقية في البرمجة. فعلى سبيل المثال، من الضروري اختبار ما إذا كانت خطوات الحركة المالية تكتمل بشكل صحيح في النظام المصرفي الإلكتروني، وما إذا كانت الرسائل الموجهة للمُستخدِم تظهر وفق الخطوات المناسبة، مثل: رسائل الخطأ والتأكيد.

4. اختبار الاتصال بالأنظمة الأخرى:

يتم فيه اختبار ارتباط النظام بالأنظمة الأخرى بشكلٍ صحيح. في مثال نظام المصرف الإلكتروني، يتطلب ذلك اختبار مدى تكامل النظام الجديد بشكل جيدٍ مع أنظمة تقنية المعلومات الأخرى في المصرف مثل: قواعد بيانات العملاء، وأنظمة تحويل العملات، وأنظمة الصرف الآلي.

التنفيذ Implementation

تبدأ مرحلة التنفيذ بعد الحصول على موافقة المُستخدِم على النظام الجديد الذي تم تطويره واختباره، ويتم في هذه المرحلة تحويل الخطط التي تم وضعها وإقرارها إلى واقع وذلك من خلال وضع المُنتج (النظام) في الخدمة، ويتم فيها أيضًا تجهيز النظام للنشر والتثبيت ليكون جاهزًا للعمل وللاستخدام.

يشمل التنفيذ أيضًا تدريب المُستخدِمين النهائيين، وذلك للتأكد من معرفتهم بكيفية استخدام النظام ووظائفه. قد تستغرق مرحلة التنفيذ وقتًا طويلًا حسب درجة تعقيد النظام ووظائفه، ويتطلب التنفيذ أحيانًا نقل البيانات من النظام السابق إلى النظام الجديد، وغالبًا ما يُفضّل إدخال النظام الجديد للخدمة بشكلٍ تدريجي.

قد يتطلب تنفيذ نظام مصرفي إلكتروني جديد للبنك وضع مرحلةٍ انتقالية يتم فيها نشر إصدارٍ تجريبي للنظام، حيث يمكن للعملاء خلال هذه المرحلة اختبار النظام الجديد وتقديم ملاحظات حول تجربة استخدامه قبل إنتاج الإصدار النهائي من النظام.

شكل 1.2: تصميم وتنفيذ مُنتج برمجي

الصيانة Maintenance

تُعدُّ الصيانة ضرورية لمعالجة أي أخطاء خلال عمل النظام، وكذلك لضبط النظام حسب التغيرات في بيئات عمله المختلفة، ويجب أن يفي النظام بالاحتياجات الحالية، وأن يقدم كذلك تحسيناتٍ وميزات وظيفيّة مضافة أخرى للتعامل مع الاحتياجات المستقبلية. يتم تقييم النظام بشكل مستمر لضمان حداثته من خلال ملاحظات المُستخدِمين وتقييم فريق تقنية المعلومات، ومن الطبيعي أن يحتاج العمل بالنظام الجديد إلى إدخال بعض الإصلاحات أو التعديلات الصغيرة، ومن المهم إدراك أنّ الاحتياجات والمتطلّبات تتغير بانتظام، ولذلك يتعين على فريق تقنية المعلومات المحافظة على عمل النظام بشكلٍ مستمر كما هو متوقع.

عند تنفيذ النظام المصرفي الإلكتروني، ستكونُ الحاجة إلى الصيانة المستمرة للنظام قائمةً لضمان استمرار عمل النظام وأمنه وحداثته.

تتطلب الأنظمة توافر تحديثات البرامج والأجهزة الرئيسة والثانوية للحماية من التهديدات الأمنية الجديدة، وإصلاح الأخطاء غير المتوقعة، وتنفيذ وظائف جديدة، ويمكن لبعض عمليات صيانة النظام أن تتم تلقائيًا، مثل تحديثات الأمان التلقائية، ولكن كثيرًا من مهام الصيانة الأخرى كتحديثات الأجهزة ستتطلّب وجود المهندسين لتنفيذها.

التوثيق Documentation

تتضمن عملية التوثيق وصف جميع تفاصيل تحليل النظام وتصميمه وتطويره واختباره وتنفيذه وصيانته، ويتم استخدام تلك التفاصيل لإنشاء قاعدة معرفية لكيفية عمل النظام، وتتم الإشارة إلى توثيق النظام عند الحاجة إلى أي تغيير أو إصلاح أو تعديل. يجب أن يتم تحديث التوثيق ذاته من حين إلى آخر، ويُعدُّ التوثيق مهمًا في تطوير البرمجيات، حيث يساهم في تعزيز التواصل والشفافية، ويتيح القيام بالصيانة بشكلٍ فعّال، ويزيد من كفاءة النظام وسهولة التدريب وتحديث النظام، كما يساعد التوثيق على ضمان تطويرٍ نظام البرنامج باستمرار وبفعالية بما يلبي احتياجات المُستخدِمين المستهدفين بصورة دائمة.

التقييم Evaluation

يجب تقييم كل مرحلة من مراحل دورة حياة تطوير البرمجيات، وقد يتضمن ذلك اتخاذ بعض القرارات الصعبة، حيث أن وجود مشكلة في التصميم قد يؤدي إلى مشاكل أكبر لاحقًا أثناء التطوير أو عند تنفيذ النظام واستخدامه.

يمكن إجراء التقييم من قبل الجهات التالية للتأكد من أن النظام يفي بالمتطلّبات:

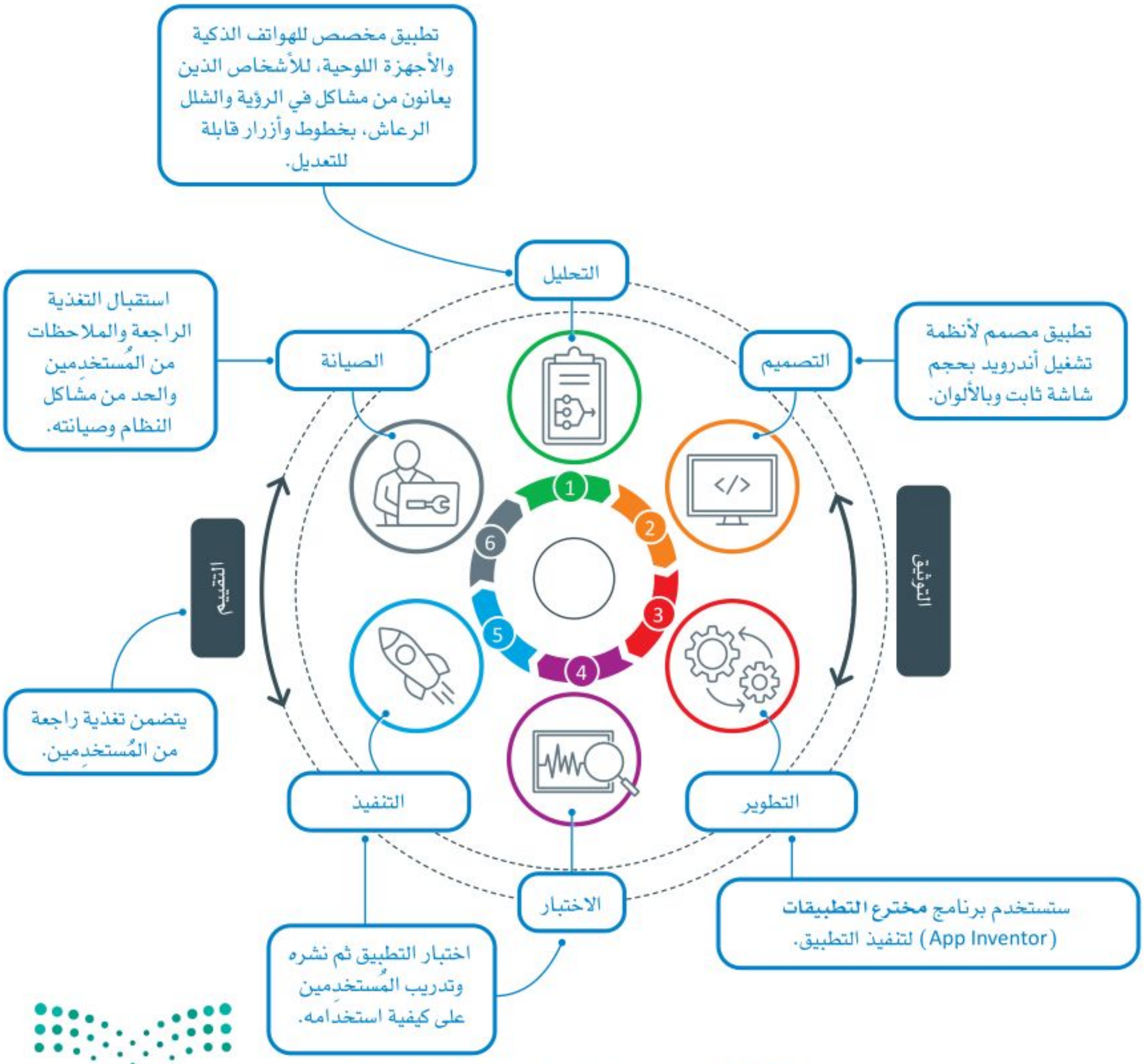
- فريق تقنية المعلومات.
- المُستخدِمون.
- الإدارة.

من المجالات التي تحتاج إلى تقييم مستمر:

- كفاءة النظام.
- سهولة الاستخدام والتعلُّم.
- ملاءمة النظام للمهام المطلوبة.

دورة حياة تطوير البرمجيات لتطبيق هاتف ذكي SDLC for a Smartphone Application

لنفترض أنك تريد إنشاء تطبيق للهاتف الذكي يوفر معلومات حول المناطق السياحية المختلفة في المملكة العربية السعودية. يهدف هذا التطبيق بشكلٍ خاص إلى مساعدة كبار السن الذين يعانون من مشاكل في الرؤية أو ارتجاف اليدين (الشلل الرعاش/ مرض باركنسون) في الحصول على معلومات حول المواقع السياحية التي يمكن زيارتها في المملكة العربية السعودية، وسيسمح التطبيق للأشخاص الذين يعانون من مشاكل في الرؤية بتعديل حجم خط النص بما يناسبهم ليتمكنوا من قراءة المعلومات بسهولة، وسيسمح لمن يعانون من مرض الشلل الرعاش بضبط حجم الأزرار لمنعهم من الضغط على زر بشكل غير صحيح أو عن طريق الخطأ، وأخيرًا سيتمكن المُستخدِمون من تغيير الألوان في التطبيق إلى الأبيض والأسود لتسهيل القراءة وتقليل إجهاد العين.



شكل 1.3: ملخص دورة حياة تطوير البرمجيات لتطبيق الهاتف الذكي

كما تعلمت سابقًا في هذا الدرس، ستتضمن دورة حياة تطوير البرمجيات لهذا التطبيق المراحل التالية: **التحليل والتصميم والتطوير والاختبار والتنفيذ والصيانة والتوثيق والتقييم المستمر.**

التحليل:

يتم في مرحلة التحليل تحديد المشكلة التي تحتاج إلى حل، وهكذا فإن التطبيق سيُصمم لما يلي:

- للهواتف الذكية والأجهزة اللوحية.
- للأشخاص الذين يعانون من مشاكل في الرؤية.
- للأشخاص الذين يعانون من ارتجاف اليدين بسبب مرض الشلل الرعاش.

بناءً على هذه المتطلّبات، يجب أن يتميز التطبيق بإمكانية تغيير حجم الخط وتعديل الأزرار، ويجب أن تكون الأزرار كبيرة جدًا بحيث يسهُل الضغط عليها، حتى يمكن استخدامها من قبَل الأشخاص الذين يعانون من ارتجاف اليدين.

التصميم:

تشمل مرحلة التصميم تحديد جميع التفاصيل الفنية للتطبيق، وبشكل أكثر تحديدًا، تشمل المتطلّبات الفنية ما يلي:

- يجب أن يكون التطبيق مصممًا لنظام تشغيل أندرويد.
- يجب أن يكون حجم الشاشة ثابتًا.
- يجب ألا يحتوي على ألوان كثيرة لتجنب إرباك المُستخدِمين.

التطوير والاختبار والتنفيذ:

يقوم مهندسو البرمجيات والمختبرين في مرحلتي التطوير والاختبار بالتنفيذ العملي للمتطلّبات والمواصفات الموضحة في الخطوات السابقة. سيتمّ استخدام برنامج **مخترع التطبيقات** (App Inventor) في هذه المرحلة لتطوير التطبيق المطلوب، وسيحتاج التطبيق بعد ذلك إلى اختبار شامل قبل نشره في متجر التطبيقات مثل **قوقل بلاي** (Google Play). قد يكون من الأفضل البدء بإصدار تجريبي من التطبيق يُتاح لعدد محدود من المُستخدِمين، من أجل إجراء اختبار إضافي للتطبيق قبل إصداره النهائي.

الصيانة:

تشمل مرحلة الصيانة جمع التغذية الراجعة من المُستخدِمين، وذلك من أجل استخدامها لتحسين التطبيق،ويتم الاستعانة بملاحظات المُستخدِمين لتقييم التطبيق بشكل مستمر لتحسينه ولضمان استمراريته في العمل، ويمكن عمل بعض التصحيحات أو التعديلات الصغيرة حسب الحاجة.

التوثيق والتقييم:

يعرّف التوثيق الخاص بتطبيق الهاتف الذكي على أنه مجموعة من الملاحظات والمواد المكتوبة التي توفر معلومات شاملة حول التطبيق، بما في ذلك تصميمه وتطويره وصيانته، وتساعد عملية التوثيق المطورين وأصحاب العمل والمُستخدِمين في فهم الغرض من التطبيق ووظائفه وعمله. أما فيما يتعلق بالتقييم، فيمكن جمع المعلومات من تصنيفات متجر قوقل بلاي ومراجعات التطبيق. يجب القيام بالخطوات التالية لتوثيق التطبيق:

- كتابة مُستند واضح يوضح تصميم النظام.
- إضافة التعليقات التوضيحية داخل أقسام التعليمات البرمجية أثناء عملية التطوير.
- توثيق عمليات اختبار النظام بشكلٍ دقيق.
- إعداد دليل المُستخدِم.

فرص العمل في هندسة البرمجيات Job Opportunities in Software Engineering

تقدم هندسة البرمجيات فرص عمل متنوعة في مختلف المجالات والصناعات. فيما يلي بعض فرص العمل الأكثر شيوعًا في هندسة البرمجيات:

- مطور ويب: يختص بتطوير مواقع وتطبيقات الويب.
- مطور تطبيقات الأجهزة المحمولة: يختص بتطوير تطبيقات الأجهزة المحمولة لمنصات آي أوإس (iOS) أو أندرويد (Android).
- مهندس عمليات التطوير: يختص بأتمتة عملية توزيع وتشغيل الأنظمة البرمجية.
- مهندس التخزين السحابي: يختص بإنشاء وصيانة أنظمة البرمجيات السحابية.
- مسؤول قاعدة البيانات: يختص بإدارة قواعد البيانات وصيانتها.
- مهندس ضمان الجودة: يختص باختبار الأنظمة البرمجية للتأكد من تلبيتها لمعايير الجودة.
- مسؤول النظام: يختص بصيانة وإدارة أنظمة وشبكات الحاسب.

منهجيات تطوير البرمجيات Software Development Methodologies

تختلف عملية تطوير نظم المعلومات الضخمة بشكلٍ جوهري عن عملية تطوير البرامج الصغيرة والتطبيقات، حيث يتطلب تطوير البرمجيات الضخمة كأنظمة المؤسسات الحكومية والشركات التجارية جهدًا كبيرًا وقد يستغرق شهورًا أو سنوات، وقد يُشكِّل فهم متطلّبات العملاء وطبيعة وظائف تلك الأنظمة أو البرامج تحديًا لفريق التطوير.

تهدف هندسة البرمجيات إلى تطوير الأساليب والبروتوكولات وطرق سير العمل والمراقبة للتغلب على هذه التحديات، وتوجد العديد من منهجيات تطوير البرمجيات، ويستخدم كل منها لأغراض مختلفة، ومن أكثر المنهجيات شيوعًا ما يلي:

منهجية الشلال The Waterfall Methodology

تُعدُّ هذه المنهجية من أقدم طرق تطوير البرمجيات، وقد سُميت بذلك لأن مراحل تطورها تتسلسل من مرحلة إلى أخرى بشكلٍ أحادي الاتجاه، وعند اكتمال مرحلة معينة من التطوير، يتم الانتقال إلى المرحلة التالية بدون العودة إلى المراحل السابقة، وتمثل مُخرَجات كل مرحلة مُدخَلات للمرحلة التالية.

شكل 1.4: التخطيط لتطوير منتج برمجي

تمر منهجية الشلال بكافة مراحل تطوير النظام بصورة متسلسلة، فكل مرحلة تعتمد على مُخرَجات المراحل التي تسبقها كما يلي:

1. مرحلة التحليل Analysis stage

يتم في هذه المرحلة جمع المتطلّبات بالطرق التي تم شرحها سابقًا، ويتم توثيقها بدقة وبالتفصيل عند الانتهاء من العمل. يتم التحقق من هذه المتطلّبات والموافقة عليها من قبل العميل قبل البدء بمرحلة التصميم، ويمكن أن تتسبب التغييرات التي يتم إجراؤها على المتطلّبات لاحقًا في العملية إلى تأخيرٍ كبير أو مشاكل خطيرة في النظام.

2. مرحلة التصميم Design stage

تتم ترجمة المتطلّبات الموثقة من المرحلة السابقة في تصميم يوضح مخطط النظام ويحدد احتياجاته من الموارد، ويعكس تصميم النظام كيفية تنفيذ المتطلّبات من ناحية فنية، والتسلسل المنطقي للعمليات التي تتم به.

3. مرحلة التطوير Development stage

يتم في هذه المرحلة إنشاء النظام وكتابة البرامج بناءً على مُخرَجات التصميم.

4. مرحلة الاختبار Testing stage

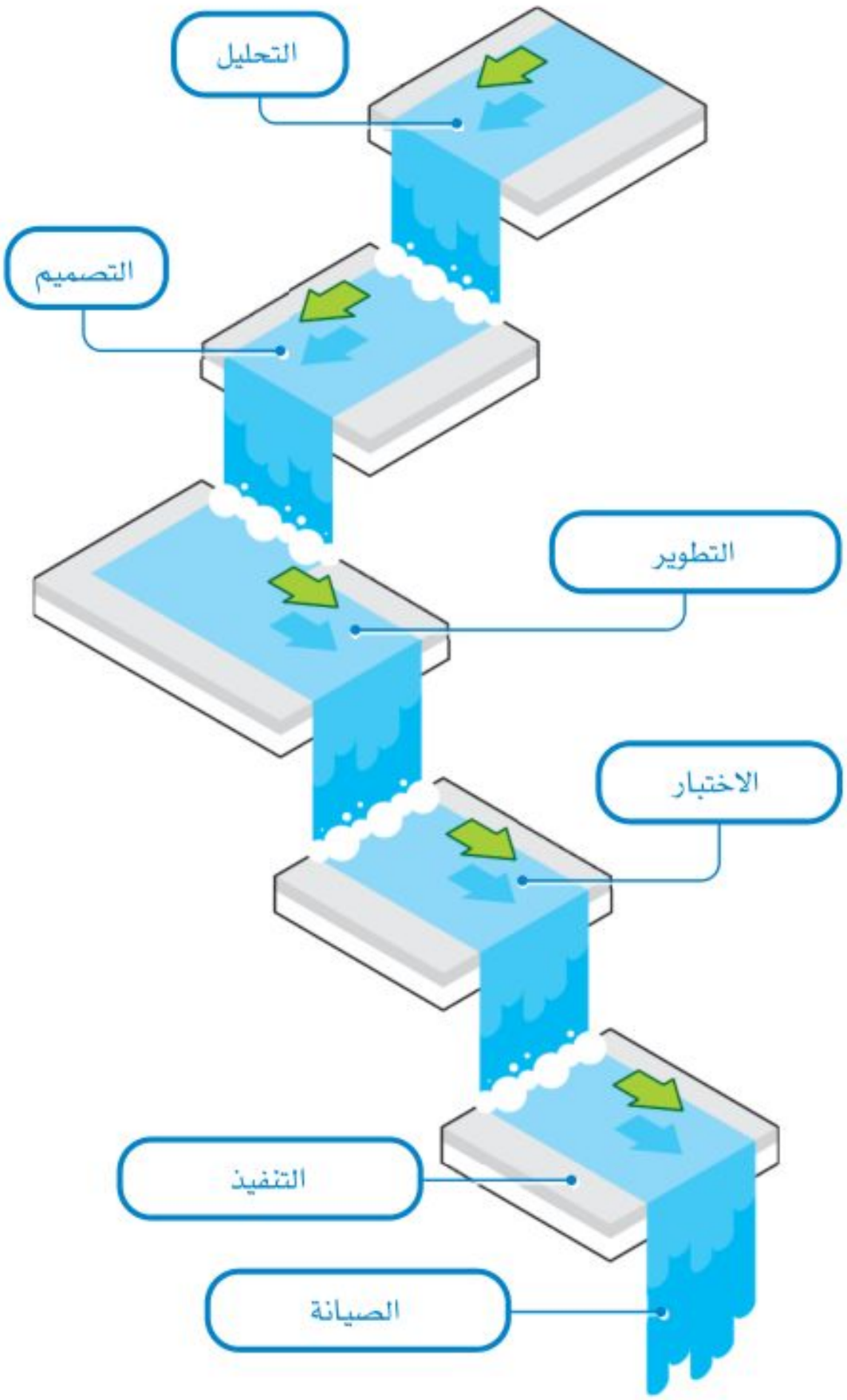
يتحقق مختبرو النظام في هذه المرحلة من استيفاء النظام للمتطلّبات الموثقَة في المراحل السابقة، ويقومون بالتحقق فيما إذا كانت هناك أية أخطاء يجب إصلاحها، ويطّلع العميل على المنتج الحقيقي لأول مرة في هذه المرحلة.

5. مرحلة التنفيذ Implementation stage

يتم في هذه المرحلة تنفيذ النظام وتسليمه للعميل، ويتم تدريب المُستخدِمين أو تأهيلهم، كما يتم وضع وتشغيل آليات مراقبة أداء النظام للتأكد من عدم وجود أخطاء أثناء التنفيذ.

6. مرحلة الصيانة Maintenance stage

تتضمن هذه المرحلة إصلاح الأخطاء التي تظهر أثناء الاستخدام اليومي للنظام، بالإضافة إلى إجراء بعض التطويرات والتحسينات على النظام.



شكل 1.5: مراحل منهجية الشلال

وزارة التعليم
Ministry of Education
2023 - 1445

مزايا منهجية الشلال في التطوير:

- تُعدُّ المراحل في هذه المنهجية واضحة ومُحددة، ولا تتداخل مع بعضها.
- يُعدُّ تخطيط المشروع وإدارته ومتابعته أمرًا سهلًا وذلك بسبب وضوح المراحل.
- تُناسب هذه المنهجية المشاريع الصغيرة ذات المتطلّبات الواضحة والثابتة.

تحديات منهجية الشلال:

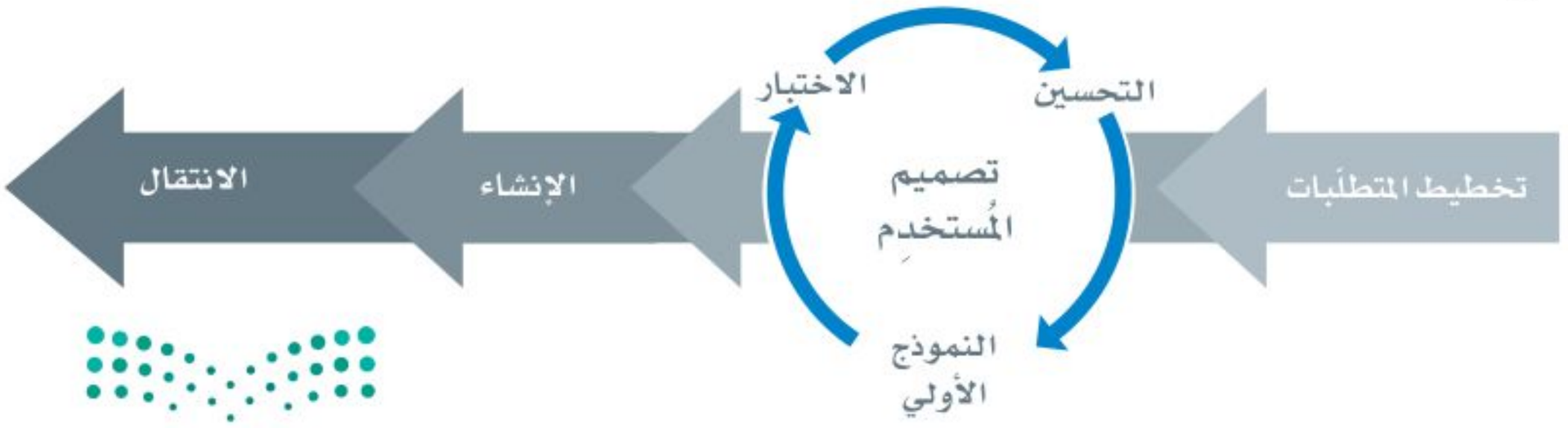
- في هذه المنهجية يصعب تعديل مرحلة ما بالعودة إلى سابقتها؛ وذلك لاعتماد كل مرحلة على المرحلة السابقة، ويؤثر الرجوع وإجراء التغييرات في المراحل السابقة على المراحل التالية بشكل كبير، ويزيد من تكلفة التطوير.
- غير مناسبة للأنظمة والبرامج الكبيرة والمعقدة.
- غير مناسبة للبرامج والأنظمة التي قد تخضع متطلّباتها للتغيير خلال مراحل المشروع.
- لا يمكن البدء بمرحلة جديدة في هذه المنهجية قبل إتمام المرحلة السابقة، ويؤدي هذا الأمر إلى تأخير اكتشاف أي خلل في فهم متطلّبات العميل، والذي قد يظهر في مرحلةٍ متأخرة، مما يجعل التعديل بعد ذلك عملية صعبة ومُكلفة، بل ويفاقم من احتمال فشل المشروع.

منهجية التطوير السريع للتطبيقات

Rapid Application Development Methodology (RAD)

على عكس منهجية الشلال التي تتم فيها عملية التطوير من خلال مراحل مُستقلة، تعتمد منهجية التطوير السريع للتطبيقات على التطوير من خلال دورات تكرارية، وتتضح السمة الرئيسة لهذه المنهجية في تطوير نماذج أولية للنظام من أجل الحصول على التغذية الراجعة والاقتراحات من العميل في المراحل الأولى من التطوير.

يساعد هذا في تجنب سوء فهم المتطلّبات، وبالتالي تجنب التكلفة الكبيرة التي تنطوي عليها الحاجة لتعديل النظام بعد اكتمال التطوير، ومن المهم الإشارة إلى أن النماذج الأولية التي يتم تطويرها تُعدّل لتصبح جزءًا من المنتج النهائي.



شكل 1.6: مراحل منهجية التطوير السريع للتطبيقات (RAD)

تتضمن هذه المنهجية المراحل التالية:

1. تخطيط المتطلّبات Requirements Planning

إن مرحلة تخطيط المتطلّبات في **منهجية التطوير السريع للتطبيقات** (RAD) هي مرحلة في عملية تطوير البرمجيات يتم فيها تحديد نطاق المشروع وأهدافه، وتُركِّز مرحلة تخطيط المتطلّبات على التأكد من أن متطلّبات المشروع والبرنامج واضحة ومُحدَّدة بشكل جيّد، وبأن المخطط الزمني والموارد اللازمة لإكمال المشروع قد تمّ تحديدهما.

2. تصميم المُستخدِم User Design

مرحلة تصميم المُستخدِم في التطوير السريع للتطبيقات هي مرحلة في عملية تطوير البرمجيات يتم فيها إنشاء متطلّبات البرامج وتصميمها من خلال التعاون الوثيق مع المُستخدِمين النهائيين،وتُركِّز مرحلة تصميم المُستخدِم على التيقن من أن تطبيق البرنامج يلبي احتياجات وتوقعات المُستخدِمين المستهدفين.

3. الإنشاء Construction

مرحلة الإنشاء في التطوير السريع للتطبيقات هي مرحلة في عملية تطوير البرمجيات يتم فيها تطوير وإنشاء البرنامج، وتتضمن هذه المرحلة كتابة التعليمات البرمجية، ودمج المكونات المختلفة لتطبيق البرنامج، واختباره للتأكد من أنه يفي بالمتطلّبات اللّازمة ومعايير الجودة.

تتضمن هذه المرحلة أيضًا إصلاح أي أخطاء أو مشاكل تم اكتشافها أثناء الاختبار، وتركز مرحلة الإنشاء على تقديم تطبيق برمجي فعّال يفي بالمتطلّبات المحددة بكفاءة في المراحل المبكرة من عملية تطوير التطبيقات السريعة.

4. الانتقال Cutover

مرحلة الانتقال في التطوير السريع للتطبيقات هي المرحلة الأخيرة من عملية تطوير البرمجيات، ويتم فيها نقل تطبيق البرنامج الجديد إلى بيئة التشغيل الواقعية، وتتضمن مرحلة الانتقال سلسلة من الأنشطة التي يجب القيام بها من أجل نقل التطبيق البرمجي الجديد إلى مرحلة التشغيل بنجاح، ولتدريب المُستخدِمين على كيفية استخدامه.

مزايا منهجية التطوير السريع للتطبيقات:

- يتلقى المطورون التغذية الراجعة المستمرة من المُستخدِمين النهائيين منذ بداية المشروع، مما يقلل من احتمالات الفشل وتكلفة التعديل.
- يشارك المُستخدِمون في التغذية الراجعة مما يساهم في إنتاج برمجيات أكثر كفاءة وأقل أخطاء.
- يساهم استخدام الدورات التكرارية والنماذج الأولية في تقليل الوقت اللازم لتطوير النظام.

تحديات منهجية التطوير السريع للتطبيقات:

- تُعدُّ دورة التطوير أكثر تعقيدًا ويجب إدارتها بعناية.
- قد يؤدي تفاعل أصحاب المصلحة في تقديم الملاحظات بشكل غير كافٍ إلى توفير مُنتج لا يلبي المتطلّبات المرجوّة.
- يتطلب اتباع هذه المنهجية وجود فريق عملٍ من المُبرمِجين والمُصمِّمين ذوي تأهيل عالٍ وقدرة على إنجاز المهام المنوطة بهم بشكلٍ فعَّال.

منهجية التطوير الرشيقة Agile Methodology

تَستخدِم منهجية التطوير الرشيقة طريقة تسليم المشروع على شكل إصدارات متتالية تُسمى **فترة التطوير الزمنية القصيرة** (Sprint). يضيف كل إصدار ميزات جديدة إلى الإصدار السابق، ويمر كل إصدار بجميع مراحل تطوير النظام من التخطيط إلى الاختبار والموافقة من قبل المُستخدِم. تتميز منهجية التطوير الرشيقة عن منهجية التطوير السريع للتطبيقات في أنها تُقدم مُنتجًا فعّالًا للمُستخدِم في كل مرحلة. يمكن في هذه المنهجية العمل على المنتج وتعديله أو إضافة وظائف جديدة عند الحاجة، وذلك على العكس من منهجية التطوير السريع للتطبيقات، والتي يتم بها تقديم نموذج أولي غير مكتمل للمُستخدِم من أجل تقديم التغذية الراجعة فقط، ولا يتم إصدار المنتج النهائي الذي تم تطويره إلا بعد تحديد جميع المتطلّبات بشكل كامل.

فترة التطوير الزمنية القصيرة (Sprint):

فترة التطوير الزمنية القصيرة هي تكرارٌ مُحدَّدٌ زمنيًا للعمل في تطوير البرمجيات الرشيقة، ويستمر عادةً من أسبوع إلى أربعة أسابيع، والغرض منها هو بناء المزيد من الخصائص في البرنامج وتوفيرها، مع التركيز على تحقيق الأهداف والغايات المحددة في البداية.

شكل 1.7: مراحل منهجية التطوير الرشيقة

مزايا منهجية التطوير الرشيقة:

- يمكن من خلالها إنجاز الإصدار الأول للمنتج بوقتٍ قصير.
- يتم تحديد مخاطر المشروع بسهولة من خلال التغذية الراجعة من المُستخدِمين.
- تمنح مشاركة أصحاب المصلحة في تطوير النظام المزيد من الثقة في البرنامج أو النظام الجاري تطويره.

تحديات منهجية التطوير الرشيقة:

- تُركِّز هذه المنهجية بشكل كبير على عملية التطوير وبدرجة أقل على التوثيق، ولذلك فإنه من الصعب دمج أعضاء جدد في فريق المشروع بعد البدء به.
- تؤثر استجابة وأداء المُستخدِمين على سرعة الإنتاج وجودة المنتج.
- يمكن أن يؤدي التغيير المتكرر في متطلّبات المشروع إلى تعطيل المشروع بأكمله، خاصّة عند قيام المُستخدِمين بتغيير رأيهم بشكل متكرر.

معلومة

تتطلب منهجية التطوير الرشيقة التواصل الفعّال والتعاون المستمر بين جميع الفرق المُشاركة في التخطيط والتصميم والتطوير واختبار المُستخدِم.

تمرينات

1 اختر المنهجية المناسبة التي تتوافق مع كلٍ من العبارات التالية:

1 منهجية الشلال	○ لا يمكن الانتقال للمرحلة التالية إلا بعد الانتهاء من المرحلة السابقة.
2 منهجية التطوير السريع للتطبيقات	○ تعتمد على طريقة تصميم النماذج الأولية وتحسينها.
3 منهجية التطوير الرشيقة	○ تستخدم دورات متكررة لتقليل وقت الإنتاج.
	○ أسرع طريقة للحصول على معاينة للمنتج الحقيقي.
	○ غير مناسبة للأنظمة الكبيرة والمعقدة.
	○ يتم تطوير المنتج على شكل إصدارات متتالية.

2 قارن بين مرحلتي التصميم والتنفيذ في المنهجيات الثلاث الموضحة في الدرس.

	منهجية الشلال	منهجية التطوير السريع للتطبيقات	منهجية التطوير الرشيقة
مرحلة التصميم			
مرحلة التنفيذ			

3 وضّح دور المُستخدِم في المنهجيات الثلاث، مع الإشارة إلى المرحلة التي يظهر فيها هذا الدور بشكل واضح.

4

حدّد الجملة الصحيحة والجملة الخاطئة فيما يأتي:	صحيحة	خاطئة
1. تشير عملية تطوير البرمجيات إلى التخطيط.	○	○
2. منهجية تطوير البرمجيات هي إطار يُستخدم في العديد من الإجراءات.	○	○
3. يتضمن تطوير البرمجيات إنشاء نظام المعلومات واختباره وتطويره.	○	○
4. يتم التحكم في عملية تطوير نظام المعلومات من خلال منهجية تطوير البرمجيات المستخدمة.	○	○
5. تقسم عملية تطوير البرمجيات ككل إلى مراحل متميزة، وتسمى أيضًا بدورة حياة تطوير البرمجيات .	○	○

5 املأ الفراغات بمراحل دورة تطوير البرمجيات في المخطط أدناه.

6 تأمل الشكل التالي ثم أجب عن الأسئلة التالية.

1. ما منهجية تطوير البرمجيات التي يمثلها الشكل؟

2. ما المقصود بمصطلح فترة التطوير الزمنية القصيرة؟

3. ما أهم ميزتين لهذه المنهجية؟

4. ما التحديان الرئيسان أمام هذه المنهجية؟

7 طابق كل مرحلة من مراحل تطوير النظام أدناه مع العمليات المناسبة في كل جُملة من الجُمل التالية:

1	التحليل	يتم تحويل النظرية (الخطط) إلى ممارسة (عمل).
2	التصميم	يتم إزالة أخطاء النظام أثناء العمل.
3	التطوير والاختبار	يتم تحويل المتطلّبات والمواصفات إلى تعليمات برمجية فعّالة.
4	التنفيذ	يتم هنا تعريف جميع تفاصيل النظام الجديد.
5	الصيانة	يعتمد على مقدار المعرفة التي يجب على الشخص المَعني معرفتها لفهم كيفية عمل النظام.
6	التوثيق	يتم تحديد المشكلة التي يجب حلها.
7	التقييم	لا يقتصر تنفيذه على فريق تقنية المعلومات ولكن يعتمد أيضًا على المُستخدِمين والإدارة

8 أنشئ مخططًا توضيحيًا لطريقة عمل كل من منهجيات تطوير البرمجيات الثلاث.

منهجية الشلال

منهجية التطوير السريع للتطبيقات

منهجية التطوير الرشيقة

9

اختر الإجابة الصحيحة:		
1. المرحلة التي تُستخدم فيها أدوات جمع البيانات هي:	التحليل.	○
	التصميم.	○
	التنفيذ.	○
	الصيانة.	○
2. المرحلة التي تُستخدم فيها لغة برمجة أو برنامج حاسوبي لإعداد النظام هي:	التصميم.	○
	التقييم.	○
	التطوير.	○
	التوثيق.	○
3. المرحلة التي يتم فيها إعداد دليل المُستخدِم الخاص بالنظام هي:	التحليل.	○
	التوثيق.	○
	التقييم.	○
	الاختبار.	○
4. في مرحلة تقييم تطبيق الهاتف الذكي:	يتم إنشاء التطبيق باستخدام برنامج مخترع التطبيقات.	○
	يتم تحديد احتياجات المُستخدِم.	○
	يتم تلقي المراجعة من المُستخدِمين.	○
	يتم تصميم التطبيق للعمل على منصة أندرويد.	○

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

لغات البرمجة ومُعالجاتها

تاريخ مختصر لتطور لغات البرمجة

A Brief History of the Development Programming Languages

شهد العالم تغيُّرًا كبيرًا منذ أن تم بناء أول جهاز حاسب وحتى يومنا هذا، وتطورت مكونات وتقنيات الحاسب بشكل كبير، كما تطورت قدرات المُعالجة الحاسوبية المتقدمة. ورغم هذا التقدم، ما زالت الكثير من مفاهيم تشغيل الحاسب التي صاغها **فون نيومان** (Von Neumann) في عام 1945 سارية حتى يومنا هذا.

شكل 1.8: تطور لغات البرمجة

تم اختراع لغات البرمجة لغرض التواصل بين الإنسان والآلة.

لغة الآلة Machine Language

يجب إعطاء الحاسب الأوامر على شكل سلسلة أرقام ثنائية تتكون من الرقمين 0 و 1، ويُطلق على اللغة التي تقدم بها هذه الأوامر اسم **لغة الآلة** (Machine Language)، ولا يستطيع الإنسان فهم هذه اللغة، ولا يمكن للمبرمج كتابة البرامج وتنفيذها بشكلٍ مباشر باستخدام هذه اللغة، حيث يتطلب الأمر معرفة عميقة بمكونات الحاسب، خاصةً أن لغة الآلة تختلف باختلاف بنية **وحدة المعالجة المركزية** (Central Processing Unit -CPU) في الحاسب.

يتكون البرنامج بلغة الآلة من سلسلة من التعليمات الصادرة للمعالج لتنفيذ العمليات الأساسية على شكل الوحدات الثنائية (binary bits).

لغة التجميع Assembly Language

- لغة التجميع هي لغة برمجة تقع من حيث سهولتها بين لغة الآلة ولغات البرمجة عالية المستوى، ويطلق عليها أيضًا اسم لغة البرمجة الرمزية.
- تتشابه لغة التجميع مع لغة الآلة، ولكنها تُعدُّ أكثر سهولة في البرمجة حيث تتيح للمبرمج استبدال الرقمين (0، 1) بالرموز.

يتم تحويل أوامر لغة التجميع التي يمكن فهمها من قبل الإنسان إلى سلسلة من الأعداد التي تتكون من الرقمين 0 و 1 لكي يفهمها الحاسب وينفذها.

في لغة التجميع على سبيل المثال، يتم استخدام كلمة ADD متبوعة برقمين للقيام بعملية الجمع.

تُعَدُّ هذه الأوامر سهلة للفهم والحفظ، ولكن يجب ترجمتها إلى سلسلة من الأرقام الثنائية (0 و1) داخل الحاسب لتنفيذ العملية المطلوبة، وتتم عملية الترجمة هذه بواسطة برنامج يُسمى **المُجمّع** (Assembler).

تتكون أوامر لغة التجميع من مقاطع رمزية تتوافق مع أوامر لغة الآلة.

تحديات استخدام لغة التجميع Challenges of Assembly Language

- رغم كونِها لغة منخفضة المستوى، إلا أنّ استخدامَها يمكّن من برمجة العمليات البسيطة ذات التسلسلات الثنائية بسهولة.
- تختلف لغة التجميع المُستخدمة حسب بنية كل حاسب.
- تقتصر أوامر تنفيذ الوظائف في لغة التجميع على العمليات البسيطة كالجمع والضرب والمقارنات، وتفتقر للأوامر التي يمكن أن تدعم تنفيذ وظائف أكثر تعقيدًا، مما يُجبر المبرمج على كتابة برامج طويلة ومعقدة يصعب فهمها وتصحيحها.
- لا يمكن نقل برنامج من جهاز حاسب إلى آخر له بنية مُختلفة.

يوضِّح الجدول التالي برنامجًا للجمع تمت كتابته بلغة برمجة عالية المستوى وما يكافئه في لغة التجميع ولغة الآلة، وذلك لجهاز حاسب مزود بوحدة معالجة مركزية 6502 ذات 8 بت. يُمكن استخدام برامج اللغة عالية المستوى على معظم أجهزة الحاسب، بينما تعمل لغة التجميع ولغة الآلة فقط على جهاز حاسب بنفس عمارة وحدة المعالجة المركزية.

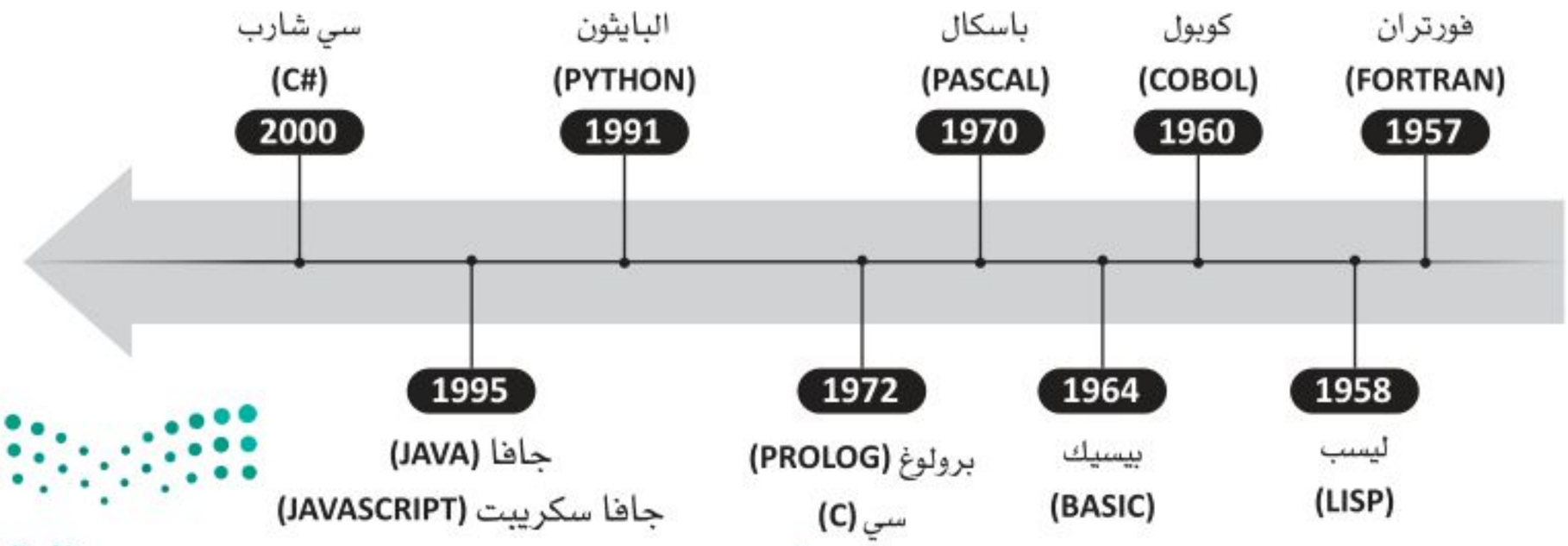
جدول 1.1: حساب عملية جمع

لغة عالية المستوى	لغة التجميع	لغة الآلة
sum = 0	LDA #0	10101001 00000000
	STA sum	10000101 00000000
	LDA sum	10100101 00000000
	CLC	00011000 00000000
sum = sum + 5	ADC #5	01101001 00000101
	STA sum	10000101 00000000
	LDA sum	10100101 00000000
print (sum)	JSR print	00100000 11100001

لغات البرمجة عالية المستوى High-Level Programming Languages

أدَّت أوجه القصور في لغة الآلة ولغة التجميع إلى تضافُر الجهود لتحقيق تواصل أفضل بين الإنسان والآلة، مما أدى إلى ظهور أول لغة برمجة عالية المستوى في الخمسينيات من القرن الماضي.

تَستخدم لغات البرمجة عالية المستوى أوامر برمجية تشبه لغة الإنسان، ثم يتم ترجمتها إلى لغة الآلة بواسطة الحاسب نفسه باستخدام برامج خاصة تُسمى مُترجِمات اللغة. تستخدم **المُترجِمات** (Compliers) و**المُفسِّرات** (Interpreters) مع أنواع مختلفة من لغات البرمجة.



شكل 1.9: تطوُّر لغات البرمجة

وزارة التعليم
Ministry of Education
2023 - 1445

تطور لغات البرمجة عالية المستوى
The Evolution of High-Level Programming Languages

يحدد المطور لغة البرمجة المناسبة لتطوير التطبيقات بسهولة وذلك بناء على معرفته ومهاراته وتفضيلاته، مع الأخذ بالاعتبار البيئة التقنية المطلوبة لتنفيذ الحل البرمجي.

تحتوي كل لغة برمجة على مجموعة فريدة من الكلمات المحجوزة (الكلمات التي تحتويها تلك اللغة)، وعلى الصيغ الخاصة التي يجب على المطور أن يستخدمها لكتابة التعليمات البرمجية.

جدول 1.2: المعلومات الأساسية للغات البرمجة

لغة البرمجة	المُطور	أصل الكلمة	الخصائص
فورتران (FORTRAN)	المؤسسة الدولية للحواسيب (IBM)	ترجمة الصيغة (FORmula TRANslation)	مناسبة لحل المشاكل الرياضية والعلمية، ولكنها غير مناسبة لإدارة ملفات البيانات.
ليسب (LISP)	معهد ماساتشوستس للتقنية (MIT)	معالج القائمة (LISt Processor)	لغة خاصة بالذكاء الاصطناعي.
كوبول (COBOL)	CODASYL	لغة الأعمال الموجهة المشتركة (Common Business Oriented Language)	مناسبة لتطوير التطبيقات التجارية وتطبيقات الإدارة العامة.
بيسيك (BASIC)	كُلية دارتموث (Dartmouth)	لغة تعليمات رمزية لجميع الأغراض للمبتدئين (Beginner's All Purpose Symbolic Instruction Code)	لغة برمجة متعددة الاستخدامات.
باسكال (PASCAL)	البروفيسور نيكولاس ويرث (Nicholas Wirth)	سُميت على اسم عالم الرياضيات بليز باسكال (Blaise Pascal)	تشتهر بإدخال تقنيات البرمجة المنظمة. وتعتمد على تصميم البرنامج بطريقة منهجية ودقيقة.
سي (C)	دينيس ريتشي (Dennis Ritchie) ومختبرات بيل (Bell Labs)	تمت تسمية لغة سي (C) على اسم لغة سابقة تدعى بي (B)	استخدمت لتطوير نظام التشغيل يونكس (UNIX)، وتناسب أنظمة التشغيل المختلفة.
جافا (JAVA)	أنظمة Sun	سُميت نسبة إلى نوع من القهوة (جافا)	لغة برمجة موجهة للكائنات تُستخدم لتطوير التطبيقات التي يمكن تشغيلها على مجموعة كبيرة من أجهزة الحاسب أو أنظمة التشغيل المختلفة.

ميزات لغات البرمجة عالية المستوى:

تتمتع لغات البرمجة عالية المستوى بالعديد من المزايا مقارنة بلغة التجميع حيث:

- تَستخدم صيغة منطقية وبرمجية يُمكن فهمها لقربها من لغة الإنسان.
- تَتَّسِمُ بعدم ارتباطها بنوع وبُنية حاسب معين، وبالتالي يمكن استخدامها على أي جهاز دون الحاجة إلى تعديلات، أو بعد عمل تعديلاتٍ طفيفة.
- يُمكن للمطورين تعلم لغات البرمجة عالية المستوى بسهولة وسرعة أكبر.
- تُعدُّ عملية تصحيح أخطاء البرامج وصيانتها أسهل بكثير.

بشكل عام، تعمل لغات البرمجة عالية المستوى على تقليل وقت وتكلفة تطوير البرامج بشكل كبير بالمقارنة مع لغات البرمجة منخفضة المستوى.

لغات برمجة الجيل الرابع Fourth-Generation Programming Languages

توجد ضمن لغات البرمجة عالية المستوى فئة يُطلق عليها لغات برمجة الجيل الرابع، والتي عادةً يتم اختصارها بالرمز 4GL، وتُعدُّ هذه اللغات أقرب إلى لغة الإنسان من اللغات عالية المستوى الأخرى ويمكن استخدامها من قِبل الأشخاص العاديين غير ذوي الاختصاص كالمبرمجين لأنها تتطلب قدرًا أقل من كتابة التعليمات البرمجية.

تُعدُّ لغات الجيل الرابع أكثر ملاءمةً للمبرمجين، وتُعزِّز كفاءة عملية البرمجة باستخدام كلمات وعبارات تشبه اللغة الإنجليزية، إضافةً إلى الرموز والتمثيلات الرمزية والواجهات الرسومية عند الحاجة، ويعتمد الحصول على أفضل النتائج باستخدام هذه اللغات على التوافق بين الأداة وطبيعة التطبيق.

يمكن لمُستخدِمي الحاسب في لغات الجيل الرابع إجراء تغييرات على البرنامج بشكلٍ سريع من أجل تلبية متطلّبات جديدة، مع القدرة على حل المشاكل الصغيرة بأنفسهم، ويمكن في هذه اللغات إجراء عمليات متعددة باستخدام أمر واحد يُدخله المبرمج. لغات البرمجة النصية هي نوع من لغات البرمجة يتم عادةً تفسيرها مباشرةً لإعطاء النتائج بدلًا من ترجمتها إلى برنامج بلغة التجميع.

يتم استخدام هذه اللغات لأتمتة المهام المتكررة، ولتبسيط العمليات المعقدة، ولإنشاء النماذج الأولية السريعة لأنظمة البرامج. تتضمن بعض الأمثلة الشائعة للغات البرمجة النصية **جافا سكرييت** (JavaScript) **وروبي** (Ruby) **وبي إتش بي** (PHP) **وبيرل** (Perl)، وتتميّز هذه اللغات باحتوائها على مكتبات برمجية متعددة تتيح القيام بالمهام المطلوبة من البرامج بشكلٍ فعّال، مما يجعلها مناسبة تمامًا للمهام التي تتطلب نماذج أولية سريعة ومتكررة. تتسم بعض هذه اللغات بمحدوديةٍ قدراتها وإمكانية تطويرها كما هو الحال في اللغات المُترجمة، وقد لا تناسب التطبيقات عالية الأداء، أو تلك التي تتطلب الكثير من الموارد لتشغيلها.

عند الحاجة لاستخراج البيانات، يُمكن للمُستخدِم إنشاء استعلامات وتقارير باستخدام **لغة الاستعلام الهيكلية** (SQL)، وذلك عند الحاجة لبرمجة العمليات الإحصائية والمشاريع العلمية، حيث يمكن للباحث أو لعالم الرياضيات استخدام برامج مثل: برنامج **إس بي إس إس** (SPSS) **وماتلاب** (MATLAB) **ولابفيو** (LabVIEW) لتحليل هذه البيانات.

تصنيفات لغات البرمجة Classifications of Programming Languages

هناك عدة تصنيفات للغات البرمجة، ويمكن تصنيف تلك اللغات من حيث نوع الأوامر المستخدمة، مثل **لغات البرمجة الإجرائية** (Procedural Programming Languages) و**لغات البرمجة الموجهة للكائنات** (Object-Oriented Programming Languages).

تستخدم البرمجة الإجرائية مجموعة من التعليمات لإخبار الحاسب بما يجب القيام به خطوة بخطوة. من الأمثلة على لغات البرمجة الإجرائية **كوبول** (COBOL) و**فورتران** (Fortran) وكذلك لغة البرمجة **سي** (C).

يعتمد البرنامج في البرمجة كائنية التوجه على وحداتٍ تسمى **الكائنات** (Objects). من الأمثلة على لغات البرمجة كائنية التوجه لغات **سي شارب** (C#) و**سي بلس بلس** (C ++) و**جافا** (Java) و**البايثون** (Python).

يمكن أيضًا تصنيف لغات البرمجة وفقًا لغرض الاستخدام إلى:

1. لغات برمجة عامة: يُمكن نظريًا استخدام أي لغة برمجة عامة لحل أي مشكلة، ولكن عند النظر بشكلٍ عملي، فقد تم تصميم كل لغة لحل نوع معين من المشاكل.
   تُقسَّم هذه اللغات على النحو التالي:
   - لغات علمية مثل الفورتران.
   - لغات موجهة للأعمال مثل كوبول.
   - لغات متعددة المجالات مثل بيسيك و باسكال.
   - لغات برمجة لأنظمة التشغيل مثل سي.
   - لغات للذكاء الاصطناعي مثل برولوغ.
   - لغات إدارة قواعد البيانات المتخصصة مثل لغة الاستعلام الهيكلية.
2. اللغات المتخصصة: مثل **ليسب** (LISP) والتي تستخدم لنوع معين من التطبيقات مثل الروبوتات أو الدوائر المتكاملة.

كيف تفهم أجهزة الحاسب لغات البرمجة؟ How Computers Understand Programming Languages

يتم تحويل البرنامج الذي تمت كتابته بأي من لغات البرمجة إلى لغة آلة يمكن للحاسب فهمها وتنفيذها باستخدام برامج ترجمة خاصة.

توجد طريقتان لتشغيل البرامج المكتوبة بلغة عالية المستوى، أكثرها شيوعًا هو ترجمة البرنامج باستخدام **مُترجِم** (Compiler)، وتعتمد بعض اللغات على استخدام **المُفسِّر** (Interpreter) عِوضًا عن ذلك.

شاهد كيفية تنفيذ هاتين الطريقتين المختلفتين.

المُترجِم (Compiler):

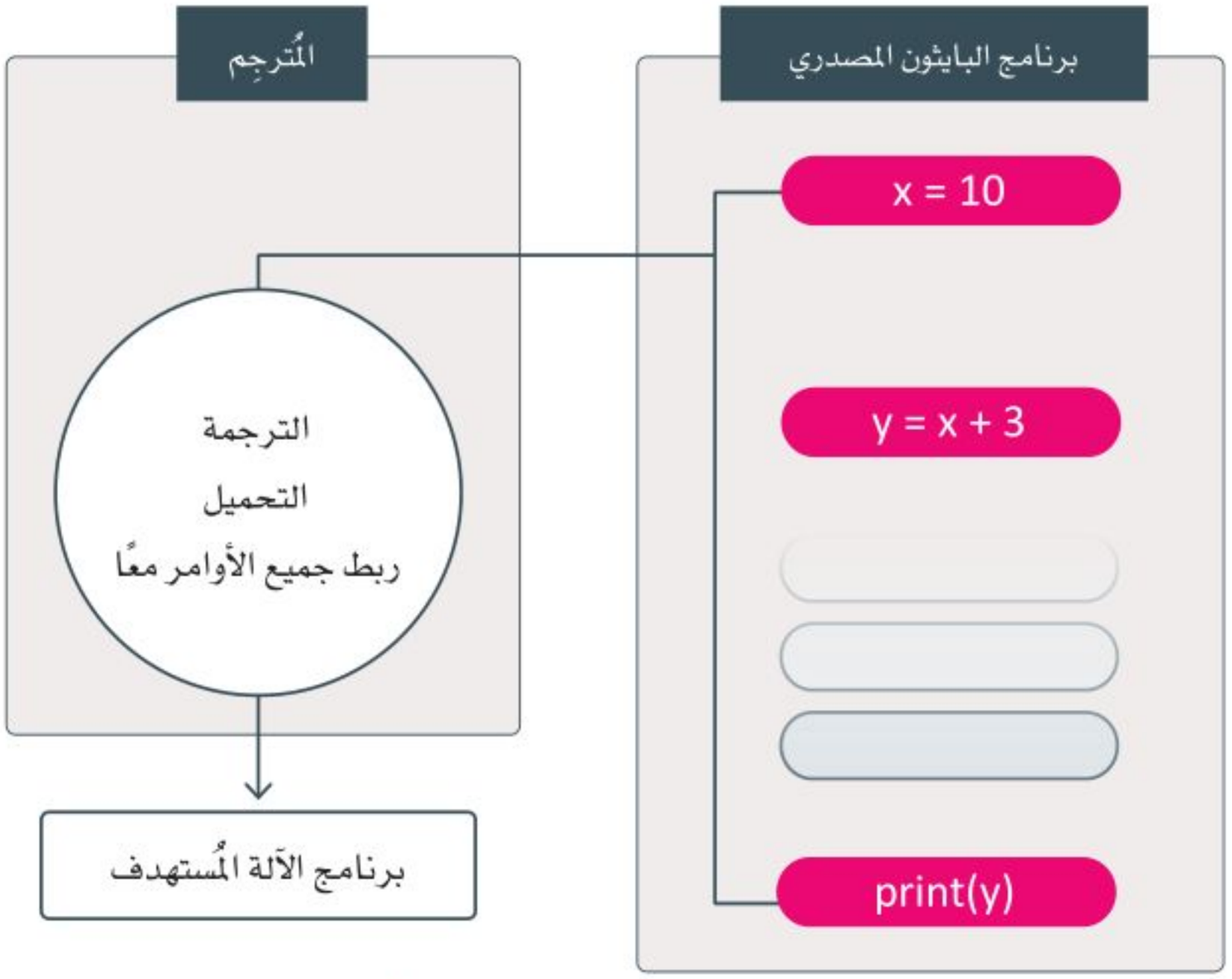
المترجِم هو برنامج حاسب يقوم بتحويل كتلة كاملة من التعليمات البرمجية المكتوبة بلغة برمجة عالية المستوى إلى لغة الآلة التي يفهمها معالج الحاسب.

المُفسِّر (Interpreter):

المُفسِّر هو برنامج حاسب يقوم بتحويل كل سطر من التعليمات البرمجية من مجموعة التعليمات البرمجية المكتوبة بلغة عالية المستوى إلى لغة الآلة ويرسلها للتنفيذ مباشرة قبل الانتقال إلى السطر التالي من التعليمات البرمجية.

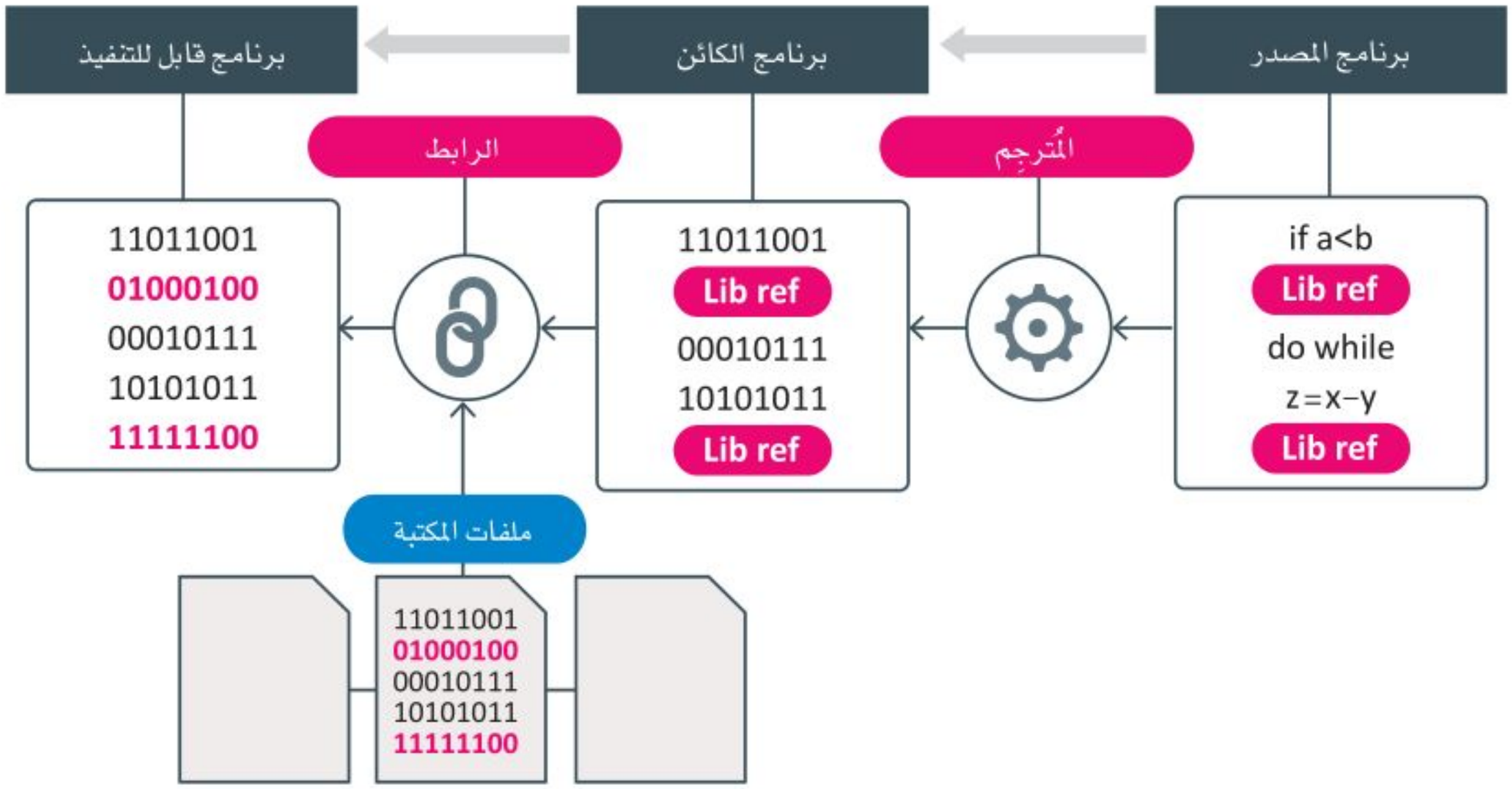
عملية ترجمة البرنامج وربطه:

- يقبل المُترجِم برنامجًا مكتوبًا بلغة عالية المستوى كملف إدخال أو **كبرنامج المَصدر** (Source Code)، وينتج برنامجًا مكافئًا بلغة الآلة يُسمى **برنامج الكائن** (Object Code).
- لا يستطيع المُترجِم تجميع الجُمل التي تشير إلى المكتبات القياسية أو الموارد خارج برنامج المصدر، ولهذا تتطلب العملية خطوة إضافية لربط وتحويل هذه الجُمل.
- يتولى برنامج آخر يسمى **الرابط** (Linker) أو **المُحمِّل** (Loader) عملية الربط، حيث يربط ملف برنامج الكائن بملفات المكتبة القياسية، ليُنتج البرنامج القابل للتنفيذ، وهو البرنامج النهائي الذي ينفذه الحاسب.

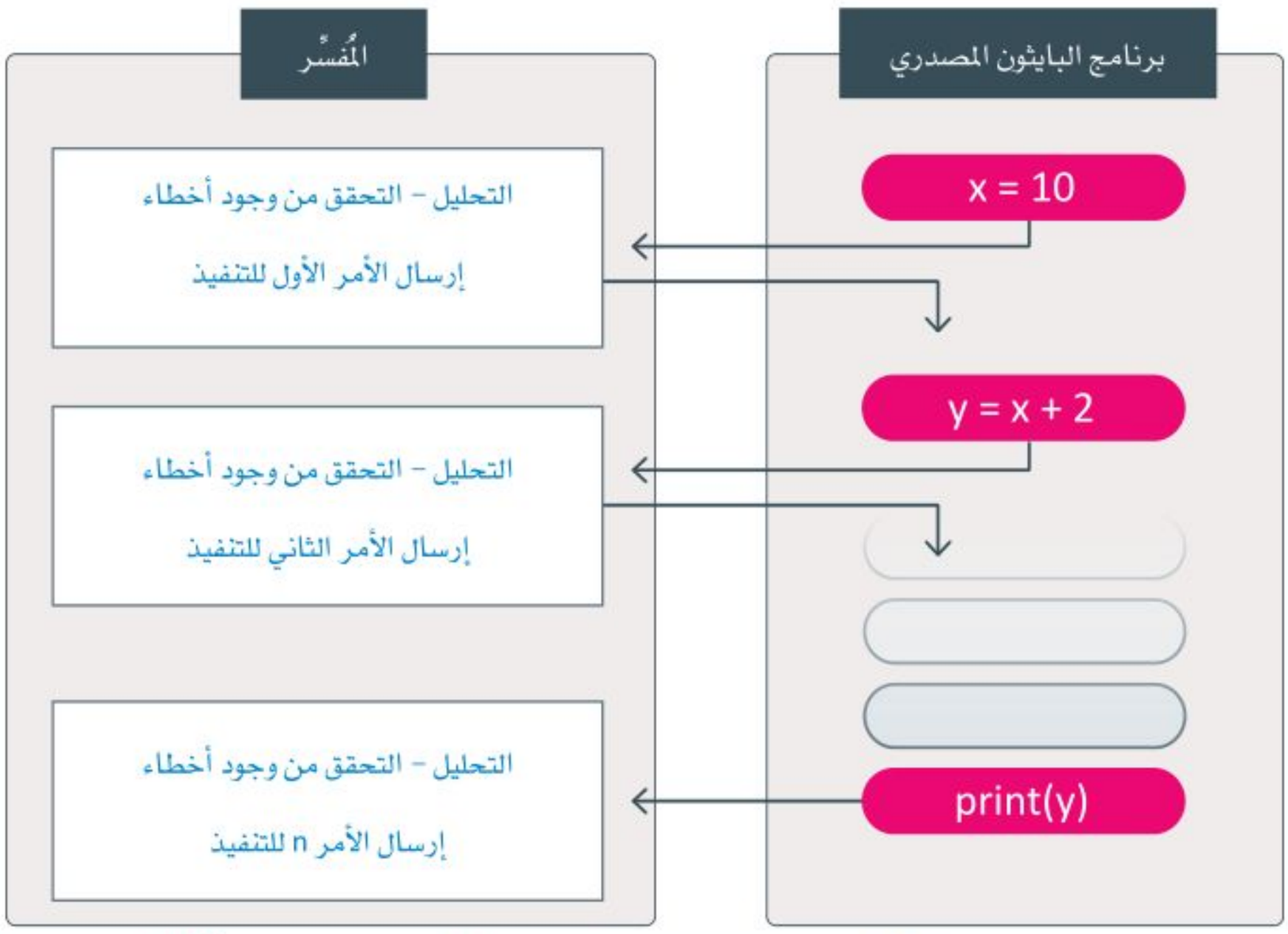


شكل 1.10: عملية تجميع وتنفيذ برنامج باستخدام مُترجِم

البرنامج المصدري هو برنامج تمت كتابته بلغة برمجة عالية المستوى.



شكل 1.11: مراحل ترجمة وربط البرنامج



شكل 1.12: عملية تجميع وتنفيذ برنامج باستخدام المُفسِّر

تقوم المُترجِمات والمُفسِّرات الفورية بالمهمة ذاتها، وهي تحويل البرنامج المكتوب بلغة البرمجة عالية المستوى إلى لغة الآلة، ولكن تعمل كل منهما بطريقةٍ مختلفة.

لغات البرمجة المُترجَمة والمُفسّرة

Interpreted and Compiled Programming Languages

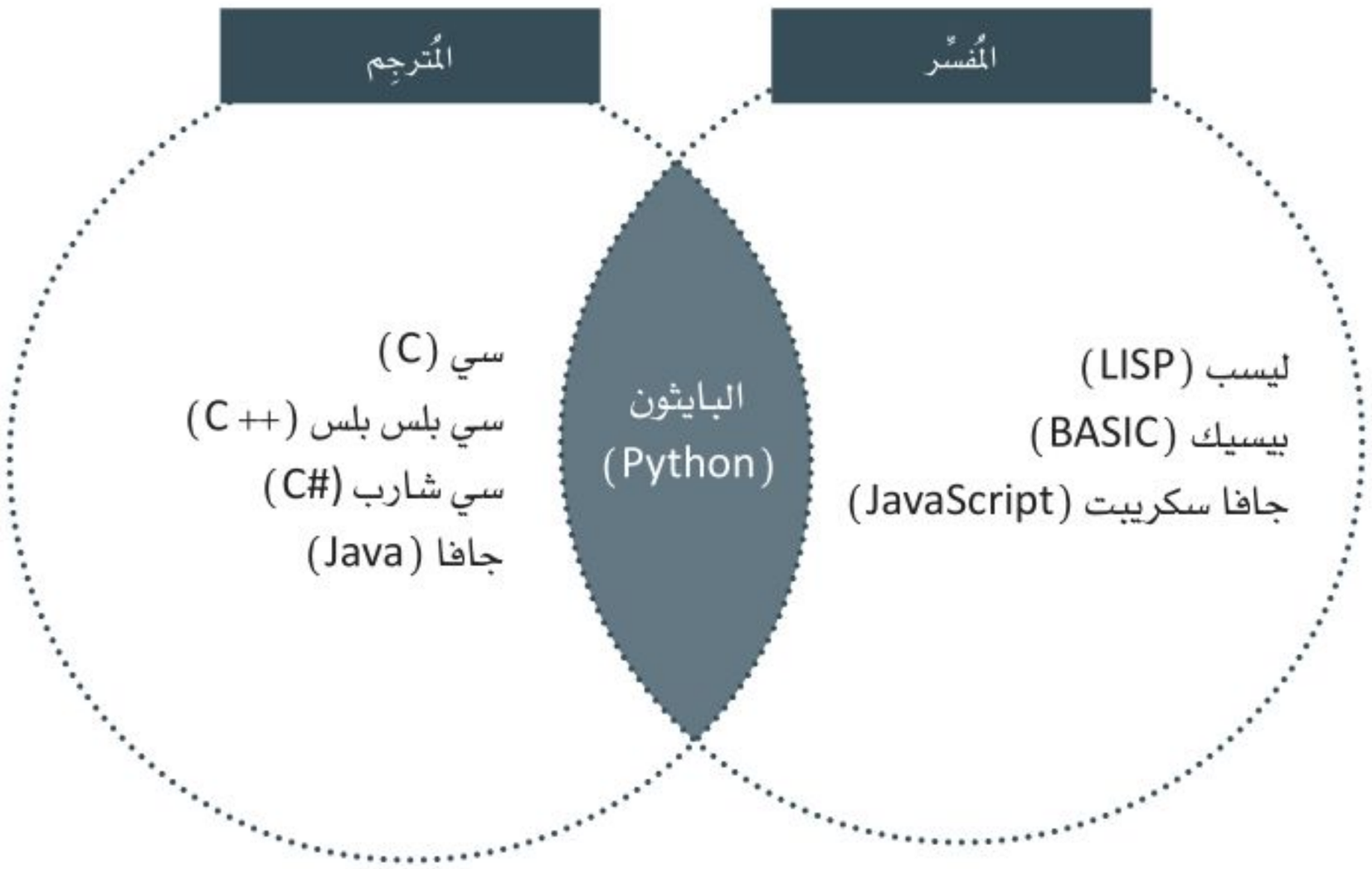
تَستخدِم معظم لغات البرمجة الحديثة المُترجِمات لإنتاج البرامج المُحسَّنة بسرعة، ولكن بعض اللغات ما زالت تستخدم المُفسِّرات عند الحاجة لإنشاء برنامج بسيط لا تعد السرعة المعيار الأهم به.

اللغات المُترجَمة Compiled Languages

تستخدم لغات البرمجة **سي** (C) و**سي بلس بلس** (C ++) و **سي شارب** (C#) و**جافا** (Java) مُترجِمات اللغة لإنشاء برامج سريعة وموثوقة. يتم إنشاء البرنامج القابل للتنفيذ لكل نوع من أنواع أجهزة الحاسب، مما يجعل اكتساب المعرفة بأجهزة حواسيب المُستخدِمين النهائيين واجبًا على المطوِّرين.

اللغات المُفسّرة Interpreted Languages

اعتمدت الإصدارات الأوليّة للغة **جافا سكريبت** (JavaScript)، وكذلك لغتي **ليسب** (LISP) و**بيسيك** (BASIC) على المُفسِّرات، مما يعني أن البرامج التي تكتب بهذه اللغات تعمل بشكل بطيء، ولكن كان تشغيل برامجها المصدرية على أي جهاز حاسب يحتوي على مُفسِّر لغة البرمجة ممكنًا، فعلى سبيل المثال، يمكن تشغيل تطبيق ويب مكتوب بلغة جافا سكريبت على جهاز حاسب يعمل بنظام ويندوز أو على حاسب لوحي يعمل بنظام أندرويد باستخدام متصفح ويب مُدمج مع مُفسِّر .



شكل 1.13: بايثون لغة مُترجمة ومُفسّرة

إن البايثون هي لغة مُترجمة ومُفسَرة. يُترجم تطبيق بايثون كل سطر من التعليمات البرمجية بحيث يمكن للمُفسِّر قراءته على الجهاز المُستخدَم. لا يتغير تركيب الجُمل الذي يستخدمه المبرمج لأن تطبيق بايثون يحوله إلى الشكل الصحيح للمُفسِّر المُستخدَم على هذا الجهاز.

جدول 1.3: المقارنة بين المُترجِم والمُفسِّر

	المُترجِم	المُفسِّر
الوظيفة الرئيسة	يحوِّل البرنامج المصدري المكتوب بلغة البرمجة عالية المستوى بأكمله إلى لغة آلة، وينتج برنامجًا قابلًا للتنفيذ.	يحوِّل كتلة من التعليمات البرمجية إلى لغة الآلة بحيث يترجمها ثم ينفذها، ثم ينتقل إلى الكتلة التالية أثناء تشغيل البرنامج.
المُدخَلات	يأخذ المُترجِم البرنامج المصدري بأكمله كمدخلات.	يأخذ المُفسِّر أحد تعليمات برنامج المصدر كمُدخل في كل مرة.
المُخرَجات	ينشئ المُترجِم ويخزن ملف برنامج الكائن كمُخرج.	لا يُنشئ المُفسِّر ملف برنامج الكائن.
الذاكرة	يتطلب المزيد من الذاكرة بسبب إنشاء برنامج الكائن.	يتطلب ذاكرة أقل.
عملية التنفيذ	تتم عملية تجميع برنامج المَصدر بالكامل قبل بدء التنفيذ.	تتم عملية التفسير لكل عبارة برمجية بالتوازي مع عملية التنفيذ.
التحقق من الأخطاء	يعرض المُترجِم جميع أخطاء اللغة والتحذيرات عند ترجمة البرنامج، ولا يمكن تشغيل البرنامج حتى يتم تصحيح جميع الأخطاء.	يقرأ المُفسِّر سطرًا واحدًا من التعليمات البرمجية ويعرض أي أخطاء فيه، ويجب تصحيح الأخطاء قبل الانتقال لقراءة السطر التالي.
ربط الملفات	يحتاج إلى برنامج لربط ملف الكائن بملفات المكتبة القياسية لإنشاء الملف القابل للتنفيذ.	لا يحتاج إلى عملية الربط، ولا يُنشئ ملفًا قابلًا للتنفيذ.
السرعة	يتيح توفر الملف التشغيلي (exe.) التنفيذ بشكلٍ أسرع.	تكون عملية التنفيذ أبطأ لأن الملف القابل للتنفيذ غير متوفر، ويتم تفسير البرنامج مرة أخرى عند كل عملية تنفيذ.
الاعتماد على الأجهزة وأنظمة التشغيل	يعتمد الملف القابل للتنفيذ الذي تم إنشاؤه بواسطة المُترجِم على الأجهزة التي سيتم تشغيله عليها، ولا يمكن تشغيله على وحدات معالجة مركزية ذات عمارة مختلفة، أو على أنظمة تشغيل مختلفة.	المُفسِّر هو مُكون مستقل عن نظام التشغيل، فعلى سبيل المثال يمكن لمُترجِم البايثون العمل على نظامي ويندوز ولينكس بنفس البرنامج المصدري وبنفس النتائج.

التعامل مع أخطاء البرامج Dealing with Software Errors

تعمل المُترجِمات والمُفسِّرات بشكل مختلف عند وجود أخطاء أو خلل في البرامج المصدرية.

المُفسِّر:

1. إنشاء البرنامج.
2. يقرأ المُترجِم سطرًا واحدًا من التعليمات البرمجية ويعرض أي خطأ في تركيب الجُمل، ويجب تصحيح هذا الخطأ قبل الانتقال إلى السطر التالي.
3. يتم تنفيذ جميع أسطر التعليمات البرمجية المصدرية سطرًا تلو الآخر أثناء تنفيذ البرنامج بواسطة المُفسِّر.

المُترجِم:

1. إنشاء البرنامج.
2. يقوم المُترجِم بتحليل ومعالجة جميع أسطر التعليمات البرمجية ويتأكد من صحتها.
3. عند وجود خطأ تظهر رسالة خطأ.
4. إذا لم يكن هناك خطأ، يقوم المُترجِم بتحويل برنامج المصدر إلى لغة الآلة. يتم ربط ملفات التعليمات البرمجية المتعددة ببرنامج واحد قابل للتنفيذ يُعرف باسم ملف EXE.

عملية تصحيح الأخطاء Correction of Errors During the Debugging Process

غالبًا ما يحتوي البرنامج المصدر في نسخته الأولى على العديد من الأخطاء، وتنقسم هذه الأخطاء إلى ثلاثة أنواع:

- **الأخطاء المنطقية** (Logical errors): أخطاء في منطق البرمجة.
- **أخطاء وقت التشغيل** (Runtime errors): أخطاء تحدث أثناء تنفيذ البرنامج.
- **أخطاء تركيب الجُمل** (Syntax errors): أخطاء في تركيب الجُمل البرمجية.

تحدث الأخطاء المنطقية وأخطاء وقت التشغيل فقط عند تنفيذ البرنامج، بينما تحدث أخطاء تركيب الجُمل أثناء الترجمة. يتم تنفيذ البرنامج فقط إذا كان برنامج المصدر لا يحتوي على أخطاء في تركيب الجُمل.

تصحيح أخطاء تركيب الجُمل:

- الخطوة الأولى هي قيام المُترجِم أو المُفسِّر باكتشاف أخطاء تركيب الجُمل، وبالتالي عرض الرسائل التي تشير إلى الخطأ وموقعه. يمكن لبعض هذه الرسائل المساعدة في تشخيص سبب الخطأ.
- الخطوة التالية هي تصحيح الأخطاء في البرنامج.
- في النهاية يتم ترجمة البرنامج الذي تم تصحيحه بشكل صحيح بدون أي رسائل خطأ.

تمرينات

1 ما أوجه القصور في لغة التجميع؟

2 ارسم مخططًا يُظهر الفرق بين عمليّتي ترجمة وتنفيذ البرنامج في كل من المُترجِم والمُفسّر.

3 اكتب ثلاث مزايا للغات البرمجة عالية المستوى.

4 اربط كل لغة برمجية بالتصنيف الذي تنتمي إليه.

1	البايثون (Python)	○	لغات كائنية التوجه
2	بيسيك (BASIC)	○	لغات ذكاء اصطناعي
3	برولوغ (PROLOG)	○	لغات متعددة الأغراض
4	سي (C)	○	لغات برمجة للأطفال

5 اختر الكلمة أو العبارة المناسبة لإكمال الجُمل أدناه (يوجد عبارات لا تنتمي لأي من الفراغات):

المكتبات القياسية	اللغات المُفسّرة	أخطاء في تركيب اللغة	الذاكرة	رابط
البرنامج المصدر	المُترجِم	أخطاء وقت التشغيل	برنامج الكائن	

1. ..... يقبل برنامج المصدر كمُدخل، ليُنتج برنامج مكافئ بلغة الآلة يُسمى .....

2. يتم استخدام ..... بواسطة المُفسِّر بصورة أقل مما يستخدمه المُترجِم.

3. يُعدُّ استخدام ..... مِيزة من حيث تصحيح الأخطاء بصورة فورية، ولكن يتم تنفيذ البرنامج بصورة أبطأ.

4. لا يستطيع المُترجِم تحويل الجُمل التي تشير إلى ..... ولذلك يحتاج إلى ربط هذه الجُمل وتحويلها.

5. يمكن إنشاء الملف التنفيذي إذا لم توجد ..... في برنامج المصدر.

6. يُطلق على الأخطاء التي تحدث أثناء تنفيذ البرنامج اسم ..... .

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

أدوات تطوير البرمجيّات

أدوات وبرامج تطوير البرمجيات

Software Development Tools and Programs

يستخدم المطورون مجموعة كبيرة من الأدوات لتطوير التطبيقات البرمجية، لكل منها مزاياها وعيوبها. تتطلب عملية البرمجة من المطورين التحلي بالكثير من المرونة والإبداع لتحقيق الاستفادة الكاملة من إمكانيات أدوات تطوير البرمجيات المختلفة لتطوير برمجيات عالية الجودة لعملائهم.

تُستخدم أدوات وبرامج تطوير البرمجيات لمساعدة فريق تطوير البرمجيات في مهامهم المُختلفة، بما فيها إنشاء البرامج وتعديلها وصيانتها، بالإضافة إلى تصحيح الأخطاء وتنفيذ مهام البرامج وعمليات التطوير، وتوجد العديد من البرامج المتخصصة التي تُقدم أو تدعم مهامًا محددة في مراحل دورة تطوير البرمجيات.

جدول 1.4: تصنيف أدوات تطوير البرمجيات

أدوات تطوير البرمجيات	الوصف
مُحرِّرات النصوص البرمجية (Code Editors)	تُستخدم لكتابة النصوص البرمجية وإجراء التعديلات عليها.
المُترجِمات والرابطات (Compilers and Linkers)	تترجم البرامج إلى لغة الآلة القابلة للتنفيذ.
مُصحِّحات الأخطاء (Debuggers)	تساعد في تصحيح الأخطاء في البرنامج.
مُنشِئات المشروع (Project Builders)	تتأكد من تجميع جميع الملفات الضرورية ببرنامج نهائي واحد.
أدوات إدارة التعليمات البرمجية (Code Management Tools)	تتأكد من عدم استبدال ملفات البرنامج أو مسحها عن طريق الخطأ عند عمل عدة مبرمجين على البرنامج نفسه، في الوقت نفسه.
بيئة التطوير المتكاملة (Integrated Development Environment -IDE)	تزود المبرمجين ببيئة برمجية متكاملة تتضمن محرر نصوص برمجي ومُترجِم ورابط ومُصحح أخطاء.
المُحلّلات (Profilers)	تقدم هذه البرامج تحليلًا لاحتياجات البرنامج، والتعامل مع وقت المعالج وموارد الذاكرة أثناء التشغيل.
مُحلّلات الشبكة (Network Analyzers)	هذه الأدوات ضرورية عند كتابة برامج لتطبيقات الشبكات على وجه الخصوص.
مُستكشف ومُحلّل قاعدة البيانات (Database Explorer and Analyzer)	تسمح بالتعامل مع قواعد البيانات وتحليل أداء الاستعلامات على قاعدة بيانات محددة.

مُحررات النصوص البرمجية

Code Editors

يسمح محرر النصوص البرمجية بإنشاء وتحرير العديد من ملفات لغة البرمجة المرتبطة، ويُمكنه التعامل مع العديد من لغات البرمجة المختلفة مثل **لغة ترميز النص التشعبي** (HTML) **وسي إس إس** (CSS) **وجافاسكريبت** (JavaScript) **وبي إتش بي** (PHP) **وروبي** (Ruby) **وبايثون** (Python) **وسي** (C) وغيرها. تَستخدم محررات النصوص البرمجية المسافات البادئة والألوان المختلفة لتنسيق النص البرمجي في أقسام البرنامج، مما يجعلها أكثر ملاءمةً لكتابة النصوص البرمجية من مُحررات النصوص العادية مثل مايكروسوفت وورد أو المُفكرة.

هناك العديد من برامج تحرير النصوص البرمجية التي يُمكن للمبرمج اختيارها وفقًا لتفضيلاته. تُعدُّ كفاءة برنامج التحرير في تنفيذ المهمات المطلوبة هي المعيار الأساسي لاختياره. من الأمثلة على هذه المُحررات:

- سابليم تكست (Sublime Text)
- أتوم (Atom)
- **فيجوال ستوديو كود** (Visual Studio Code)
- **اسبريسو** (Espresso)
- **بيئة تطوير متكاملة لبايثون** (Python IDLE)
- **كودا 2** (Coda 2)
- **نوتباد++** (Notepad++)
- **فيم** (Vim)
- **بي بي إيدت** (BBedit)
- **الترا إيدت** (Ultraedit)

ميزات مُحررات النصوص البرمجية

Features of Code Editors

- التحقق من الخطأ.
- الإكمال التلقائي واقتراحات النصوص البرمجية.
- وجود تعليمات برمجية قابلة لإعادة الاستخدام.
- تمييز النصوص البرمجية.
- سهولة التنقل في ملفات النصوص البرمجية والموارد.
- إمكانية إضافة المزيد من الوظائف للمحرر بإضافة الملحقات.

average.py

```
1  # calculate the average class grade
2  total_grades = 0
3  total_students = int(input("Enter the number of students: "))
4  for  n in range (1, total_students + 1):
5      print("Student #", n)
6      student_name = input("Enter the name of the student: ")
7      student_grade = input("Enter the grade of " + student_name + ": ")
8      total_grades = total_grades + float(student_grade)
9  average_grade = total_grades / total_students
10 print("The average grade of the class is ", average_grade)
```

شكل 1.14: برنامج بايثون في مُحرر نصوص برمجية

مزايا وتحديات استخدام مُحررات النصوص البرمجية

Advantages and Challenges of Using Code Editors

المزايا:

- يمكن للمحررات العمل كبديل لمحرر **بيئة التطوير المتكاملة** (Integrated Development Environment – IDE) في مهام البرمجة القياسية، وذلك عند استخدام الامتدادات المناسبة لدعم لغات البرمجة المختلفة.
- أصغر حجمًا وأسرع في التحميل من بيئات التطوير المتكاملة.
- واجهاتها البسيطة تُسهل عملية التركيز على النصوص البرمجية.

التحديات:

- تفتقر المحررات إلى الكثير من ميزات التحرير التي يقتصر توفرها على بيئة التطوير المتكاملة، كخاصية التحرير الذكية على سبيل المثال.
- يحتاج المُستخدِم إلى تهيئة مُحرر النصوص البرمجية بالامتدادات المناسبة قبل البدء باستخدامه لكي يعمل بشكل صحيح.

بيئات التطوير المتكاملة Integrated Development Environments (IDE)

تتوفر بيئات التطوير المتكاملة للمبرمجين مع تطبيقاتها المُدمجة، والتي تتضمن عددًا من أدوات تطوير البرمجيات مثل المُفسِّر الذي يُستخدم أثناء مرحلة إنشاء البرنامج، والمُترجِم الذي يقوم بتجهيز البرنامج ونشره.

لا تقتصر مهمة بيئات التطوير المتكاملة الحديثة على توفير مُترجِم للغة البرمجة، بل تحتوي كذلك على جميع البرامج والأدوات اللازمة للمساعدة في كتابة وتنفيذ النصوص البرمجية. والأهم من ذلك، تتيح هذه البيئات تشخيص الأخطاء في البرامج وتصحيحها. من بين أهم الأدوات المُدمجة في بيئات البرمجة المتكاملة ما يلي:

- مُستكشف الملفات (File Explorer)
- مُحرر النصوص البرمجية (Code Editor)
- مُفسّر (Interpreter)
- مُترجِم (Compiler)
- رابط (Linker)
- مصحح الأخطاء (Debugger)
- عارض الإخراج (Output Viewer)

يجب أن تتضمن بيئات التطوير المتكاملة مُحررًا خاصًا لتسهيل إنشاء كائنات رسومية مثل النماذج والقوائم وصناديق الحوار، وذلك لتزويد المطور بالأدوات المناسبة لإنشاء مجموعات النصوص البرمجية المتعلقة بهذه الكائنات.

ميزات بيئات التطوير المتكاملة Features of IDEs

- الإكمال الذكي للنصوص البرمجية في مُحررها.
- التكامل مع أدوات إدارة النصوص البرمجية للتحكم في الإصدار (Version Control).
- توفير أدوات اختبار متقدمة.
- الربط التلقائي لمكتبات النصوص البرمجية المصدرية.
- توفير الأدوات لأتمتة عملية كتابة النصوص البرمجية ونشرها.

يمكن الحصول على كل ما ذكر أعلاه من خلال واجهة مُستخدِم واحدة.

أمثلة على بيئات التطوير المتكاملة Examples of IDEs

اقتصر دعم معظم بيئات التطوير المتكاملة على لغة برمجة واحدة فقط، حيث كان بناؤها في الماضي من قِبَل شركات البرمجيات أو المؤسسات التي تقوم بتطوير تلك اللغة.

أصبحت معظم مشاريع تطوير البرمجيات في الوقت الحالي تدمج بين تقنيات ولغات برمجة مختلفة، الأمر الذي يتطلب بيئات تطوير متكاملة يمكنها دعم مجموعة واسعة من اللغات.

على سبيل المثال، يدعم **مايكروسوفت فيجوال ستوديو** (Microsoft Visual Studio) عدة لغات كلغة **سي** (C) و **سي بلس بلس** (C++) و **سي شارب** (C#) و**فيجوال بيسك دوت نت** (VB.Net) و**البايثون** (Python) و**روبي** (Ruby) و**نود جي إس** (Node.js) و**جافا سكريبت** (JavaScript) و **سي إس إس** (CSS) و**لغة ترميز النص التشعبي** (HTML) وغيرها. من الأمثلة الأخرى على بيئات التطوير المتكاملة الشائعة: **نت بينز** (NetBeans) و**إكليبس** (Eclipse) و**بيئة التطوير المتكاملة - أتوم** (Atom-IDE) و**إكس كود** (Xcode) و**أندرويد ستوديو** (Android Studio) و**إنتيليج إيديا** (IntelliJ IDEA) و**باي تشارم** (PyCharm). يُستخدم **إكس كود** (Xcode) لتطوير برمجيات تطبيقات الهاتف المحمول للأجهزة التي تعمل بنظام iOS، بينما يُستخدم أندرويد ستوديو لتطوير برامج الأجهزة العاملة بنظام أندرويد.

مزايا وتحديات استخدام بيئات التطوير المتكاملة
Advantages and Challenges of Using Integrated Development Environments (IDE)

المزايا:

- توفر أدوات ذكية للتحليل ولإكمال النصوص البرمجية، وذلك من أجل برمجة أسرع وأخطاء أقل.
- توفر أدوات قوية لتصفح واكتشاف النصوص البرمجية، وتسهل الوصول إلى كافة أجزاء البرنامج، بغض النظر عن حجم المشروع.
- توفر طرقًا متعددة لتصحيح الأخطاء واختبار البرنامج دون الحاجة إلى برامج أخرى أو حتى مغادرة برنامج التحرير.
- تدعم العديد من لغات البرمجة تلقائيًا، وتوفر العديد من أدوات التنقل داخل النصوص البرمجية وتحليلها لتسهيل العمل وزيادة الإنتاجية في المشاريع الكبيرة.

التحديات:

- تحتوي واجهات المُستخدِم في هذه البيئات كثيرًا من الوظائف والميزات التي قد تجعلها معقَّدة وصعبة الاستخدام.
- تتطلب قدرًا جيدًا من التدريب لاستخدامها بشكل صحيح.
- يؤدي وجود كثيرٍ من الوظائف الاختيارية فيها إلى بطء الأداء في بعض الأحيان.

بيئات البرمجيات السحابية Cloud Software Environments

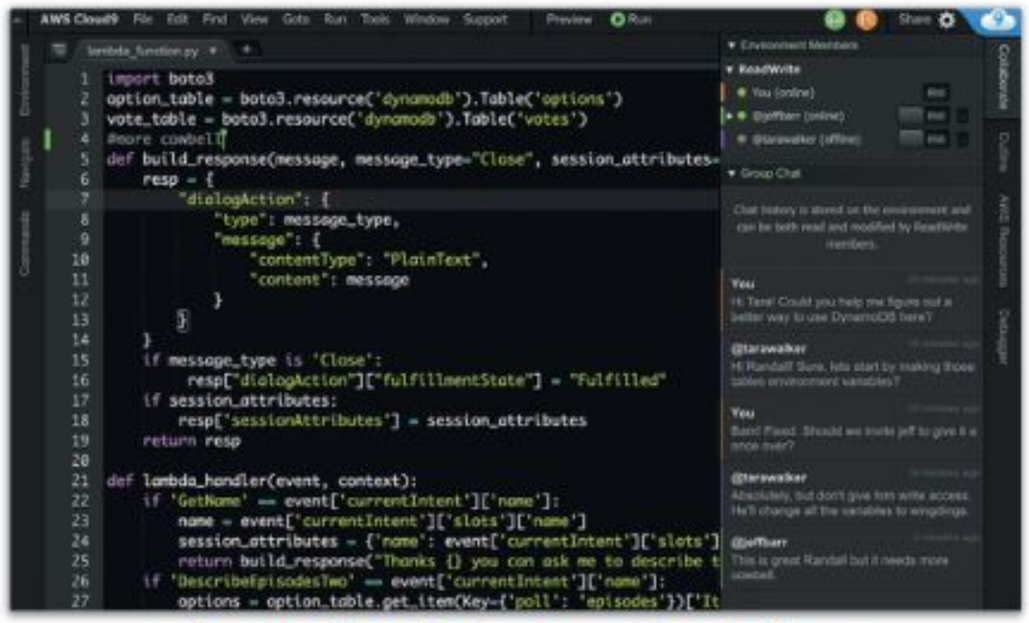
إلى جانب بيئات تطوير البرمجيات التقليدية، توجد بيئات تطوير سحابية قائمة على الويب مثل **أمازون كلاود** 9 (Amazon Cloud9).

توفّر بيئات البرامج السحابية إمكانيّة العمل على المشروع من أي جهاز حاسب ومن أي مكان في العالم، حيث تتوفر بيانات مشروع تطوير البرمجيات سحابيًا.

أحد العوائق الرئيسة أمام استخدام هذه البيئات هو ضرورة توفر اتصال مناسب بالإنترنت للوصول إلى البيانات وتنفيذ العمل.

مزايا استخدام بيئات تطوير البرمجيات السحابية
Advantages of Using Cloud Software Development Environments

- الوصول إلى أدوات تطوير البرمجيات من أي مكان في العالم.
- إمكانية استخدام أي جهاز بواسطة متصفح الويب.
- عدم وجود متطلّبات لتنزيل وتثبيت بيئة البرنامج.
- إمكانية التعاون بين المطورين عن بُعدٍ بسهولة.



شكل 1.15: بيئة أمازون كلاود 9 (Amazon Cloud9)

يقضي المبرمجون معظم وقت البرمجة في الاختبار وتصحيح الأخطاء، لذا فإن تكامل محرر النصوص البرمجية مع المُترجِم والمُصحح يُعدُّ أمرًا مهمًا للغاية، وهو ما تتميز به بيئة التطوير المتكاملة.

الأدوات المتخصصة لمراحل محددة من تطوير البرمجيات

Specialized Tools for Specific Stages of Software Development

يتطلب إنشاء حلول برمجية احترافية العمل ضمن فريق من الخبراء والاستعانة بمجموعة متنوعة من الأدوات التي لا تقتصر على دعم مرحلة البرمجة، بل تمتد إلى دعم عملية إنشاء الحلول البرمجية بكاملها.

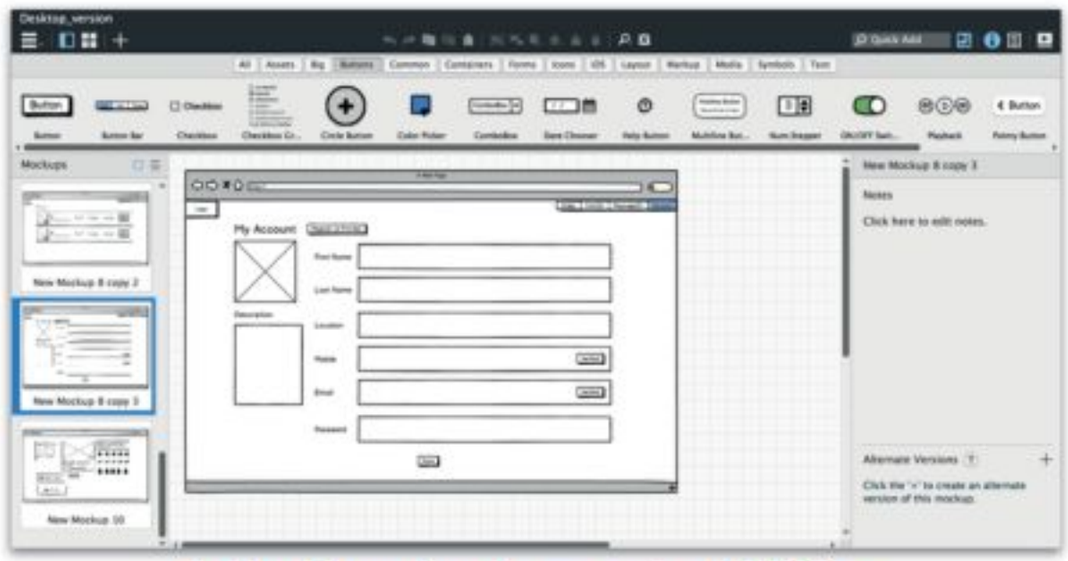
توجد العديد من الأدوات التي يمكن استخدامها أثناء دورة حياة تطوير البرمجيات لمُنتج برمجي، ونظرًا لكثرتها يصعب عرض جميع هذه الأدوات، ولكن تم وصف مجموعة مختارة منها أدناه.

إنشاء النموذج الأولي Prototype Creation

عادةً ما يكون النموذج الأولي للبرنامج عبارة عن مخطّط هيكليّ، أو صورة، أو مجموعة صور تُظهر العناصر الوظيفيّة لتطبيق معين، أو قد يكون موقعًا على الويب يُستخدم لتخطيط التطبيقات أو وظائف موقع الويب وهيكلته.

أمثلة على الأدوات المُستخدمة :

- القلم (Pencil)
- نماذج بلسمق (balsamiq mockups)
- أدوبي إكس دي (Adobe Xd)



شكل 1.16: أداة نماذج بلسمق (balsamiq mockups)

إدارة التحكم في الإصدار - البرنامج المَصدر

Version Control Management - Source Code

يخضع البرنامج المَصدر للعديد من التعديلات أثناء عملية تطويره، وقد يكون التراجع عن خطوات معينة ضروريًا في البرنامج، أو إعادة استخدام البرنامج الذي تم تغييره أو حذفه. عند العمل ضمن فريق من المبرمجين، قد يحتاج اثنان أو أكثر إلى العمل على الملفات نفسها في الوقت نفسه، وإجراء تغييرات على البرنامج نفسه.

تسمى الأداة التي يمكن استخدامها للتحكم في هذه العملية باسم أداة **إدارة التحكم في الإصدار** (Version Control Management) أو **إدارة النصوص البرمجية** (Code Management)، وتتيح هذه الأداة ما يلي:

1. يمكن لأعضاء الفريق المختلفين الوصول إلى البرنامج المَصدر في ذات الوقت دون التعارض بين أعضاء الفريق.
2. يمكن الاحتفاظ بالإصدارات السابقة من ملفات البرامج للرجوع إليها عند حدوث بعض المشكلات.

يَستخدِم التحكم في الإصدار ما يشبه **المستودع** (Repository)، ومهمّته تسجيل جميع التغييرات التي يتم إجراؤها، وإنشاء نسخة عمل من ملفات برنامج المشروع تُسمى أحيانًا **نسخة التحقق** (Checkout Copy) عندما يريد المبرمج العمل على البرنامج. تتم الموافقة على جميع التغييرات التي تم إجراؤها على البرنامج بواسطة برنامج إدارة التحكم في الإصدار عندما يتم حفظها في المستودع.

أمثلة على الأدوات المُستخدمة:

- قيت (Git)
- سبفيرجن (Subversion)
- ميركوريال (Mercurial)
- أزور ديف أوبس (Azure DevOps)
- ديف ميرق (DiffMerge)

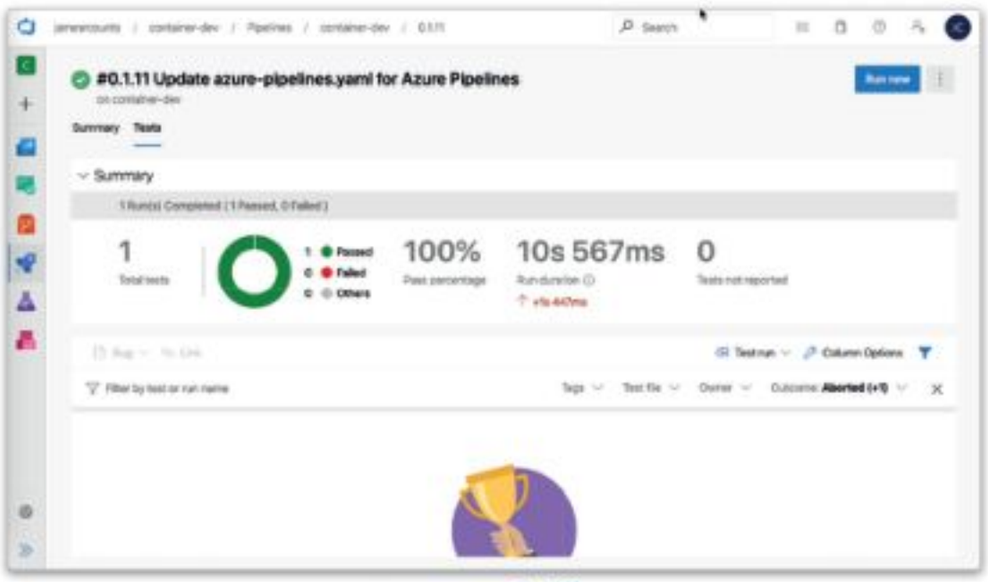
شكل 1.17: إدارة التحكم في الإصدار

نشر البرنامج Code Deployment

كان نشر البرامج أمرًا يسيرًا في الماضي، حيث يتم وضع البرنامج المُترجَم، أي الجاهز للعمل على قرص مدمج ليكون جاهزًا للاستخدام. ولكن مع ظهور الإنترنت أصبح نشر التطبيقات عبر الإنترنت ضروريًّا، وذلك كبرامج قابلة للتثبيت من خلال متاجر التطبيقات، أو مباشرةً كتطبيقات ويب، وأدى ذلك إلى وجود حاجة لتطوير البرامج والأدوات الخاصة بنشر التطبيقات على الويب.

أمثلة على الأدوات المُستخدمة:

- تيم سيتي (TeamCity)
- مدير نشر قوقل كلاود (Google Cloud Deployment Manager)
- قيت لاب (GitLab)
- جينكينز (Jenkins)
- ناشر الكود AWS (AWS CodeDeploy)
- أزور ديف أوبس (Azure DevOps)



شكل 1.18: نشر البرنامج

معلومة

ميزة التفريعة (Branching) مفيدة جدًا في التحكم في الإصدار، فهي تعني إمكانية نسخ برنامج المشروع كمشروع موازي جديد يسمح بالاختبار أو بإجراء التعديلات لإنشاء إصدار محدَّث أو جديد من التطبيق، ويمكن لاحقًا نقل أجزاء من البرنامج الجديد إلى المشروع الأصلي لاستخدامها به أيضًا.

الاختبار Testing

لا يقتصر الاختبار على تصحيح الأخطاء البرمجية في البرنامج فحسب، بل يشمل أيضًا اختبار تشغيل البرنامج وفعالية استخدامه من قِبَل عدد كبير من المُستخدِمين، بالإضافة إلى إجراء اختبارات الأمان وغيرها من الاختبارات.

أمثلة على الأدوات المُستخدمة:

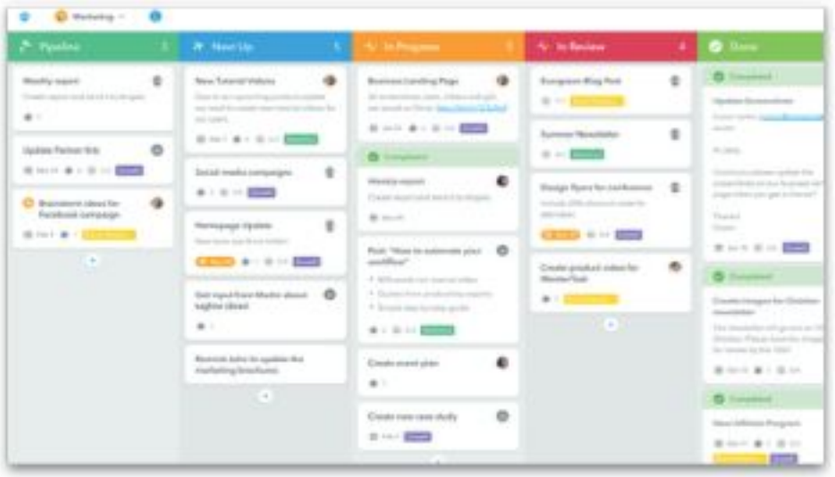
- اباتشي جميتر (Apache JMeter)
- قوست لاب (Ghostlab)
- سيلينيوم (Selenium)
- استوديو اختبار Telerik (Telerik Test Studio)
- أزور ديف أوبس (Azure DevOps)
- ايرون واساب (IronWASP)
- زِد أتاك بروكسي (Zed Attack Proxy)
- وابيتي (Wapiti)

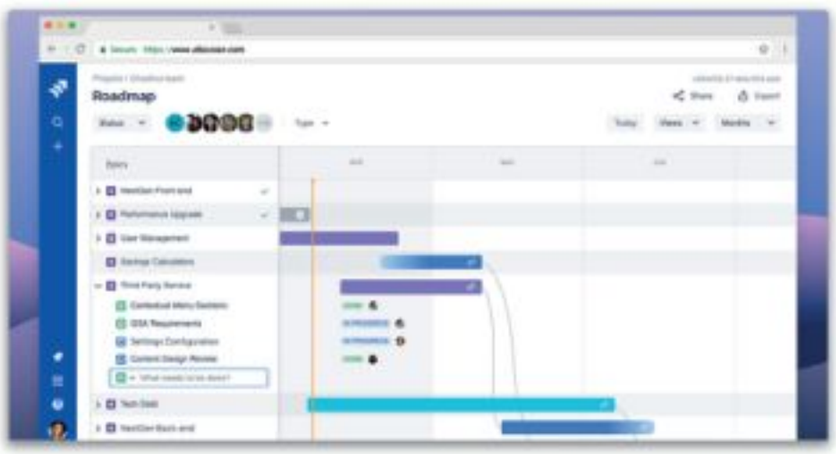
إدارة المشروع والتعاون وتتبع المشكلات Project Management, Collaboration and Issue Tracking

كما تعلَّمت سابقًا، يتطلب الحصول على منتجٍ ناجح أن يبقى فريق العمل بكافة أعضائه الحاليين وكذلك المستقبليين على اطِّلاع على عملية التطوير بأكملها ومشاركة المعرفة، ويبرز هنا دور عملية إدارة المشروع بشكل خاص.

أمثلة على الأدوات المُستخدمة:

- مايكروسوفت تيمز (Microsoft Teams) للتعاون والتواصل.
- سكرم تريلو (Scrum Trello) للتخطيط السريع والتتبع.
- جيرا (Jira) لتتبع متطلَبات المشاريع وإدارتها.
- مايستر تاسك (MeisterTask) لإدارة المهام.
- سلاك (Slack) للتعاون والتواصل.
- بيسكامب (Basecamp) لإدارة المشاريع والتواصل مع العملاء.
- أزور ديف أوبس (Azure DevOps) لإدارة دورة حياة التطبيق (Application Life Cycle Management - ALM).





شكل 1.19: أمثلة على أدوات إدارة المشروع والتعاون وتتبع المشكلات

استخدام أدوات التطوير لتقديم حلول مختلفة
Using Development Tools to Provide Different Solutions

تعتمد فرق التطوير على الأدوات التي تم التطرق إليها سابقًا لإنتاج مجموعة واسعة من حلول تقنية المعلومات، وتُستخدم العديد منها حاليًا لإنشاء تطبيقات من أنواع مختلفة مثل:

- التطبيقات العامة.
- تطبيقات الويب.
- الأنظمة المُدمجة.
- تطبيقات الهواتف الذكية.

إنشاء تطبيق ويب Building a Web Application

تطبيق الويب هو برنامج تفاعلي يتم إنشاؤه باستخدام تقنيات الويب **بلغة ترميز النص التشعبي** (HTML) **وسي إس إس** (CSS) **وجافاسكريبت** (JavaScript)، ويتم تخزين البيانات في هذا التطبيق على خوادم قواعد البيانات. يستخدم هذا التطبيق لتنفيذ المهام عبر الإنترنت من طرف مستخدِمي التطبيق.

مراحل إنشاء تطبيق الويب Stages of Building a Web Application

1. مرحلة تصور الأفكار:

يجب تحديد الأهداف والفكرة الرئيسة للتطبيق قبل البدء بإنشاء تطبيق للويب.

2. بحوث وتحليل السوق:

يجب أن تتم عملية البحث وتحليل السوق لمعرفة:

- ما حاجة المُستهلك المُستهدف إلى هذا المُنتج أو الخدمة؟
- هل هناك مُنتج أو خدمة مماثلة متوفرة؟

3. تحديد مهمات ووظائف تطبيقات الويب:

يجب أن يتم تحديد الوظائف أو المهمات التي تقدم حلولًا لمشاكل السوق المستهدفة بالتطبيق.

4. النمذجة الأولية / تجهيز المُخطط:

يُعنى هذا الموضوع بتصميم وإنشاء المخططات لتطبيق الويب، وبإعداد النماذج الأولية التي تساهم في تشكيل المخطط التنظيمي للتطبيق وتتيح اختبار وظائف التطبيق بطريقة تفاعلية.

5. مرحلة التحقق:

يتم في هذه المرحلة جمع الآراء والتعليقات حول التصميم من الأطراف ذات العلاقة ومن المُستخدِمين المحتملين.

6. التخطيط الهيكلي وإنشاء قاعدة بيانات:

يتم في هذه المرحلة تحديد البيانات التي يحتاجها المبرمجون والمُستخدِمون، والأدوات المستخدمة لإنشاء قاعدة البيانات المطلوبة لتطبيق الويب.

هناك العديد من أدوات تصميم قواعد البيانات التي تُستخدم لأغراضٍ مختلفة. تحدد طبيعة البرنامج وطريقة نشر الحل البرمجي اختيار أداة محددة. من الأمثلة على الأدوات المستخدمة في تصميم قواعد البيانات وبنائها، ما يلي:

- ماي إس كيو إل (MySQL)
- أزور إس كيو إل (Azure SQL)
- إس كيو إل سيرفر (SQL Server)
- مونجودب (MongoDB)
- أمازون دينامو دي بي (Amazon DynamoDB)
- فايربيس (Firebase)

7. إنشاء الواجهة الأمامية (جزء البرنامج لدى العميل):

الواجهة الأمامية هي العنصر المرئي لتطبيق الويب، وهي تمثل واجهة النظام التي يراها المُستخدِم ويتفاعل مع النظام من خلالها. تتضمن أمثلة الأدوات المستخدمة لإنشاء واجهة مُستخدِم الويب ما يلي:

- جي كويري (jQuery)
- فيو جي إس (Vue.js)
- ريكت جس (Reactjs)
- أنقيولر (Angular)
- جانغو (Django)

8. إنشاء الواجهة الخلفية (الجزء الخلفي للبرنامج لدى الخادم):

تُستخدم الواجهة الخلفية لإدارة بيانات البرنامج، وتشير إلى قواعد البيانات والخوادم بالإضافة إلى جميع أجزاء النظام الأخرى غير المرئية للمُستخدِم داخل تطبيق الويب.

يتضمن إنشاء الواجهة الخلفية كتابة البرنامج الأساسي الذي يوفر وظائف التطبيق، بالإضافة إلى إعداد قاعدة البيانات والشبكات والتحقق من التكامل بين الأنظمة الفرعية المختلفة، ويتم أيضًا إعطاء أهمية خاصة لاعتبارات الأمان والأداء. من الأمثلة على الأدوات المستخدمة في إنشاء الواجهة الخلفية:

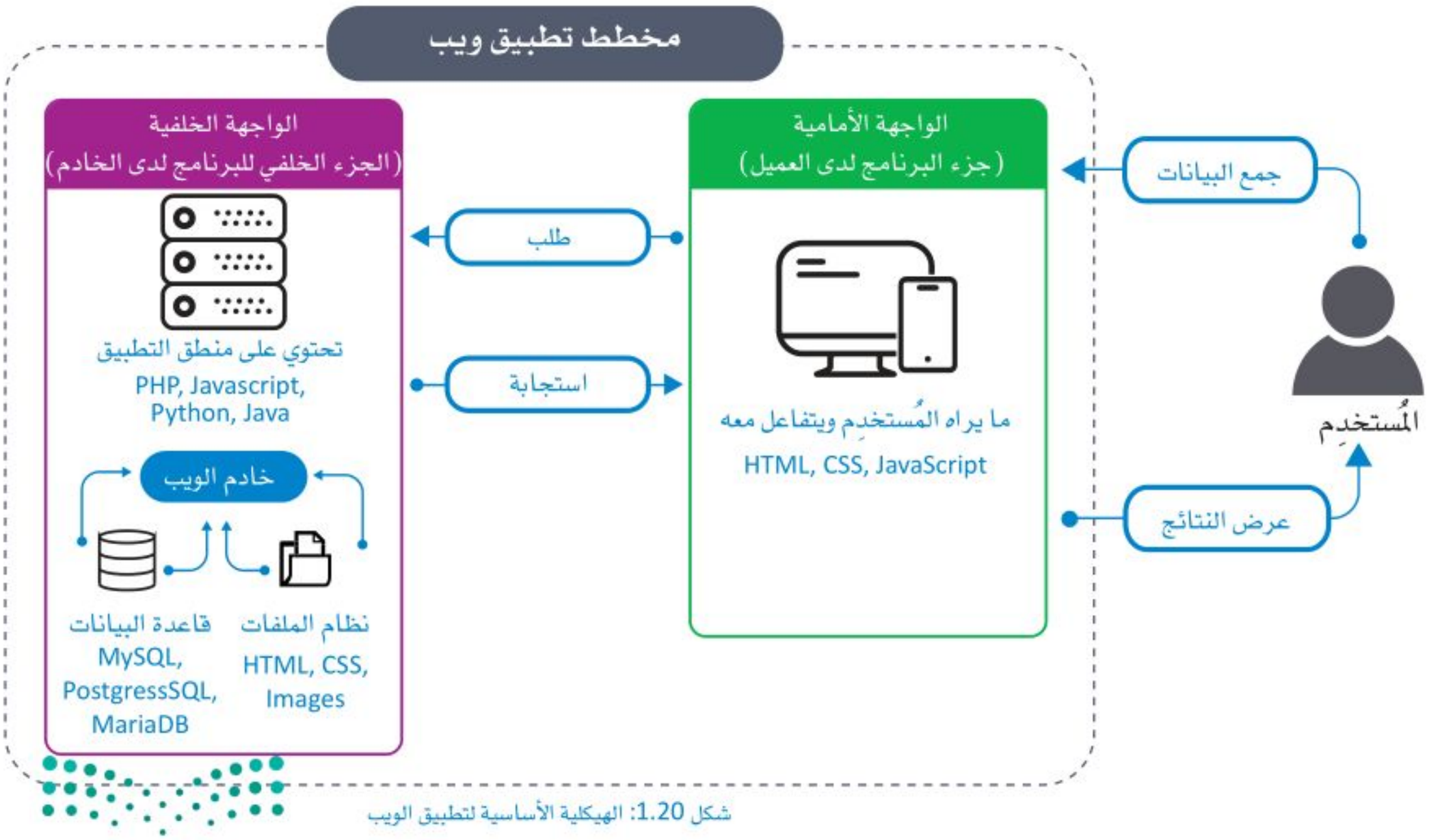
- اكسبرس جي إس (Express JS)
- فلاسك (Flask)
- إيه إس بي دوت نت (ASP.NET)
- لارافيل (Laravel)
- روبي أون ريلز (Ruby on Rails)
- سبرنغ بوت (Spring Boot)

9. استضافة تطبيق الويب :

يجب توفر مزوِّد استضافة ويب لتشغيل تطبيق الويب على خادم معين، وتتوفر خدمات الاستضافة حسب الحاجة، وتتراوح بين الخدمات البسيطة غير المكلّفة، وخدمات الحوسبة السحابية الكبيرة التي تسمح بالتوسع حال ازدياد عدد مُستخدِمي التطبيق وتزايد السعة المطلوبة للخدمات.

جدول 1.5: مزودو خدمات استضافة الويب

الأنواع	أمثلة
مزودو الاستضافة	• بلوهوست (Bluehost) • هوست قيتور (HostGator) • قودادي (GoDaddy) • راك سبيس (Rackspace)
مزودو الخدمات السحابية	• آي بي إم الحوسبة السحابية (IBM Cloud) • مايكروسوفت أزور (Microsoft Azure) • خدمات أمازون ويب (Amazon Web Services) • منصة قوقل السحابية (Google Cloud Platform) • سحابة علي بابا (Alibaba Cloud)

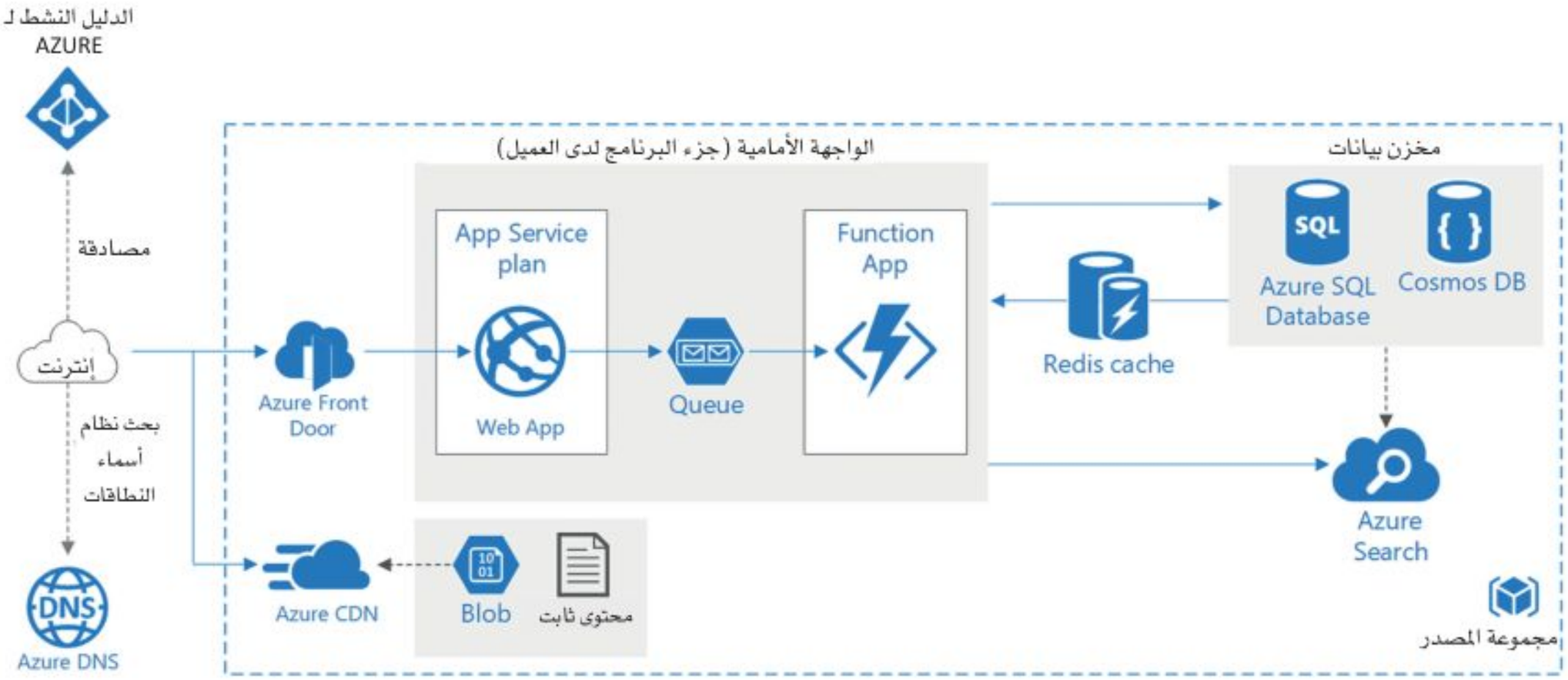


شكل 1.20: الهيكلية الأساسية لتطبيق الويب

هيكلية التطبيقات السحابية The Cloud-Ready Application Architecture

لقد أصبح تطوير ونشر تطبيقات الويب القائمة على التخزين السحابي محبّذًا كمجموعة من الخدمات السحابية، وتضمن هذه العملية إنشاء هياكل البيانات، ثم إنشاء الخدمات التي يتم دمجها لتشكيل النظام المتكامل.

يوضِّح المُخطط التالي كيفية إنشاء تطبيق ويب عالي الأداء وقابل للتوسع باستخدام خدمات **مايكروسوفت أزور** (Microsoft Azure)، ويعمل معظم مزودو خدمات الحوسبة السحابية بالمبدأ نفسه أيضًا.



شكل 1.21: إنشاء تطبيق ويب باستخدام مايكروسوفت أزور (Microsoft Azure)

أهم النقاط التي يجب مراعاتها عند استخدام هيكلية التطبيقات السحابية:

- تصميم التطبيق كمجموعة من الخدمات.
- الفصل بين معايير البيانات والأمان والأداء.
- أخذ متطلّبات الاتصال عبر الشبكات بين مكونات التطبيق في الاعتبار.
- قابلية التوسع في التصميم.
- يجب أن يكون أمان النظام جزءًا أساسيًا من التطبيق وليس شيئًا يتم التخطيط له لاحقًا.
- مراعاة اختيار مراكز البيانات المناسبة لمواقع وأماكن وجود مُستخدِمي النظام.

إنشاء تطبيق للهواتف الذكية Building an Application for Smartphones

تتشابه خطوات إنشاء تطبيق للهاتف الذكي مع تلك الخاصة بتطبيق الويب، ولكن مع بعض الاعتبارات الخاصة، حيث يتم استخدام تطبيق الهاتف المحمول على جهاز هاتف جوال ذي شاشة صغيرة. فكما هو معلوم سيُستخدم التطبيق أثناء التنقل، مما يشير إلى الأهمية الخاصة لواجهة التطبيق، حيث يجب أن يكون المُستخدِم قادرًا على ضبط حجم الشاشة وأن يكون الوصول إلى المعلومات المهمة واضحًا وسهلًا ، ومن المهم أيضًا ملاحظة أن اختلاف الأجهزة يخلق الحاجة إلى إنشاء تطبيقات متجاوبة ومناسبة لكل أنواع الشاشات.

تدعم كل من المنصتين الرئيسيتين للهاتف المحمول **آي أواس** (iOS) و**أندرويد** (Android) مجموعة من التقنيات المختلفة التي تتشابه في بعض الأوجه أيضًا. فعلى سبيل المثال، يوصي مطورو نظام **آي أو إس** (iOS) باستخدام **إكس كود** (Xcode) و**سويفت** (Swift) لتطوير التطبيقات، بينما يوصي مطورو نظام أندرويد باستخدام **أندرويد ستوديو** (Android Studio) و**جافا** (Java).

تسمح كل من هذه البيئات بإنشاء تطبيق نهائي جاهز للنشر في متجر التطبيقات الخاص بتلك البيئة فقط، وتقدم بعض البيئات الأخرى إمكانية نشر التطبيق في متاجر متعددة.

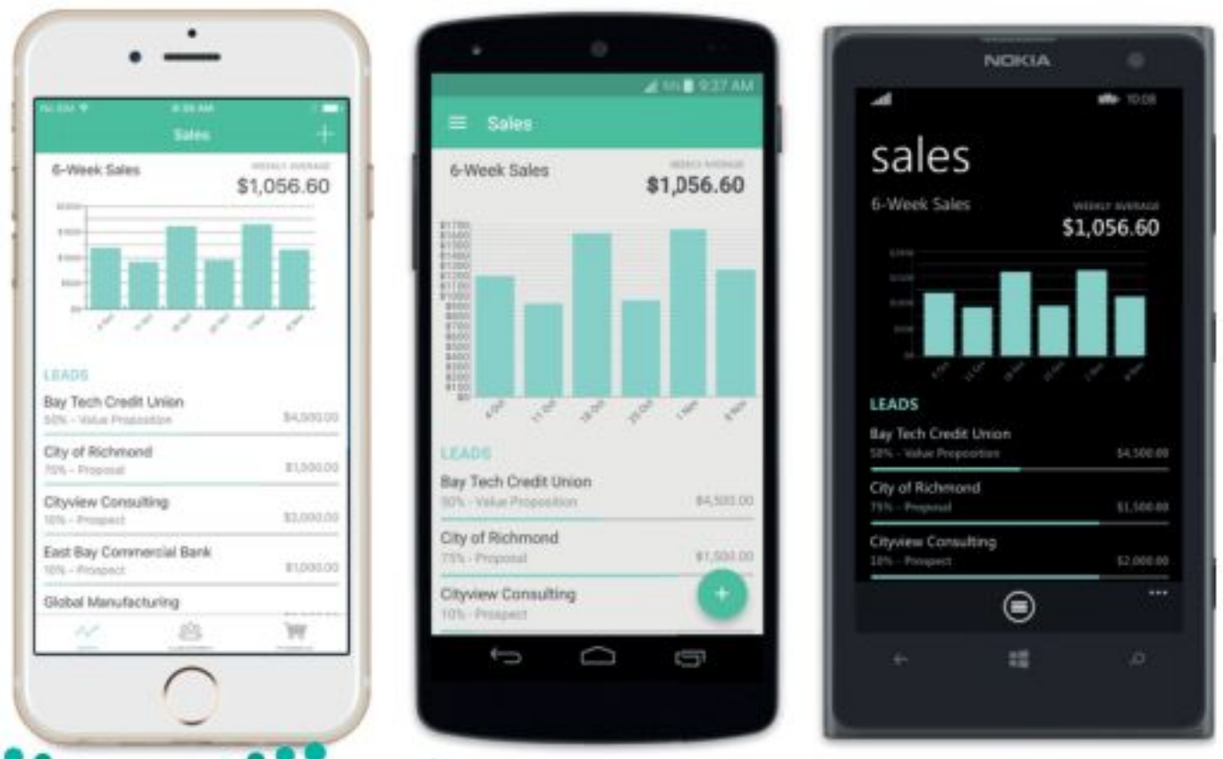
يمكن تطوير تطبيق بحيث يعمل في بيئات برامج مختلفة باستخدام الأدوات التالية:

- **أيونيك** (Ionic)
- **زامارين** (Xamarin)
- **كوتلن** (Kotlin)
- **رياكت نيتيف** (React Native)

يُعدُّ اختبار تطبيقات الأجهزة المحمولة تحديًا كبيرًا، حيث يصعب على المبرمج وشركات تطوير البرمجيات توفير جميع أنواع الأجهزة المحمولة الموجودة في السوق لإجراء الاختبار عليها. ولهذا السبب، توجد خدمات عبر الإنترنت تقدم عمليات محاكاة لمجموعة واسعة من الأجهزة المحمولة تُتيح محاكاة اختبار توافق التطبيق مع الأجهزة المختلفة.

أمثلة على بعض أدوات اختبارات التطبيق:

- **سحابة اختبار زامارين** (Xamarin Test Cloud)
- **بروزر ستاك** (BrowserStack)
- **معمل اختبار فايربيس** (Firebase Test Lab)

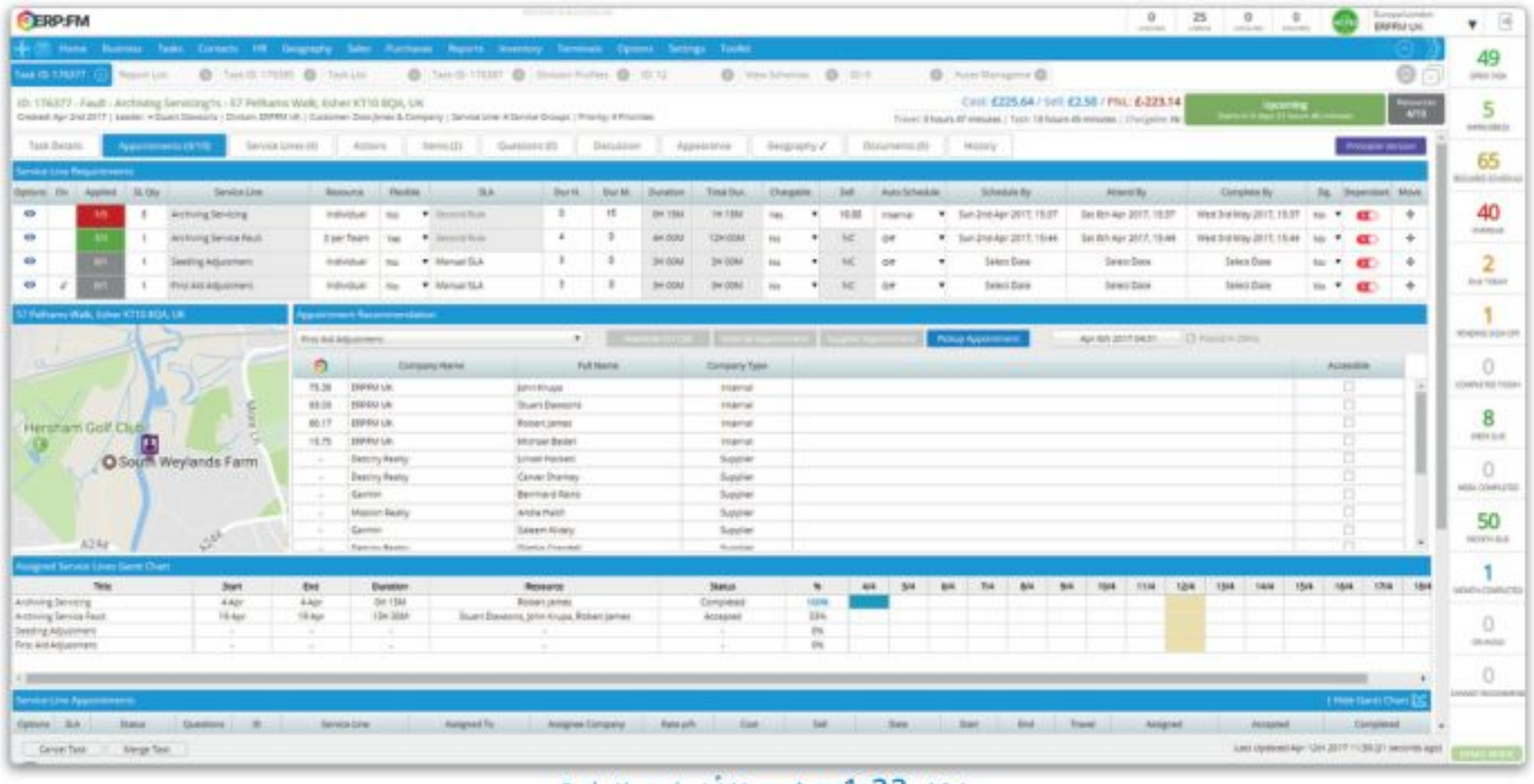


شكل 1.22: تطبيق للهاتف المحمول على أجهزة مختلفة

إنشاء تطبيق للأغراض العامة Building a General-Purpose Application

إن برامج الأغراض العامة هي نوع من التطبيقات التي يُمكن استخدامها لأداء العديد من المهام، مثل البرامج المكتبية التقليدية كمعالجات النصوص، أو برامج تصميم المخططات البيانية، أو تطبيقات الأعمال الخاصة **بتخطيط موارد المؤسسات** (Enterprise Resource Planning – ERP)، أو برامج **إدارة علاقات العملاء** (Customer Relationship Management – CRM).

على الرغم من التركيز الواسع لتقنيات تطوير البرمجيات الجديدة على الويب وتطبيقات الهواتف المحمولة، إلا أن التطبيقات التقليدية للأغراض العامة لا تزال تحتفظ بأهميتها، حيث يعتمد تطوير مثل هذه التطبيقات على مكتبات نصوص برمجية جاهزة وقابلة لإعادة الاستخدام، وبشكلٍ خاص على مكونات واجهة المُستخدِم وأدوات إعداد التقارير.



شكل 1.23: تطبيق للأغراض العامة

إنشاء تطبيق مُدمج Building an Embedded Application

النظام المُدمج هو حاسب خاص بنظام تشغيل يعمل في **الوقت الفعلي** (Real-Time)، ولا يحتوي على واجهة مُستخدِم غالبًا، ويتعامل البرنامج الموجود على النظام المُدمج مع المستشعرات والمشغلات وآليات تبادل البيانات السلكية واللاسلكية، ويجب أن تكون هذه البرامج موثوقة وآمنة وسريعة. تتطلب هذه التطبيقات أنظمة تشغيل في الوقت الفعلي مثل **أر تي لينكس** (RTLinux) **ووينـدوز 10 آي أو تي** (Windows 10 IoT) **وكيو إن إكس** (QNX)، وذلك بالإضافة إلى لغات برمجة مُحسّنة لمعالجة البيانات والاتصال بالشبكة. من الأمثلة على الأنظمة المدمجة:

إشارات المرور، وأجهزة إنذار الحريق، وأنظمة الأمن المنزلية.

شكل 1.24: نظام مُدمج

يمكن برمجة الأنظمة المُدمجة باستخدام لغات البرمجة التالية:

- لغة **التجميع** (Assembly) وهي صعبة وغير مناسبة للاستخدام العملي.
- لغة **سي** (C) و لغة **سي للأنظمة المضمنة** (Embedded C) وnesC ولغة رست (Rust).
- اللغات كائنية التوجه مثل سي شارب (C#) وسي بلس بلس (C++) وجافا (Java).

تمرينات

1

اختر الإجابة الصحيحة:		
1. مُنشِئات المشروع:	تتأكد من أن جميع الملفات التي تحددها سيتم تجميعها وربطها في برنامج واحد نهائي.	○
	تترجم البرنامج إلى نصوص برمجية قابلة للتنفيذ على الجهاز.	○
	ضرورية في حالة إنشاء برامج متخصصة تتعلق بالشبكات.	○
2.أدوات إدارة التعليمات البرمجية:	تساعد على تصحيح الأخطاء في البرنامج.	○
	تتعامل مع قواعد البيانات وتحلل أداء الاستعلامات في بعض قواعد البيانات.	○
	تتأكد من عدم مسح أو استبدال ملفات البرنامج عن طريق الخطأ عندما يعمل عدة مبرمجين على البرنامج بشكل متزامن.	○
3. أدوات التحليل:	تقوم بتقديم أو دعم مهمة محددة في أي حالة من دورة التطوير أو البرمجة.	○
	تقدم تحليلًا لاحتياجات البرنامج، وكيفية التعامل مع وقت المعالج وموارد الذاكرة أثناء التشغيل.	○
	هي أجهزة حاسب خاصة بأنظمة تشغيل في الوقت الفعلي وعادة ما تكون بدون واجهة مُستخدِم.	○

2 اختر الكلمة المناسبة لإكمال الجمل التالية:

معالجات النصوص	نموذج أولي للبرنامج	التحكم في الإصدار	تطبيق الويب	
بيئة التطوير المتكاملة	مُحررات النصوص البرمجية	أغراض عامة		

1. ..... تساعد على الكتابة وإجراء تغييرات على البرنامج.
2. لا تُعدُّ ..... مناسبة للبرمجة، حيث لا تسمح بالتنسيق السهل للبرنامج في كتل النصوص البرمجية.
3. ..... يتضمن محرر النصوص البرمجية، والمُترجِم، والرابط، ومُصحح الأخطاء.
4. ..... هو مُخطط هيكلي أو صورة أو مجموعة صور تُظهر العناصر الوظيفيّة لتطبيق معين.
5. أدوات ..... تضمن أن العمل يتكامل بشكل متزامن من قبل أعضاء الفريق المختلفين.
6. ..... هو برنامج تفاعلي يتم إنشاؤه باستخدام تقنيات الويب، ويتم تخزين البيانات في هذا التطبيق على خوادم قواعد البيانات، ويستخدم لتنفيذ المهام عبر الإنترنت.
7. ..... برامج تستخدم للأغراض العامة لأداء مجموعة واسعة من المهام.

3 طابق ما يلي:

1	محرر النص	العناصر المرئية لتطبيق الويب، وهي الواجهة بين المُستخدِم والنظام.
2	إدارة التحكم في الإصدار	يتيح الاحتفاظ بالإصدارات السابقة من ملفات البرامج للرجوع إليها عند حدوث مشاكل.
3	بيئة التطوير المتكاملة	نوع من البرامج المستخدمة لتعديل الملفات النصية.
4	الواجهة الأمامية (جزء البرنامج لدى العميل)	يدير البيانات وقواعد البيانات والخوادم وجميع المكونات التي لا يستطيع المُستخدِم رؤيتها داخل تطبيق الويب.
5	الواجهة الخلفية (الجزء الخلفي للبرنامج لدى الخادم)	يحتوي على جميع البرامج والأدوات اللازمة لكتابة البرامج وتنفيذها وتشخيص المشاكل وإصلاحها.

4 ما أهم النقاط التي يجب مراعاتها عند استخدام هيكلية التطبيقات السحابية؟

5 ما المقصود ببرامج الأغراض العامة؟ أعطِ بعض الأمثلة.

6 عدِّد أربعة أنواع من أدوات تطوير البرمجيات.

7 أعطِ ثلاثة أمثلة لمُحررات النصوص البرمجية.

8 ضع قائمة بالخطوات الأساسية لإنشاء تطبيق للويب.

9 دَوِّنْ ثلاث ميزات أساسية لبيئة التطوير المتكاملة.

المشروع

1

طورت المملكة العربية السعودية رؤية مستقبلية تستند إلى ثلاثة محاور أساسية: مجتمع نابض بالحياة، واقتصاد مزدهر، وأمة طموحة. تُعدُّ خطة رؤية 2030 الخطوة الأولى نحو تحقيق التطلعات الاقتصادية للمملكة العربية السعودية وتحسين حياة المواطنين.

2

افترض أنك تريد إنشاء تطبيق للهاتف المحمول يوفر معلومات حول المشاريع السياحية في رؤية 2030. وبشكل أكثر تحديدًا، يهدف التطبيق إلى مساعدة كبار السن الذين يعانون من مشاكل في الرؤية أو ارتجاف اليدين (شلل الرعاش) لتصفح المعلومات حول المشاريع العملاقة المتضمنة في رؤية 2030 مثل أمالا، ونيوم، والقدية.

3

ابحث عن معلومات وصور حول هذه المشاريع.

ستستخدم هذه المعلومات عند إنشاء التطبيق.

4

أنشئ مُلخصًا لدورة حياة تطوير البرمجيات الخاصة بالتطبيق، مع استعراض ما ستفعله في كل مرحلة من مراحل هذه الدورة.

في الختام أنشئ عرضًا توضيحيًا لمشروعك.

ماذا تعلّمت

- التفريق بين مراحل دورة حياة تطوير البرمجيات (SDLC).
- تصنيف مزايا وتحدّيات كلٍ من: منهجية الشلال، ومنهجية التطوير السريع للتطبيقات، ومنهجية التطوير الرشيقة (Agile).
- وصف لغات البرمجة المختلفة، وتاريخها، وتصنيفاتها، ومجالات استخدامها.
- التعرف على كيفية فهم الحاسب للغات البرمجة، والتعامل مع أخطائها من خلال المُترجِم أو المُفسِّر.
- التعرف على أدوات تطوير البرمجيات المختلفة، واستخداماتها في مختلف مراحل تطوير البرمجيات وانتاج الحلول البرمجية المختلفة.

المصطلحات الرئيسة

دورة حياة	Lifecycle
رابط	Linker
لغة الآلة	Machine language
صيانة	Maintenance
تطبيق للهاتف المحمول	Mobile Application
التطوير السريع للتطبيقات	Rapid Application Development (RAD)
دورة حياة تطوير البرمجيات	Software Development Life Cycle (SDLC)
أداة تطوير البرمجيات	Software Development Tool
منهجيات تطوير البرمجيات	Software Development Methodologies
الاختبار	Testing
التحكم في الإصدار / إدارة البرنامج المصدر	Version Control/Source Code Management
منهجية الشلال	Waterfall Methodology
تطبيق ويب	Web Application

منهجية التطوير الرشيقة	Agile Methodology
لغة التجميع	Assembly Language
محرر النصوص البرمجية	Code Editor
المُترجِم	Compiler
تطوير	Development
نظام مُدمج	Embedded System
التقييم	Evaluation
برنامج قابل للتنفيذ	Executable Program
لغة الجيل الرابع	Fourth-generation Language
تطبيق للأغراض العامة	General-purpose Application
لغة برمجة عالية المستوى	High-level Programming Language
بيئة التطوير المتكاملة	Integrated Development Environment (IDE)
المُفسّر	Interpreter

2. النمذجة الأولية

سيتمكن الطالب في هذه الوحدة من المقارنة بين الطرق المختلفة لجمع متطلبات المُستخدِم لنظام جديد. وسيتعرف على مُخطَّط سير العمل، وسيتعلم كيفية تصميمه على شكل مُخطَّط يظهر العمليات الخاصة به. وفي الختام سيتعرف على كيفية إنشاء نموذج أولي لتطبيق للهاتف المحمول باستخدام برنامج بنسل بروجكت (Pencil Project).

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

> يُعرِّف طرق جمع البيانات لتحليل النظام.
> يُعرِّف أنواع المُخطَّطات المُستخدمة في مرحلة التحليل.
> يُميِّز المقصود بالتحليل.
> يُصنِّف المتطلبات الوظيفية وغير الوظيفية للنظام.
> يسرِد أساليب جمع البيانات.
> يصف مُخطَّطات سير العمل.
> يستخدِم برنامج بنسل بروجكت لتصميم مُخطَّط سير العمل.
> يشرح التفاعل بين الإنسان والحاسب (HCI).
> يوضِّح الفرق بين تصميم واجهة المُستخدِم (UI) وتجربة المُستخدِم (UX).
> يصف الوظائف والاستخدامات الأساسية للأجهزة المحمولة وأجهزة الحاسب المكتبية.
> يُعرِّف مزايا الأجهزة المحمولة وأجهزة الحاسب المكتبية وعيوبها.
> يُصمِّم نموذجًا أوَليًّا لتطبيق هاتف محمول.

الأدوات

> بنسل بروجكت (Pencil Project)

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

التحليل

مرحلة التحليل في دورة حياة تطوير البرمجيات

Analysis Phase of the SDLC

كما تم ذكره في الدرس السابق، يُمكن تقسيم دورة حياة تطوير البرمجيات إلى خمس مراحل: أولها مرحلة التحليل، وفيها يجتمع مُحلل النظام مع المُستخدِمين لتحديد ما يحتاجون إليه بدقة، ولإجراء دراسة الجدوى، ويتم أيضًا في هذه المرحلة البحث عن تفاصيل النظام المطلوب أو أي متطلبات قد يطرحها العميل.

تنقسم هذه المتطلبات إلى قسمين:

1. **المتطلبات الوظيفية.**
2. **المتطلبات غير الوظيفية.**

المتطلبات الوظيفية Functional Requirements

تُعرَّف المتطلبات الوظيفية بأنها ما يُحدِّد بصورة أساسية الوظائف التي يجب على النظام القيام بها، ومن أكثر المتطلبات الوظيفية شيوعًا ما يلي:

- متطلبات قواعد العمل والوظائف الإدارية.
- وظائف النظام الخاصة بتصحيح المُعامَلات وتعديلها وإلغائها.
- مستويات المصادقة والصلاحيات في النظام.
- الواجهات الخارجية للنظام.
- متطلبات إصدار شهادات الاعتماد للنظام.
- متطلبات تقديم التقارير.

من الأمثلة على المتطلبات الوظيفية:

1. إرسال النظام بريدًا إلكترونيًا للتأكيد عند تقديم طلب.
2. إتاحة النظام إمكانيّة تحقّق المُستخدِمين من الحسابات باستخدام أرقام هواتفهم.
3. سماح النظام لزوار المدوَّنة بالتسجيل في النشرة الإخبارية من خلال بريدهم الإلكتروني.

المتطلبات غير الوظيفية Non-Functional Requirements

متطلبات البرامج غير الوظيفية هي مجموعة قيود أو معايير تحدد كيفية تصرف نظام البرنامج أو أدائه أو تشغيله بما يتجاوز متطلباته الوظيفية الأساسية، ومن أكثر المتطلبات غير الوظيفية شيوعًا ما يلي:

- **الأداء**: المتطلبات المتعلقة بالسرعة والاستجابة وقابلية تطوير نظام البرنامج، بما في ذلك زمن الاستجابة، واستخدام الموارد، وسرعة نقل البيانات.
- **الأمان**: متطلبات حماية البيانات الحساسة مثل: مصادقة المُستخدِم والتشفير والتحكم في الوصول.
- **سهولة الاستخدام**: متطلبات سهولة الاستخدام وتجربة المُستخدِم مثل: قابلية التصفح وتصميم واجهة المُستخدِم وإمكانية الوصول.
- **الموثوقية**: المتطلبات المتعلقة بتوافر نظام البرنامج واستقراره مثل: معالجة الأخطاء والقدرة التلقائية على الاستمرار في العمل عند حدوث خلل، وكذلك الاسترداد التلقائي للمعلومات.
- **التوافق**: المتطلبات المتعلقة بتوافق نظام البرنامج مع أنظمة التشغيل أو الأنظمة الأساسية أو الأجهزة الأخرى مثل: توافق المتصفح وتوافق الأجهزة المحمولة وقابلية التشغيل في بيئات تشغيل مختلفة.

من الأمثلة على المتطلبات غير الوظيفية :

1. **قدرة النظام على استعادة البيانات غير المحفوظة عند حدوث انقطاع مفاجئ للتيار الكهربائي.**
2. **عمل النظام بفاعلية عند استخدامه من قِبل عشرة آلاف مُستخدِم في الوقتِ نفسه مثلًا.**

بعد أن يتم تحديد المتطلبات، يمكنك التعرف على كيفية جمع هذه المتطلبات.

جمع المتطلبات Requirements Gathering

تهتم عمليات التحليل بمعرفة ما يريده الأشخاص من النظام الجديد المقترح، أو الاطلاع الجيد على النظام الحالي لمعرفة كيفية عمله وبالتالي تحسينه، وتوجد عدة طرق لجمع البيانات:

الاستبانات Questionnaires

يمكن إعطاء مُستخدِم النظام الاستبانة لتعبئتها وتسليمها، ويجب أن تتمحور أسئلة الاستبانة حول الكيفية الحالية لإنجاز المهمات، وليس حول النظام بإجماليته. يمكن أن تكون الأسئلة أيضًا حول التطلعات أو التوقُّعات لدى المُستخدِم من النظام الجديد.

خصائص استخدام الاستبانات:

- تُجمع بالعادة بدون تحديد هوية المُستخدِم للحصول على إجابات أكثر مصداقية.
- تستغرق وقتًا أقل لجمعها وتحليلها مقارنة بالمقابلات.
- يمكن تحليلها تلقائيًا باستخدام النماذج الإلكترونية والبرامج المتخصصة.

تحديات استخدام الاستبانات:

- قد يؤدي عدم اهتمام المُستخدِم بمحتوى الاستبانة، أوالفهم غير الصحيح للأسئلة، وكذلك الصياغة غير السليمة أو المُعقدة للأسئلة إلى الحصول على إجابات غير صحيحة.
- لا تُستخدم الاستبانات في جمع البيانات الوصفية.

المقابلات Interviews

تستغرق المقابلات وقتًا أطول من الاستبانات، لذلك تُعدُّ طريقة مناسبة عند وجود عدد محدود من مُستخدِمي النظام، ويجب إجراء المقابلات مع جميع مُستخدِمي النظام الجديد على مختلف المستويات في المؤسسة. يُمكنك من خلال المقابلات معرفة كيفية عمل النظام الحالي والتوقُّعات وما هو مطلوب من النظام الجديد.

خصائص استخدام المقابلات:

- يُمكن للشخص الذي يقوم بإجراء المقابلة شرح وتوضيح الأسئلة بشكلٍ فوري عند الضرورة.
- يمكن تعديل الأسئلة أو تغييرها لتتناسب مع موقع وطبيعة عمل أعضاء الفريق الذين تتم مقابلتهم.
- من الطبيعي أن يأخذ المُستخدِمون المقابلة على محمل الجد أكثر من الاستبانة.

تحديات استخدام المقابلات:

- قد يصاب الأشخاص المستهدفون بالتوتر أثناء المقابلة مما يؤثر على دقة المعلومات المقدمة.
- تُعدُّ المقابلات مُكلفة بسبب الحاجة إلى زيارة أماكن تواجد المُستخدِمين وتعطيل أعمالهم اليومية.
- يستغرق ترتيب المقابلات وإجراؤها كثيرًا من الوقت، لا سيما عندما تكون هناك حاجة إلى إجراء المقابلات لكثير من المُستخدِمين.

الملاحظة المباشرة Observation

يتم في هذه الطريقة مراقبة مُستخدِمي النظام بصورة مباشرة أثناء قيامهم بالعمل، حيث يستفسر المراقب عن المشاكل التي يواجهها ذلك المُستخدِم مع النظام القديم، ثم يتناول معه الجوانب التي يجب تحسينها وتطويرها في النظام الجديد.

خصائص استخدام الملاحظة المباشرة:

- يمكن من خلالها تمييز عمليات النظام بصورة فورية.
- يتعرف المُحلِّل خلالها على تفاصيل دقيقة موجودة في النظام الحالي يصعب الحصول عليها من خلال الاستبانات والمقابلات.
- تعد أقل تكلفة من المقابلات، ولا تتطلب مقاطعة المستخدِمين أثناء أداء المهام.

تحديات استخدام الملاحظة المباشرة:

- يتطلب استخدام طريقة الملاحظة المباشرة معرفة جيدة بالنظام الحالي وكذلك بوظائف النظام الجديد.
- قد يتصرف الشخص الذي تتم ملاحظته بشكل مختلف عن المعتاد أثناء قيام مُحلِّل النظام بمراقبته.

فحص وثائق النظام Examination of Existing Documentation

تتضمن هذه الطريقة جمع البيانات وفحص أي مستندات ووثائق خاصة بالنظام الحالي.

قد يتضمن ذلك مستندات الأعمال مثل نماذج الطلبات وقوائم المخزون وما إلى ذلك، ويمكن من خلالها أيضًا الاطلاع على الأنواع المختلفة من السجلات المحفوظة.

خصائص استخدام الفحص:

- يوفر كثيرًا من الوقت، خاصة في حال توفر وثائق تحليل النظام السابقة.
- تقدم الوثائق صورة واضحة لعملية تدفق البيانات عبر النظام.
- تسمح الوثائق للشخص الذي يقوم بالتحليل بتحديد بعض مواصفات النظام المطلوب من خلال معاينة حجم النظام والفواتير وغيرها.
- تقدم الوثائق صورة واضحة عن مُدخَلات ومُخرَجات التصاميم الحالية للنظام.

تحديات استخدام الفحص:

- يعتمد بشكل كبير على جودة وثائق المؤسسة ودقة بياناتها.
- تُعدُّ عملية جمع الوثائق وتحليلها مُكلفة، وتتطلب كثيرًا من الجهد من قبل أولئك الذين يقومون بجمع وتحليل الوثائق.

فيما يلي مقارنة بين مزايا الطرق المختلفة لجمع متطلبات النظام في مرحلة التحليل وعيوبها:

جدول 2.1: المقارنة بين طرق جمع متطلبات المُستخدِم

الطريقة	المزايا	العيوب
الاستبانات	يستجيب المُستخدِمون للاستبانات بصدق خصوصًا عندما لا يتمّ طلب كتابة اسم المشارك.	قد لا تكون الاستبانات واضحة أو مفهومة جيدًا.
	تستغرق وقتًا أقل لجمع المعلومات من عدد كبير من الأشخاص.	لا يمكن جمع كافة المعلومات التي تريدها عبر الاستبانة.
المقابلات	يمكن تعديل الأسئلة لمُستخدِمين محددين اعتمادًا على طبيعة عملهم أو على معايير أخرى.	قد لا يقدم المُستخدِمون إجابات صادقة نظرًا إلى أن هويتهم غير مخفية.
	تُؤخذ المُقابَلة على محمل الجد من قِبل المشاركين أوالمُستخدِمين أو المجموعة المُستهدفة.	تستغرق المقابلات وقتًا طويلًا بالنسبة للمُحلِّلين، ويؤخذ عليها تكلفتها العالية ومحدوديتها وصعوبة ترتيبها، وذلك لضرورة تفرُّغ المُستخدِمين من وظائفهم وقت المقابلة.
الملاحظة المباشرة	يمكن للمُحلِّل الوصول إلى فهم فعلي للنظام الحالي أثناء قيام المُستخدِمين بعملهم دون إعاقة أو مقاطعة.	قد يصاب الأشخاص الذين تتم ملاحظتهم بالارتباك أثناء ملاحظتهم، مما يؤدي إلى اختلاف طريقة عملهم.

تختلف معايير اختيار طريقة جمع البيانات وفقًا لطبيعة عمل المؤسسة، وعدد الأشخاص المُستهدفين في عملية جمع البيانات والكثير من العوامل الأخرى.

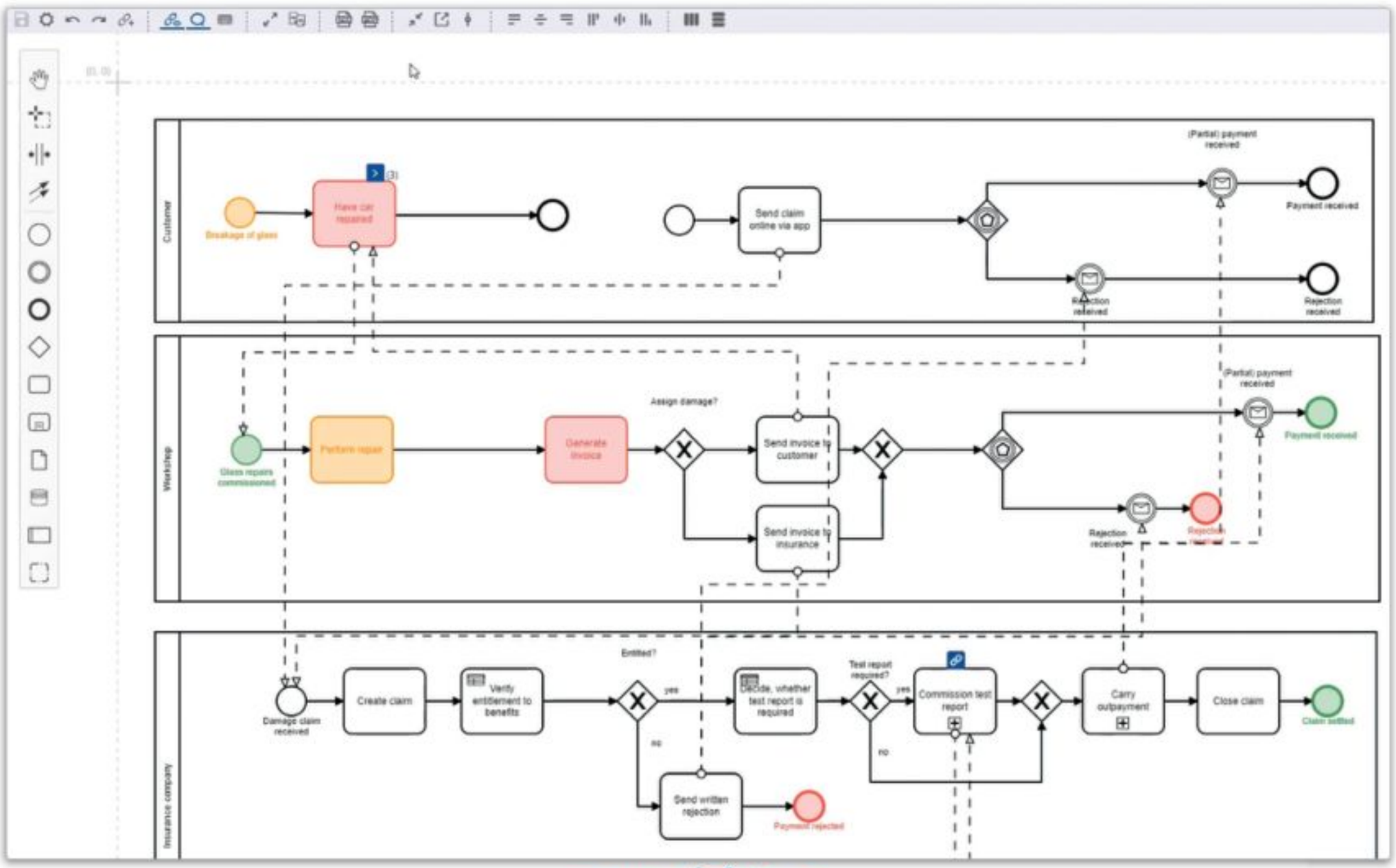
يتم في العادة استخدام أكثر من طريقة واحدة لجمع البيانات للحصول على مُخرَجات دقيقة وواقعية.

استخدام المُخطَّطات في مرحلة التحليل Using Diagrams in the Analysis Phase

تحتل المُخطَّطات والرسوم البيانية أهمية خاصة في مرحلة التحليل، وخاصة تلك التي تمثل مُخطَّطات سير العمل. يُعدُّ المُخطَّط تمثيلًا مرئيًا للمعلومات باستخدام الأشكال والأسهم لإظهار التسلسلات والعلاقات.

لماذا نستخدم المُخطَّطات؟ Why Do we Use Diagrams

يمكن من خلال المُخطَّطات تفسير البيانات الإحصائية ووظائف النظام والمُخطَّطات التنظيمية والعمليات الأخرى بشكل أفضل، ويُعدُّ التمثيل المرئي لهذه المعلومات أكثر فعاليةً في تسليط الضوء عليها. فعلى سبيل المثال، يَسهُل على القارئ المقارنة بين البيانات وتمييز النتائج عند تمثيلها باستخدام الأشكال والألوان المختلفة في المُخطَّطات. تُستخدم المُخطَّطات في مجموعة واسعة من التطبيقات، حيث يمكن استخدامها لعرض المُخطَّط التنظيمي للشركة، أو لتمثيل كيفية تدفق العمليات لإكمال مهمة، أو طريقة توصيل مكونات الشبكة وربطها معًا.



شكل 2.1: مُخطَّط لعملية معينة

أمثلة مختلفة على المُخطَّطات Different Diagram Examples

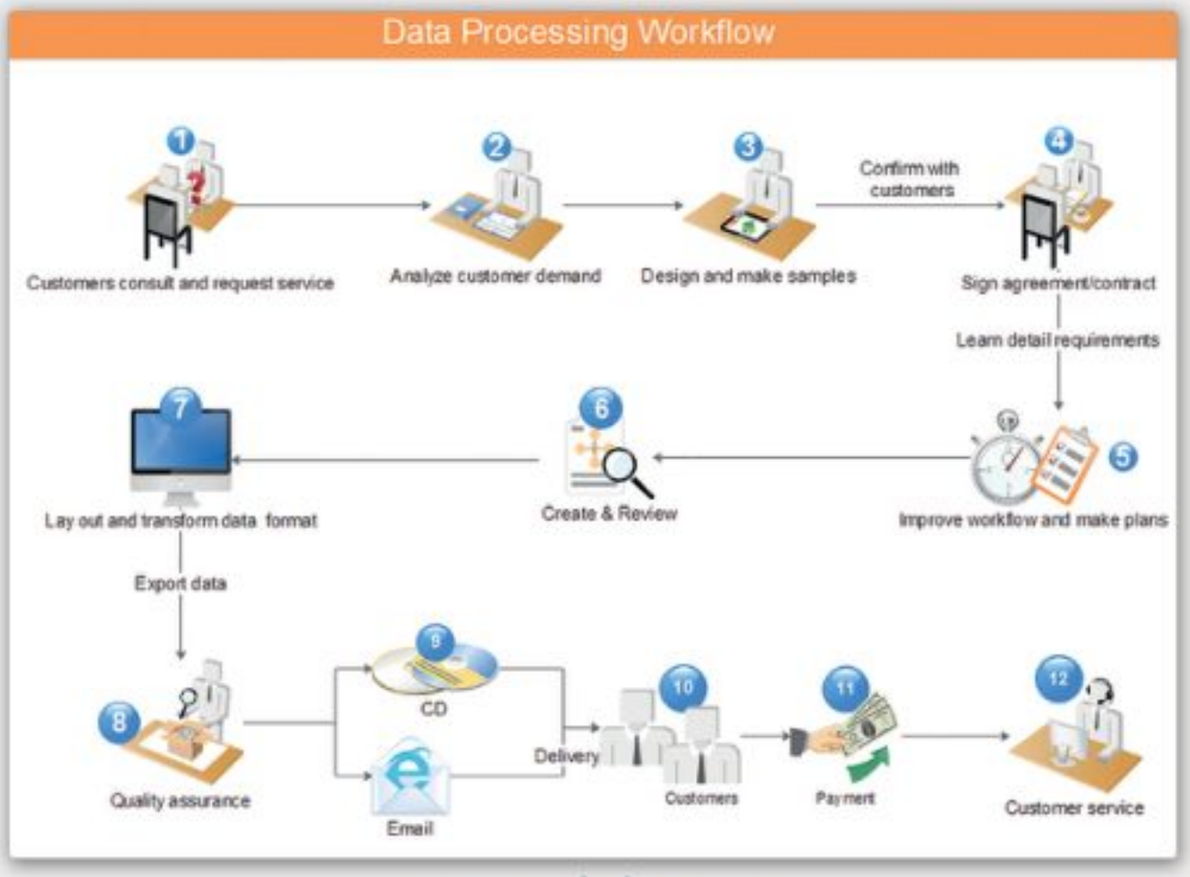
هناك العديد من أنواع المُخطَّطات التي يمكن استخدامها خلال المراحل المختلفة لدورة حياة تطوير البرمجيات، ومن ضمن هذه الأنواع:

مُخطَّط سير العمل
Workflow Diagram

يتشابه مُخطَّط سير العمل بشكل كبير مع المُخطَّط الانسيابي الذي تعلّمت تصميمه لوصف خوارزمية البرنامج.

يتكون هذا المُخطَّط في العادة من مجموعة من الرموز التي تمثل الإجراءات، ومجموعة من العمليات التي تتصل ببعضها بأسهم تشير إلى التدفق من عملية إلى أخرى.

يمكن استخدام مُخطَّطات سير العمل لإظهار تدفق العمليات أو المهام خلال كل مرحلة من مراحل دورة تطوير البرمجيات.

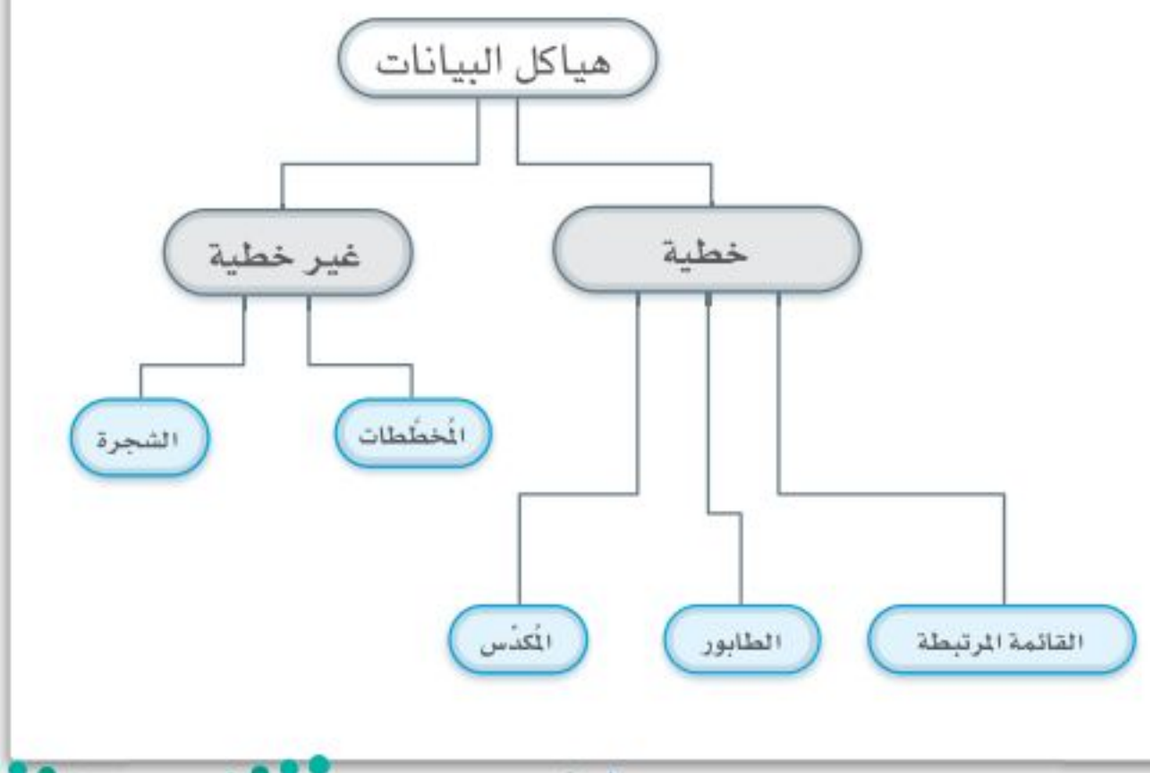


شكل 2.2: مُخطَّط سير العمل

المُخطَّط الشجري
Tree Diagram

يمثل المُخطَّط الشجري الطبيعة الهرمية للهيكلية التنظيمية لمهمة معينة على شكل رسومي. يتسم المُخطَّط الشجري بوجود ما يشبه الجذر في الأعلى، بينما توجد عناصر الشجرة التي تُسمى بالعُقَد في الأسفل.

يُستخدم هذا المُخطَّط على نطاق واسع لعرض الطبيعة الهرمية للشركة أو لمهام مشروع ما، كما يُمكن استخدامه لتمثيل الاحتمالات الشرطية في الرياضيات.

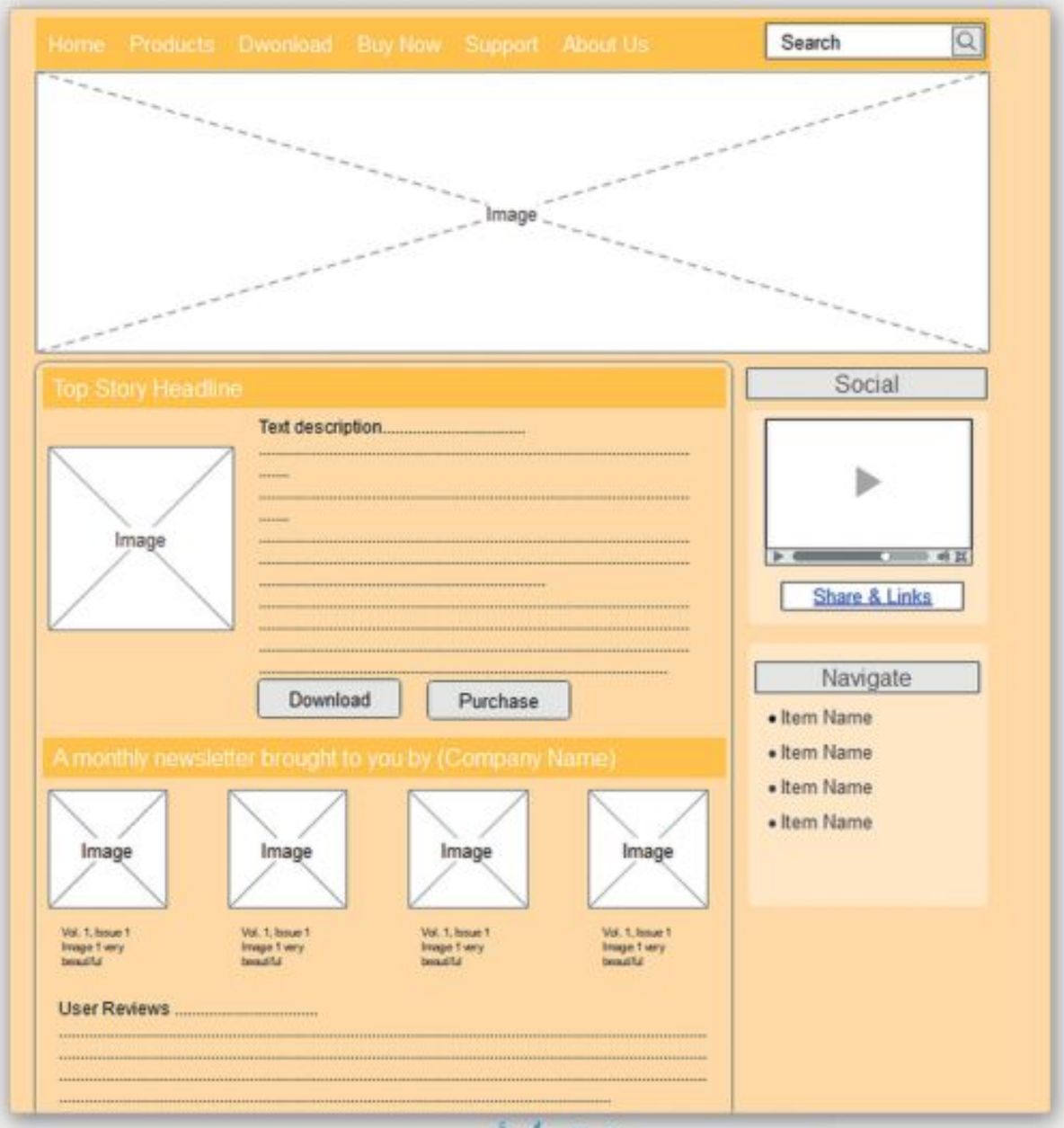


شكل 2.3: المُخطَّط الشجري

المُخطَّط الشبكي

Wireframe Diagram

المُخطَّط الشبكي هو تمثيل مرئي لتصميم موقع ويب أو تطبيق عبر الإنترنت. يفتقر هذا المُخطَّط عادةً إلى الرسومات والنصوص وذلك لتركيزه على بُنية المحتوى ووظائفه، ويُستخدم هذا المُخطَّط على نطاقٍ واسع في تطوير المواقع والتطبيقات.



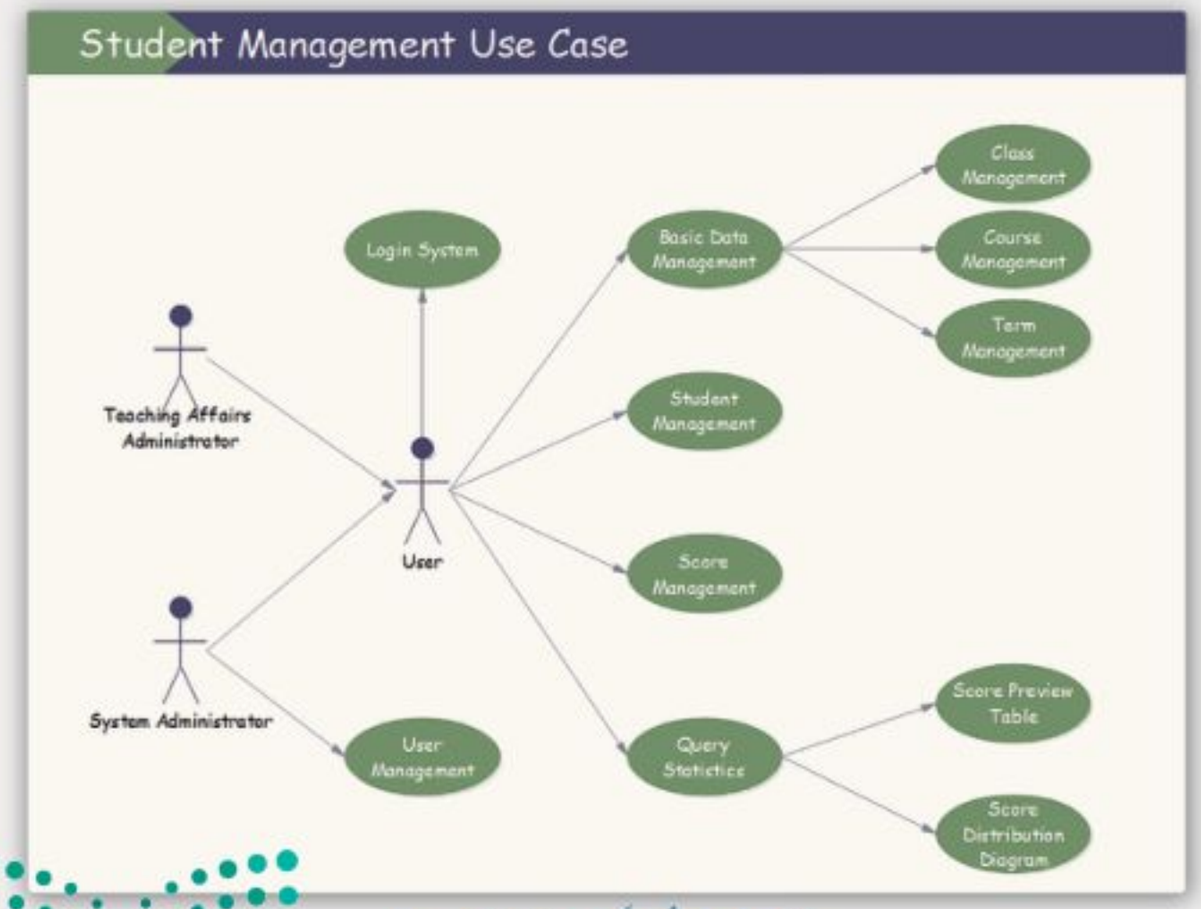
شكل 2.4: المُخطَّط الشبكي

مُخطَّط حالة الاستخدام

Use Case Diagram

مُخطَّط حالة الاستخدام هو نوع من المُخطَّطات التي تمثل الطرق المختلفة التي قد يتفاعل بها المُستخدِم مع النظام.

تُعدُّ مُخطَّطات حالة الاستخدام قَيِّمة للغاية في تمثيل المتطلبات المُجمَّعة للنظام أثناء مرحلة التحليل في دورة حياة تطوير البرمجيات.

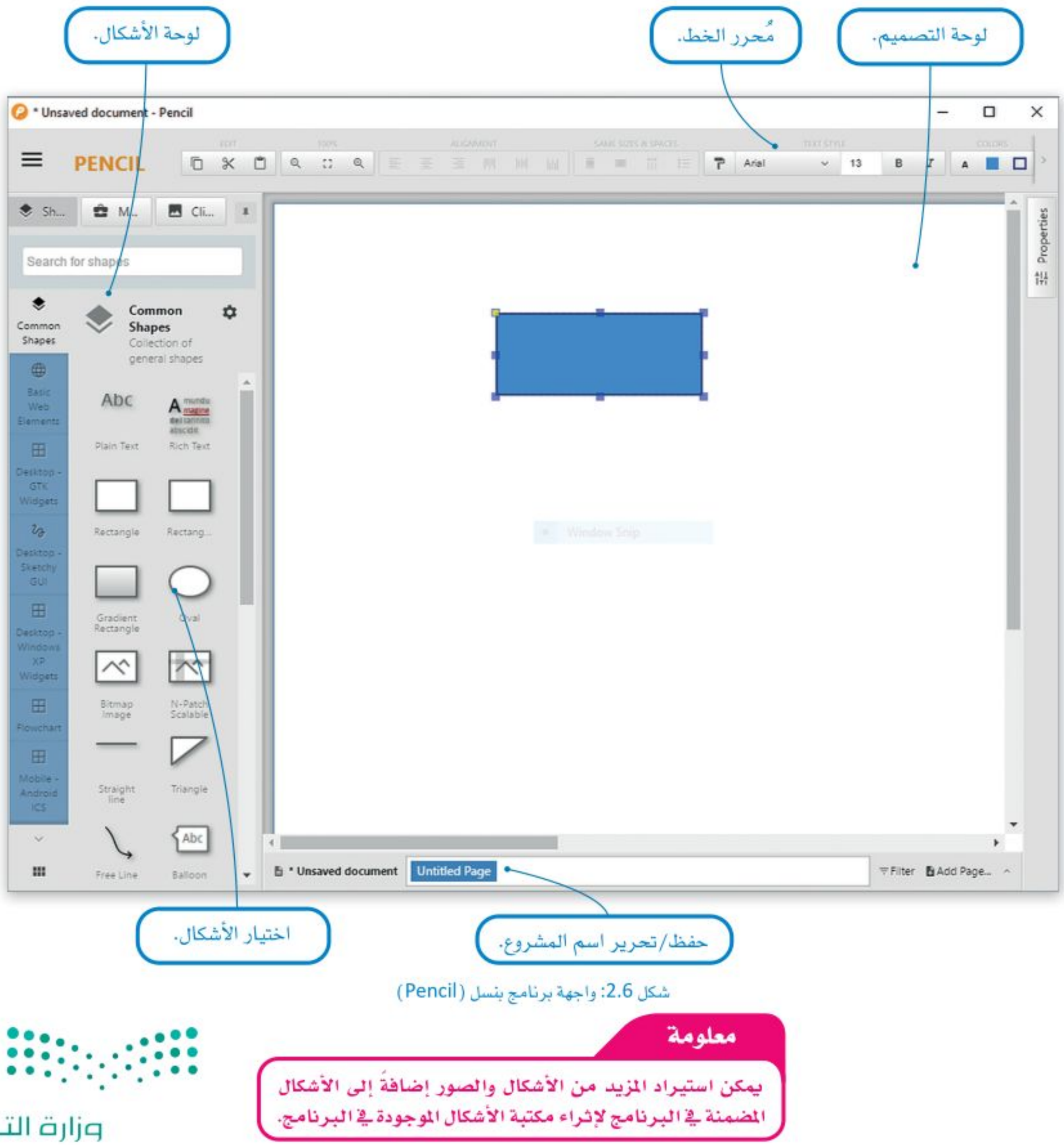


شكل 2.5: مُخطَّط حالة الاستخدام

٢ استخدام بنسل بروجكت لتصميم مُخطَّط سير العمل

Using Pencil Project to Design a Workflow Diagram

بنسل بروجكت (Pencil Project) هي أداة تُستخدم لإنشاء نماذج أولية مجانية ومفتوحة المصدر **لواجهة المُستخدِم الرسومية** (Graphical User Interface – GUI) لإنشاء المُخطَّطات. تتيح ميزات الرسم المدمجةَ ومجموعات الأشكال في بنسل بروجكت إنشاء أنواع عديدة من المُخطَّطات مثل: المُخطَّطات الانسيابية ومُخطَّطات سير العمل والمُخطَّطات الشبكية.



شكل 2.6: واجهة برنامج بنسل (Pencil)

معلومة

يمكن استيراد المزيد من الأشكال والصور إضافةً إلى الأشكال المضمنة في البرنامج لإثراء مكتبة الأشكال الموجودة في البرنامج.

أشكال مُخطَّط سير العمل الأساسية Basic Shapes of a Workflow Diagram

توجد العديد من الرموز التي يُمكن استخدامها لتمثيل الأجزاء المختلفة لمُخطَّط سير العمل. فعلى سبيل المثال، يُستخدم المستطيل لتمثيل العمليات، بينما يُستخدم المعين لتمثيل القرارات.

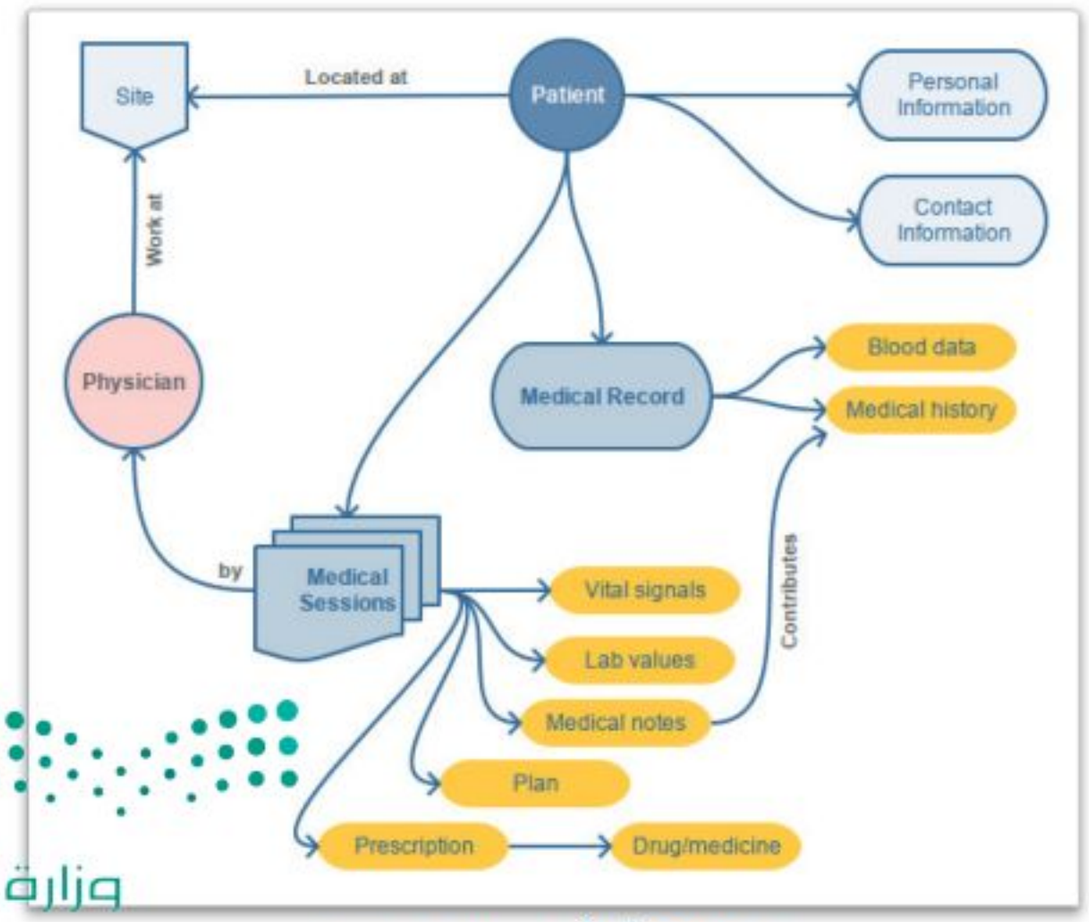
في الجدول أدناه عرض لبعض الأشكال الأساسية المُستخدمة في مُخطَّط سير العمل.

جدول 2.2: الأشكال الأساسية لمُخطَّطات سير العمل

الرمز	الاسم	الوصف
	بداية أو نهاية	يُمثل نقطة البداية أو النهاية لمُخطَّط سير العمل.
	عملية	يُمثل مجموعة من الخطوات قابلة للتكرار.
	قرار	يُمثل قرارًا يجب اتخاذه للوصول إلى عملية أو قرار آخر.
	مُستند	يُمثل وثيقة مثل تقارير الأخطاء أو أنواع أخرى من التقارير والوثائق الختامية.
	مُدخَل أو مُخرَج	يُمثل عملية إدخال أو إخراج البيانات.
	أسهم	موصِّل يوضح العلاقة بين العمليات.

إنشاء مُخطَّط جديد
Creating a New Diagram

ستستخدم في هذا الدرس برنامج بنسل بروجكت لإنشاء مُخطَّط سير العمل الخاص بعملية صيانة التطبيق الذي ستُنشئه لاحقًا؛ ليكون بمثابة الدليل السياحي للمملكة العربية السعودية لكبار السن الذين يعانون من مشاكل في الرؤية.



شكل 2.7: مُخطَّط سير العمل لتطبيق

وزارة التعليم
Ministry of Education
2023 - 1445

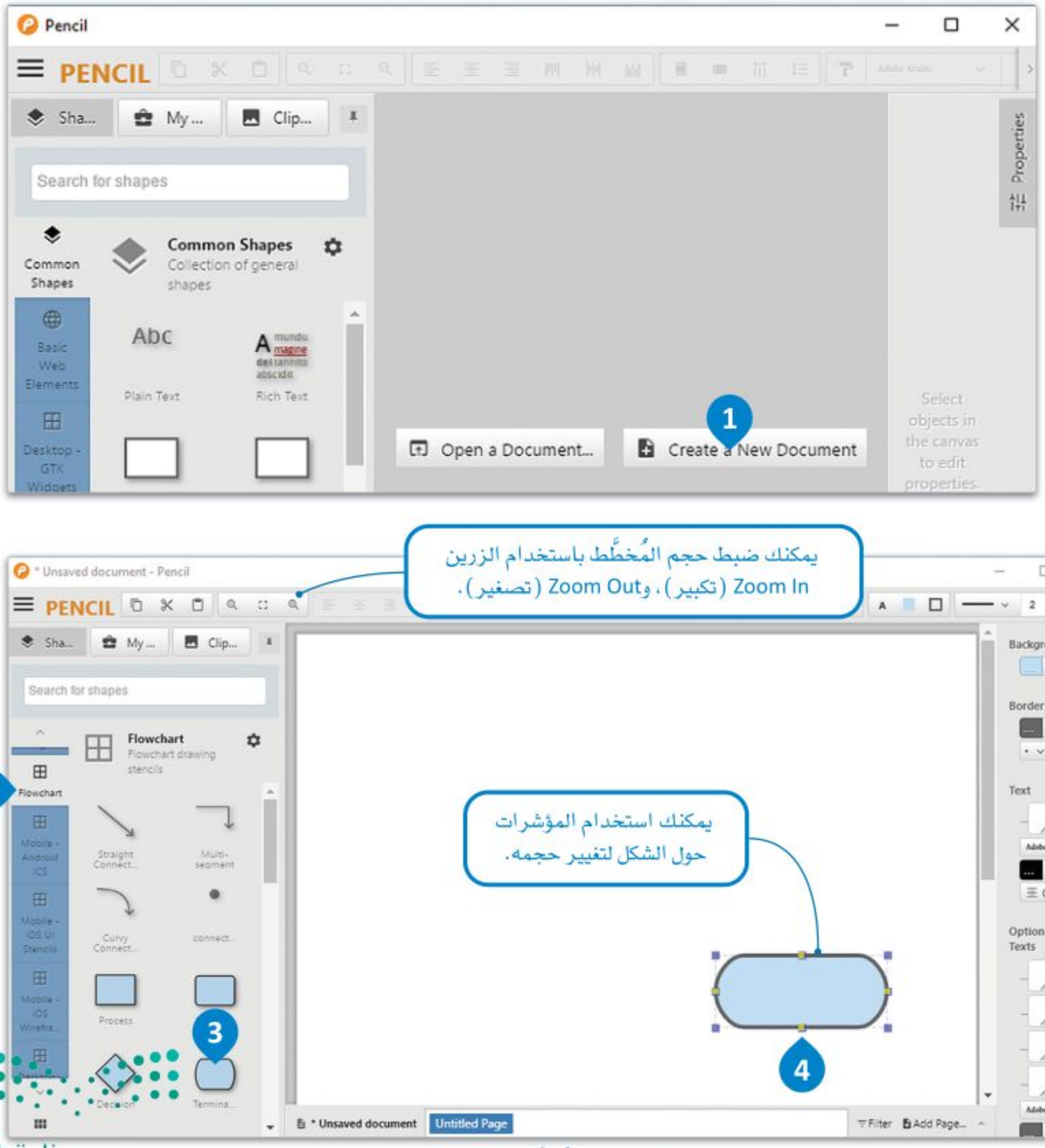
لإنشاء مُخطَّط سير العمل:

> افتح برنامج Pencil (بنسل)، واضغط على Create a New Document (إنشاء مستند جديد). 1

> من لوحة Shapes (الأشكال)، اضغط على قسم Flowchart (مُخطَّط انسيابي) لإضافة شكل. 2

> اسحب وأفلت شكل Terminator (بداية أو نهاية) في لوحة الرسم لتعيين نقطة بدء المُخطَّط. 3

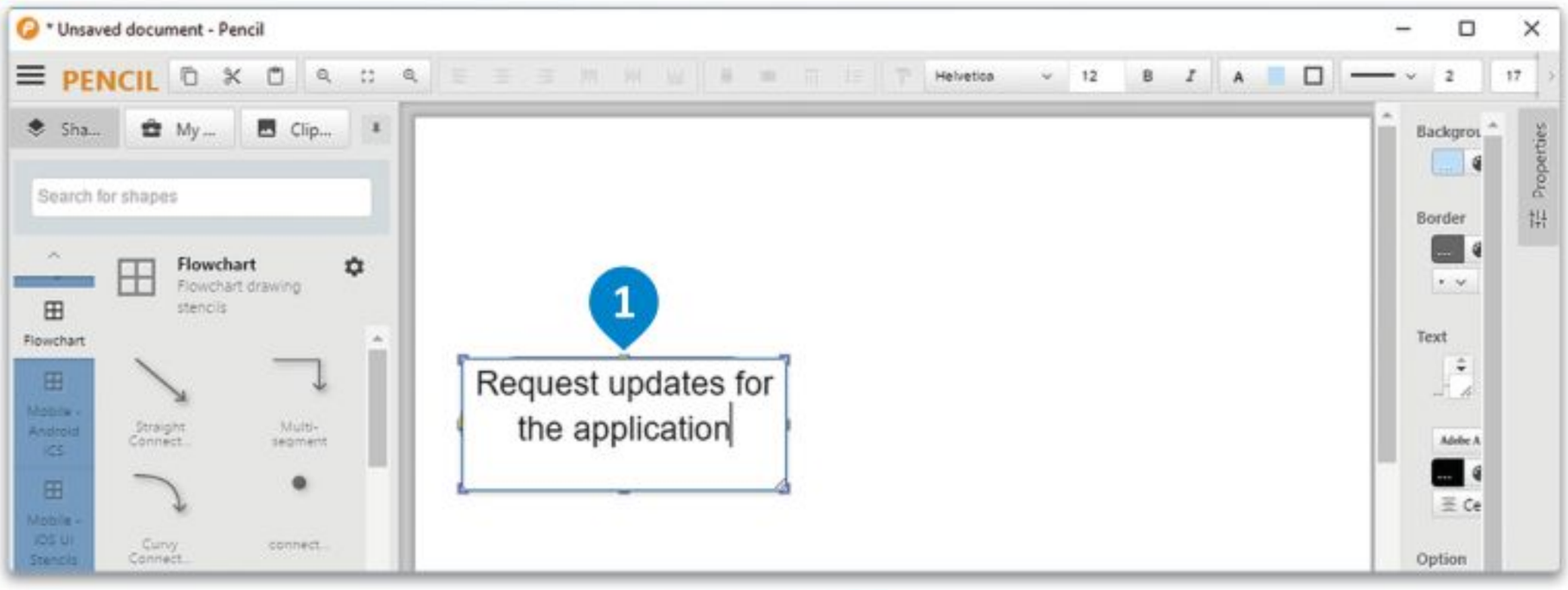
> تم إنشاء نقطة بداية المُخطَّط. 4



شكل 2.8: إنشاء مُخطَّط سير عمل جديد

لإضافة نص إلى الشكل:

> اضغط ضغطًا مزدوجًا على الشكل الذي تريد إضافة نص به، ثم اكتب النص الذي تريده. 1



شكل 2.9: إضافة نص إلى شكل

إضافة أشكال جديدة إلى المُخطَّط Adding New Shapes to the Chart

يمكنك إضافة أشكال جديدة تمثل القرارات أو العمليات أو التوثيق أو أي معلومات أخرى تريد إضافتها إلى مُخطَّط سير العمل.

لإضافة أشكال جديدة إلى المُخطَّط:

> لإضافة عملية، انتقل إلى لوحة Shapes (الأشكال)، واضغط على قسم Flowchart (مُخطَّط انسيابي)، 1 ثم اسحب شكل Decision (قرار) وأفلته في لوحة التصميم.

> تمت إضافة الشكل إلى المُخطَّط. 2

شكل 2.10: إضافة أشكال جديدة إلى المُخطَّط

إضافة الروابط والنصوص Adding Links and Texts

من الضروري إضافة الروابط التي تمثل الاتصال والعلاقات بين أشكال المُخطَّط المختلفة، ويمكنك إضافة نص بسيط لشرح أو تحليل المُخرَجات المختلفة لقرار أو عملية أو أي نموذج آخر ذي صلة داخل المُخطَّط عند الحاجة لذلك.

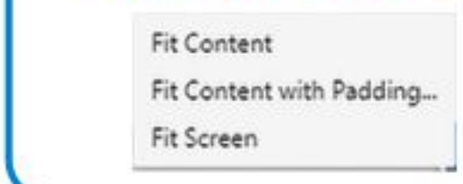
لربط شكلين:

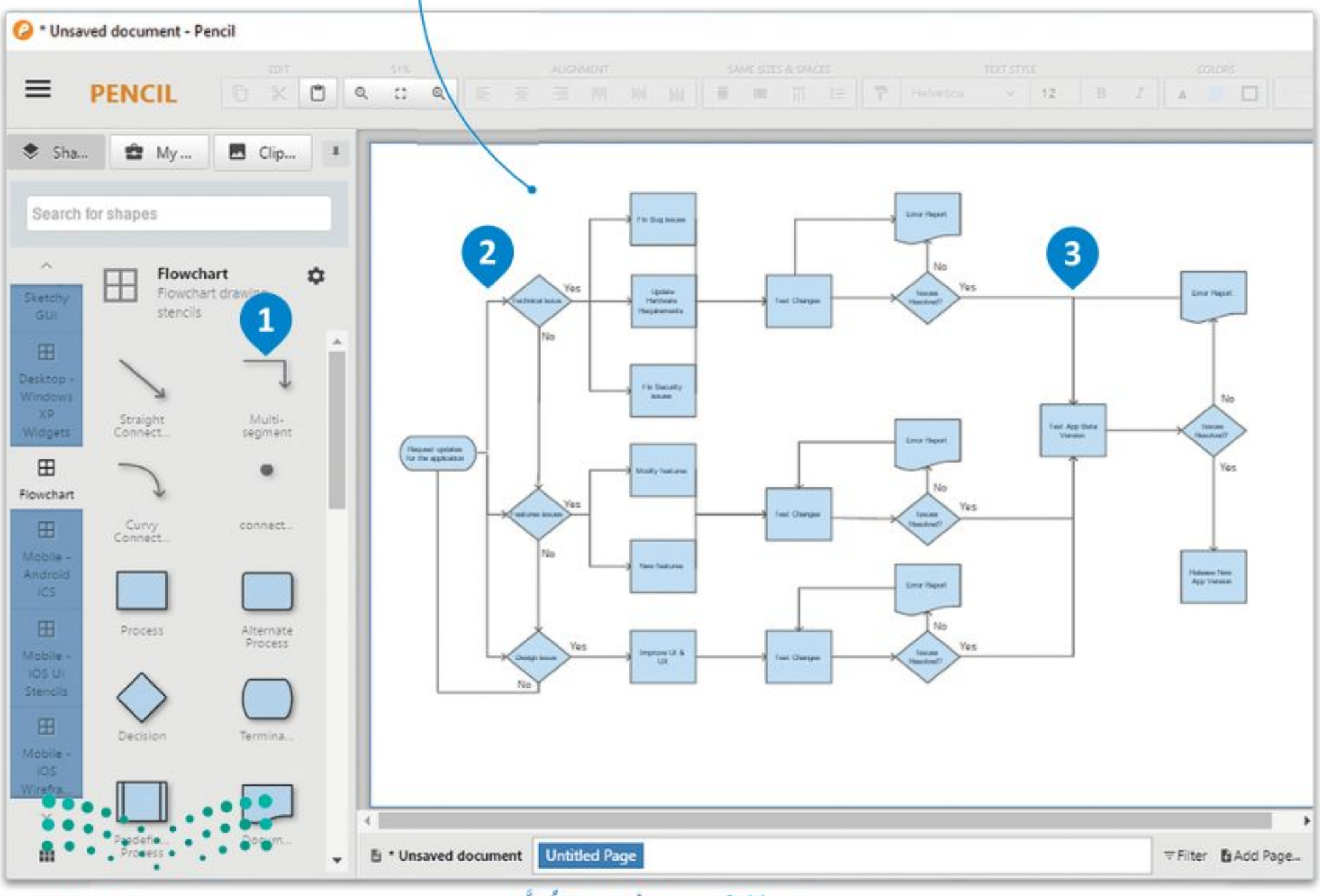
> في لوحة Shapes (الأشكال)، اضغط على قسم Flowchart (مُخطَّط انسيابي)، ثم اسحب Multi-segment Connecter (موصِّل متعدد القِطع) وأفلته في لوحة التصميم. 1

> استخدم مؤشرات الموصِّل لتوصيل نقطة بداية المُخطَّط بالقرارات الثلاثة التالية. 2

> استمر بربط جميع أشكال المُخطَّط بالموصِّلات المناسبة. 3

لجعل المُخطَّط يتناسب مع اللوحة، يتعين عليك تغيير حجمه بالضغط بزر الفأرة الأيمن عليه واختيار أحد الخيارات الثلاثة:





شكل 2.11: توصيل الأشكال في المُخطَّط

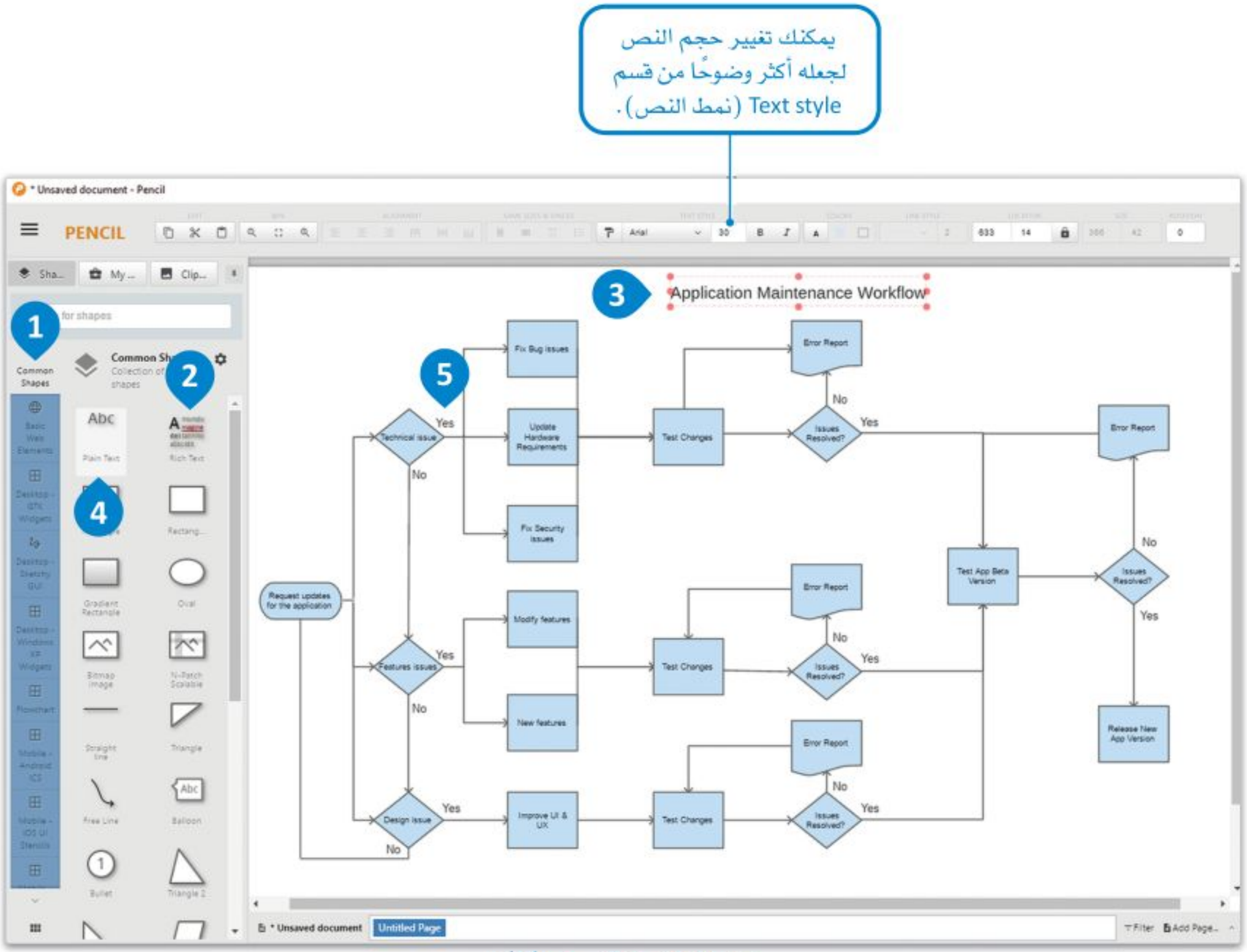
لإضافة نص إلى المُخطَّط:

> من لوحة Shapes (الأشكال)، اضغط على قسم Common shapes (الأشكال الشائعة)، ثم اسحب حقل Rich Text (نص مُنسَّق) وأفلته في لوحة التصميم. 1

> ضع حقل النص في الموضع المطلوب داخل المُخطَّط واكتب النص الذي تريده. 2

> ستتم إضافة صندوق النص إلى المُخطَّط. 3

> من لوحة Shapes (الأشكال)، اضغط على قسم Common Shapes (الأشكال الشائعة)، ثم اسحب وأفلت حقل Plain Text (نص عادي) في اللوحة، 4 لإدراج خيارات Yes (نعم) أو No (لا) في قرارات المُخطَّط. 5



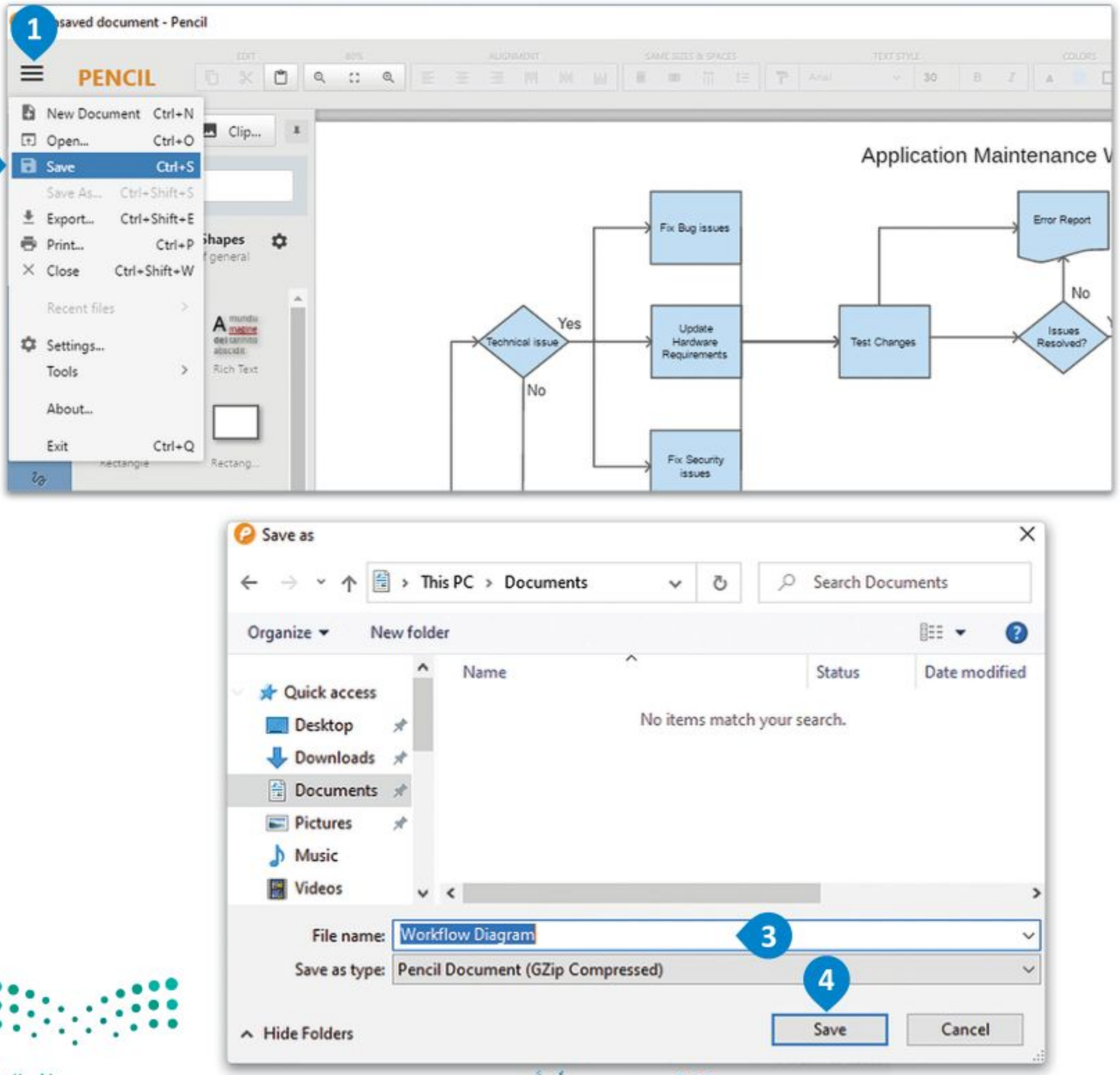
شكل 2.12: إضافة نص إلى المُخطَّط

حفظ المُخطَّط وخيارات التصدير Saving the Diagram and Export Options

عند الانتهاء من المُخطَّط النهائي، يمكنك حفظ الملف وتصديره بتنسيقات صور مختلفة مثل PNG أو PDF ، أو على شكل مُستند أو صفحة ويب.

لحفظ مشروع المُخطَّط:

> اضغط على القائمة الرئيسة، 1 ثم اضغط على Save (حفظ). 2
> في النافذة الظاهرة، اكتب اسمًا للملف المراد حفظه، 3 ثم اضغط على Save (حفظ). 4
> تم حفظ المُخطَّط.



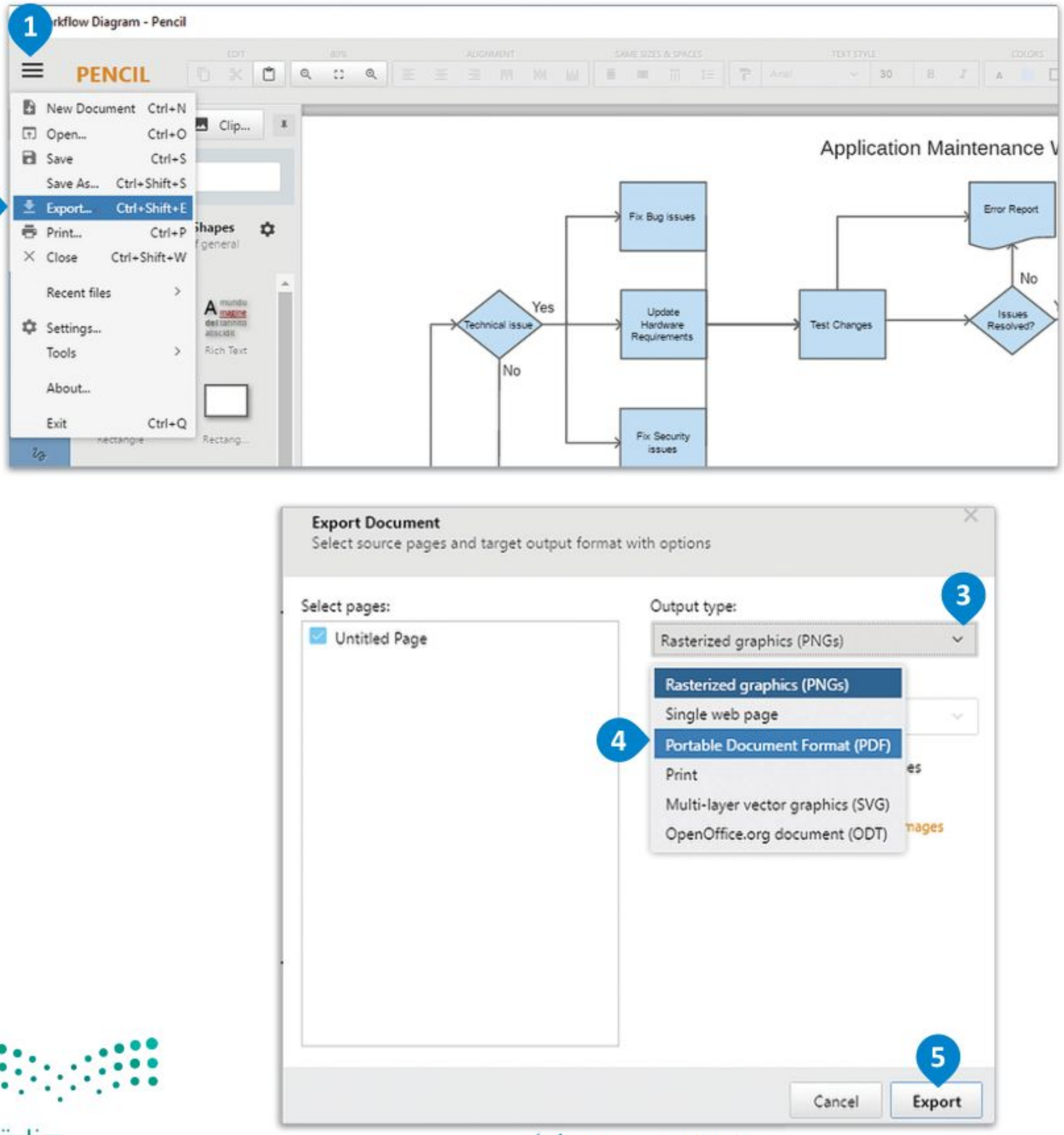
شكل 2.13: حفظ مشروع المُخطَّط

لتصدير المشروع:

> اضغط على القائمة الرئيسة، 1 ثم اضغط على Export (تصدير). 2

> في نافذة Export Document (تصدير المستند)، اضغط على Output Type (نوع الإخراج) لتحديد نوع المُخطَّط الذي تريد تصديره. 3

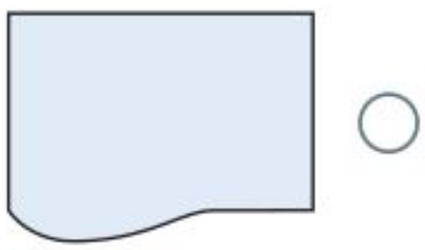
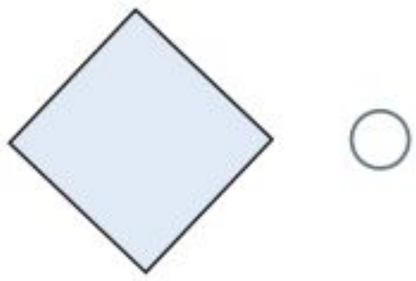
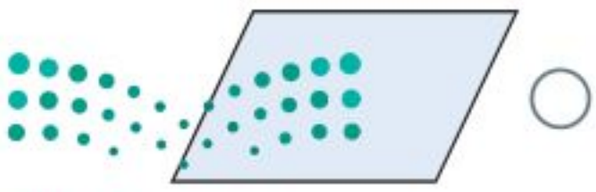
> حدد النوع الذي تريده مثل:PDF، 4 ثم اضغط على Export (تصدير). 5



شكل 2.14: تصدير مشروع المُخطَّط

تمرينات

1 افتح بنسل بروجكت وطابق بين ما تمثله الأشكال أدناه ووظيفتها:

1 نقطة بداية / نهاية	
2 مستند	
3 عملية	
4 بيانات مُدخَلة/مُخرَجة	
5 قرار	
6 موصِّل	

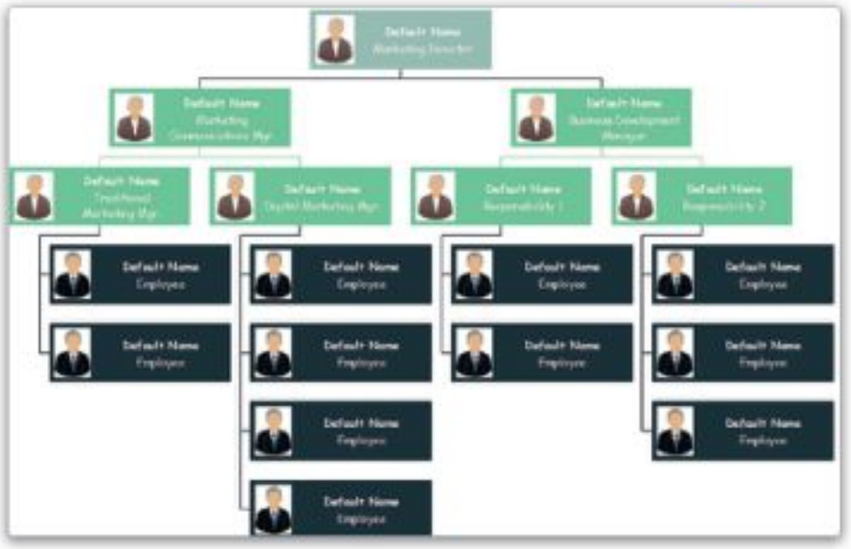
2 طابق بين المتطلبات التالية وأمثلتها فيما يلي:

متطلبات وظيفية	تكامل البيانات.
	الوظائف الإدارية.
	قابلية التوسع والقدرة.
	واجهات خارجية.
	متطلبات التقارير.
متطلبات غير وظيفية	المتطلبات التنظيمية.
	متطلبات الحصول على شهادة.
	إمكانية الخدمة والتنظيم.
	سهولة الاستخدام وقابلية التشغيل البيني.

3

حدّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يمكن أن تبقى هوية الشخص الذي تتم مقابلته مجهولة.	○	○
2. يجب أن تتم عملية الملاحظة المباشرة أثناء استخدام المُستخدِمين للنظام.	○	○
3. يُظهر فحص توثيقات النظام التصاميم الحالية للمُخرَجات والمُدخَلات.	○	○
4. يمكن الحصول على إجابات غير كافية فيما يتعلق بوظائف النظام من خلال فحص الوثائق الموجودة.	○	○
5. تُعدُّ الإجابات المقدمة من خلال الاستبانات ذات مصداقية كبيرة.	○	○
6. يمكن تقديم شرح إضافي لأسئلة الاستبانات إذا وَجد الشخص صعوبةً في فهم معنى السؤال.	○	○
7. قد يتصرف الشخص المراد مراقبته بشكل مختلف عن طبيعته أثناء الملاحظة.	○	○

4 طابق كل من المتطلبات التالية مع أمثلتها فيما يلي:



مُخطَّط سير العمل

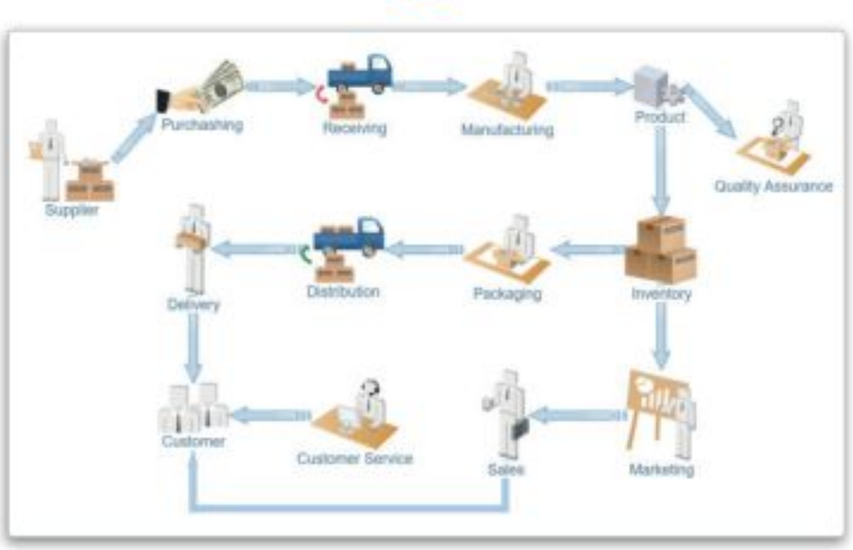
المُخطَّط الشجري

المُخطَّط الشبكي

المُخطَّط الشجري

مُخطَّط سير العمل

المُخطَّط الشبكي



المُخطَّط الشجري

مُخطَّط سير العمل

مُخطَّط حالة الاستخدام

5 اذكر استخدامًا واحدًا لكل من المُخطَّطات التالية:

1. مُخطَّط سير العمل:

2. مُخطَّط حالة الاستخدام:

3. المُخطَّط الشجري:

4. المُخطَّط الشبكي:

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

التفاعل بين الإنسان والحاسب

ما المقصود بالتفاعل بين الإنسان والحاسب ؟

What is Human-Computer Interaction (HCI)

يشير مصطلح **التفاعل بين الإنسان والحاسب** (HCI) إلى مجال دراسة التفاعل بين البشر وأجهزة الحاسب، حيث يهتم هذا المجال بتصميم وتكييف الأنظمة المُختلفة للاستخدام البشري، مع التركيز على تصميم الواجهات المُستخدمة من الأشخاص أو المُستخدمون وأجهزة الحاسب.

يهتم الباحثون في هذا المجال بالطرق التي يتفاعل بها البشر مع أجهزة الحاسب، وتقنيات التصميم المختلفة التي تسمح لهم بالتفاعل مع أجهزة الحاسب بطرق مبتكرة.

بُنيت قواعد التفاعل بين الإنسان والحاسب على ثلاثة أسس: المُستخدِم، والحاسب، وعملية التفاعل، والتي يتم تعريفها على أنها تدفُّق المعلومات بين الإنسان والحاسب.

تخصصات التفاعل بين الإنسان والحاسب

Human-Computer Interaction Majors

توسعت دراسة التفاعل بين الإنسان والحاسب لتشمل علومًا مُستقاةً من مجالات هندسة العوامل البشرية وعلم المعرفة وكذلك علوم الحاسب.

شكل 2.15: التفاعل بين الإنسان والحاسب

يهتم التفاعل بين الإنسان والحاسب بالجوانب المعرفية والأكاديمية لسلوك المُستخدِم، والتي تُعدُّ مُخرَجاتها مُدخَلات أساسية في المجال التطبيقي، والذي تستند إليه تصميمات **تجربة المُستخدم** (User Experience -UX) **وواجهة المُستخدم** (User Interface -UI) للتطبيقات المختلفة مثل تطبيقات الهواتف الذكية ومواقع الويب.

إن التعاون الفعال بين الباحثين في مجال **التفاعل بين الإنسان والحاسب** (HCI) والمُصممين لواجهة وتجربة المُستخدِم يؤدي في النهاية إلى تصميمات مثالية تلبي احتياجات المُستخدِمين.

تصميم تجربة المُستخدم User Experience Design (UX)

تشير تجربة المُستخدم (UX) إلى انطباعات الشخص ومواقفه حول استخدام مُنتَج أو نظام أو خدمة معينة، ويشمل هذا الجوانب العمليةَ والعاطفية للتفاعل بين الإنسان والحاسب.

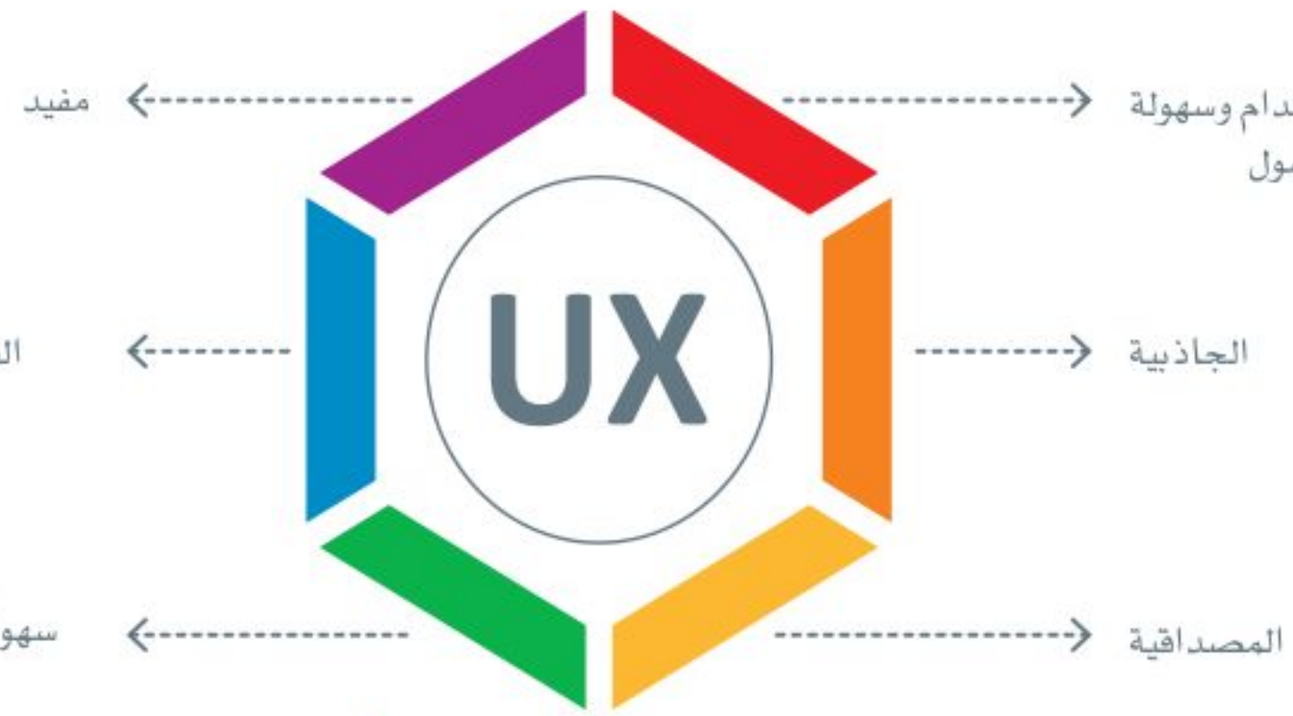
تتضمن تجربة المُستخدِم أيضًا تصور المُستخدِم لجوانب النظام المختلفة، مثل مكوناته وسهولة استخدامه وكفاءته، ويمكن تطبيق هذا المفهوم على أي نظام، مثل أجهزة الصراف الآلي والسيارات والهواتف وما إلى ذلك.

العوامل الرئيسة المؤثرة على تجربة المُستخدم:

اتسع مفهوم تجربة المُستخدِم ليشمل العديد من الجوانب بالإضافة إلى سهولة الاستخدام، وأصبح الانتباه إلى جميع جوانب تجربة المُستخدِم من أجل تقديم مُنتَجات ناجحة إلى السوق ضروريًا.

لتحسين تجربة المُستخدِم، يجب أن يمتاز تصميم النظام ومحتوياته ووظائفه بما يلي:

1. **مفيد**: يلبي احتياجات المُستخدِمين.
2. **قابلية الاستخدام**: يمكن استخدام النظام بسهولة وبديهية.
3. **ذو مظهر جذاب**: تُستخدم عناصر التصميم بطريقة فريدة لتجذب المُستخدِم وتُضفي على النظام هويته الخاصة.
4. **يَسهُل العثور عليه**: يمكن تصفح محتوياته والوصول إليها بسهولة من داخل النظام أو خارجه.
5. **سهولة الوصول**: يجب أن يشمل التصميم المُستخدِمين ذوي الاحتياجات الخاصة في خصائصه.
6. **المصداقية**: يستمد النظام محتواه من مصادر موثوقة ومعتمدة.
7. **القيمة**: يجب أن يقدم المُنتَج قيمة لمن قاموا بإنشائه وللمُستخدِمين الذي يشترونه أو يستخدمونه.



شكل 2.16: العوامل الرئيسة المؤثرة على تجربة المُستخدِم

واجهة المُستخدِم User Interface (UI)

إن واجهة المُستخدِم (UI) هي نقطة التفاعل والتواصل بين الإنسان والحاسب داخل الجهاز، ويمكن القول بأن واجهة المُستخدِم هي الشكل الرسومي للتطبيق، وتتضمن الأزرار، والنصوص المقروءة، والصور، وأشرطة التمرير، وحقول إدخال النص، إضافةً إلى العديد من العناصر الأخرى التي يتفاعل معها المُستخدِم. تحتوي واجهة المُستخدِم أيضًا على تخطيط الشاشة، والانتقالات، والتأثيرات المتحركة لواجهة المُستخدِم الرسومية، وجميع التفاصيل الدقيقة للتفاعل، وأي تأثيرات رسومية بحاجة إلى تصميم.

شكل 2.17: واجهة المُستخدِم للأجهزة اللوحية

تتمثل مهمة مُصممي واجهة المُستخدِم (UI) في تحديد مظهر التطبيق، وفي إنشاء شكل واجهة مستخدِم التّطبيق ومظهرها، ويتعين على المصمم اختيار أنظمة الألوان وأشكال الأزرار، وأنواع خطوط النص وحجمها، وطريقة عرضها.

اعتبارات إنشاء واجهة مُستخدِم جيدة:

ترتبط اعتبارات تصميم واجهة المُستخدِم بالعديد من العلوم والتخصصات الأخرى، كعلم النفس والفنون الجميلة، كما تأخذ بالاعتبار ما يلي:

1. يجب أن يشير شكل الأزرار والعناصر الأخرى إلى وظيفتها، ويجب أن يسمح التصميم للمُستخدِم باستكشاف هذه الوظائف بسهولة.

2. يجب تصميم الواجهات وإضافتها بشكل صحيح ومناسب للمُستخدِم، بحيث تكون عناصر التحكم مجاورة للأشياء المرتبطة بها.

3. يجب أن تأخذ عناصر الواجهة في الاعتبار قُدرات المُستخدِم المرئية من حيث حجم الخط وتعديل النص وسطوع الألوان والتباين.

clear text

4. يجب أن تُزوّد الواجهة مُستخدِميها بالرسائل والإشارات التي توضح استجابة النظام للأوامر المراد تنفيذها وتقديم التغذية الراجعة الفورية.

تمت إضافة المُنتَج. ✓

تحذير: لا يمكن ترك وصف المُنتَج فارغًا. !

لا يمكن إضافة المُنتَج، تأكد من اسم المُنتَج. ✗

5. يجب توفير أكبر عدد ممكن من الإعدادات أو التعبئة الافتراضية لتخفيف العِبء على المُستخدِم. على سبيل المثال، توفير النماذج المعبأة مسبقًا.

ما الذي تريد أن تراه فيما بعد؟

هل لديك أي اقتراحات أو أفكار حول المُنتَج؟ لا تتردد بالتواصل معنا.

Name

Khaled Abdullah — تعبئة تلقائية

E-mail address

khaled@email.com — تعبئة تلقائية

Tell us why you like this idea?

Send

Idea details

كيف تعمل تجربة المُستخدِم وواجهة المُستخدِم معًا؟
How Do UX and UI Work Together

غالبًا ما يتم الخلط بين مفهومي: تجربة المُستخدِم، وواجهة المُستخدِم عند تصميم صفحات الويب وتطبيقات الهواتف الذكية. يَكمُن وجه الاختلاف بينهما في أن واجهة المُستخدِم تهتم بالمُخطَّط الرسومي للتطبيق أو موقع الويب، بينما تُركز تجربة المُستخدِم على مدى سهولة أو صعوبة التفاعل مع عناصر واجهة المُستخدِم. لذلك فإنه من الطبيعي أن تفرض تجربة المُستخدِم مواصفات واجهة المُستخدِم.

الحاسب المكتبي والهاتف الذكي Desktop Computer and Smartphone

لقد أصبحت الأجهزة المحمولة أكثر استخدامًا في هذه الأيام مقارنة بأجهزة الحاسب، كما تحولت إلى أداة أساسية للتسوق الإلكتروني ولتصفح وسائل التواصل الاجتماعي، وأداء مهام أخرى عبر الإنترنت. وهكذا فإنه من المهم مراعاة جهاز المُستخدِم عند تطوير مواقع الويب والتطبيقات الذكية، والتي ستتناولها الدروس القادمة.

عند تصميم التطبيقات ومواقع الويب، يجب مراعاة إمكانيّة عملها على جميع الأجهزة مثل الأجهزة المحمولة والمكتبية، وكيفية اختلاف تجربة المُستخدِم عند استخدام التطبيق على الهاتف المحمول مقارنةً باستخدام الحاسب المكتبي. يجب الانتباه إلى العوامل المهمة التي تجعل الهاتف المحمول مختلفًا، وفهمها جيدًا، وبالتالي وضع هذه العوامل بالاعتبار عند تصميم تطبيق الهاتف المحمول أو موقع الويب.

الخصائص الرئيسة لأجهزة الحاسب المكتبية والهواتف المحمولة
The Main Characteristics of Desktop Computers and Mobile Phones

يُوضِّح الجدول التالي خصائص الأجهزة المحمولة، مثلًا: الهواتف الذكية والأجهزة اللوحية، وأجهزة الحاسب المكتبية والمحمولة.

جدول 2.3: الخصائص الأساسية للأجهزة

الخاصية	أجهزة الحاسب المكتبية والمحمولة	الهواتف المحمولة والأجهزة اللوحية
حجم الشاشة	يمكن لأجهزة الحاسب المكتبية الاتصال بعدة شاشات مما يتيح للمُستخدِم اختيار ما يناسبه. يتراوح حجم شاشات الحاسب المكتبي أو الحاسب المحمول بشكلٍ عام بين 15 و 30 بوصة.	تختلف شاشات الهواتف الذكية حسب الشركة المصنعة والطراز. ومع ذلك، فهي دائمًا أصغر من أجهزة الحاسب المكتبية أو المحمولة، ويتراوح حجم الشاشة عادةً بين أربع وسبع بوصات.
دقة الشاشة	تحتوي أصغر شاشة حاسب محمول على عدد 2304 × 1440 بكسل.	تحتوي معظم الأجهزة المحمولة على عدد بكسلات أقل من أجهزة الحاسب المكتبية، فعلى سبيل المثال، تبلغ شاشة الهاتف الذكي عالية الجودة 1334 × 750 بكسل.
إمكانية النقل	على الرغم من أن أجهزة الحاسب المحمولة عادةً ما تكون خفيفة ومحمولة، إلا أنها لا تستطيع منافسة الهواتف الذكية من هذه الناحية.	الهواتف الذكية خفيفة الوزن ويمكن وضعها في الجيب أو حقيبة اليد، وهي مصممة لتكون مع المُستخدِم في كل مكان. أما الجهاز اللوحي فلا يمكن وضعه في الجيب، ولكن يمكن حمله بيد واحدة أيضًا.

وزارة التعليم
Ministry of Education
2023 - 1445

الخاصية	أجهزة الحاسب المكتبية والمحمولة	الهواتف المحمولة والأجهزة اللوحية
طرق الإدخال	تُستخدم لوحة المفاتيح أو الفأرة للإدخال، وهي سلسة وسهلة الاستخدام لغالبية المُستخدِمين، وتأتي بأحجام مختلفة.	قد تحتوي الهواتف الذكية على لوحة مفاتيح على الشاشة أو شاشة تعمل باللمس أصغر بكثير من الحاسب، ويمكن أن يواجه المُستخدِمون الذين يعانون من مشاكل الرؤية أو لديهم أصابع كبيرة صعوبة في الكتابة.
تشغيل البرامج	يمكن لأجهزة الحاسب المكتبية والمحمولة تشغيل برامج أكثر قوة من الهاتف الذكي أو الجهاز اللوحي، وذلك نظرًا لحجمها ومكوناتها التي تتيح وجود مصادر أكبر من الطاقة.	على الرغم من التطور الهائل في إمكانيات الهواتف المحمولة، إلا أن إمكانية تشغيل البرامج الضخمة عليها ما زالت محدودة مقارنة بأجهزة الحاسب المكتبية أو المحمولة.
نظام التشغيل	تم تصميم أنظمة تشغيل أجهزة الحاسب المكتبية والمحمولة لتحقيق الاستفادة القصوى من وحدات المعالجة المركزية السريعة، ومساحة القرص الكبيرة، والمساحة الكبيرة لذاكرة الوصول العشوائي، واستخدام ميزات الرقائق الجديدة التي لا تمتلكها معظم الأجهزة المحمولة. يعتبر نظام تشغيل مايكروسوفت مغلق المصدر.	تم تصميم أنظمة تشغيل الأجهزة المحمولة أندرويد (Android) وآي أو أس (iOS) لتعمل على مجموعة محددة من الأجهزة مع منح المُستخدِم إمكانية الوصول المقيد إلى المكونات، ويوجد كذلك تقييدات على متطلبات الأجهزة لارتباط بيئة تطبيقات الأجهزة المحمولة بميزات خاصة بأجهزة معينة. لا يمكن أيضًا تشغيل أحدث التطبيقات على نظام تشغيل قديم، والعكس صحيح. يعتبر نظام التشغيل أندرويد مفتوح المصدر.
الاتصال بالإنترنت	تتميز أجهزة الحاسب المكتبية بإمكانية الاتصال بشبكة الإنترنت السلكية إيثرنت (Ethernet) من خلال بطاقة الاتصال بالشبكة، ويتطلب أغلبها بطاقة خارجية للاتصال بالشبكة اللاسلكية، أما أجهزة الحاسب المحمولة فيحتوي معظمها على بطاقتي شبكة: لاسلكية وسلكية.	يمكن للهواتف الذكية والأجهزة اللوحية الاتصال بشبكات واي فاي (Wi-Fi) اللاسلكية للوصول إلى الإنترنت، ويمكن للهواتف الذكية ومعظم الأجهزة اللوحية أيضًا الاتصال بشبكة بيانات الجوال التي تتيح الوصول إلى الإنترنت من أي مكان تقريبًا، ولكنها قد تكون أكثر تكلفة.

الاختلافات الوظيفية بين الهواتف المحمولة وأجهزة الحاسب المكتبية
Functional Differences between Mobile Phones and Desktop Computers

للهواتف المحمولة وأجهزة الحاسب وظائف مختلفة، ولكل من هذه الوظائف أهميته الخاصة، فبينما تُوفِّر الهواتف المحمولة المرونة اللازمة للمُستخدِم للبحث عبر الإنترنت أو استخدام البريد الإلكتروني من أي مكان، فإن أجهزة الحاسب تُستخدَم للمهام الأكثر تعقيدًا، واستخدام كليهما في الوقت نفسه يتيح إنجاز العمل والمهام بشكلٍ فعّال.

وزارة التعليم
Ministry of Education
2023 - 1445

واجهة مُستخدِم أندرويد ونظام تشغيل ويندوز
Android User Interface and Windows Operating System

أدَّى الاعتماد المتزايد للعديد من الشركات على تطبيقات الويب وتطبيقات الهاتف المحمول إلى تركيز الشركات على تحسين واجهة المُستخدِم من أجل تحسين تجربة المُستخدِم ككل، ولذلك توجد مجموعة متنوعة من أنواع واجهات المُستخدِم.

يدعم كل من نظام تشغيل مايكروسوفت ويندوز وقوقل أندرويد **واجهة المُستخدِم الرسومية** (Graphical User Interface - GUI)، مما يعني أنه بدلًا من كتابة الأوامر، يتم التعامل مع كائنات رسومية مختلفة مثل الأيقونات باستخدام المؤشر. يتشابه المبدأ الأساسي لكثير من واجهات المُستخدِم الرسومية المختلفة للغاية، ولذلك فإن معرفة المُستخدِم بكيفية استخدام واجهة مُستخدِم ويندوز ستمكنه من استخدام واجهة أندرويد وبعض واجهات المُستخدِم الرسومية الأخرى.

شكل 2.18: اعتبارات واجهة المُستخدِم الرسومية في تطوير التطبيقات

فيما يلي بعض الخصائص الرئيسة لواجهة وتجربة المُستخدِم لنظامي التشغيل مايكروسوفت ويندوز وقوقل أندرويد بصفتهما أكثر أنظمة التشغيل استخدامًا.

واجهة مُستخدِم مايكروسوفت ويندوز
Microsoft Windows User Interface

يستخدم ويندوز صناديق حوار تحتوي على العديد من العناصر المرئية، وتعرض هذه الصناديق للمُستخدِم أكبر قدر ممكن من المعلومات ذات العلاقة بسرعة، ويمكن للمُستخدِم باستخدام الفأرة والحد الأدنى من الكتابة على لوحة المفاتيح الاختيار بشكل مناسب وتشغيل التطبيقات أو الأوامر المطلوبة.

واجهة مُستخدِم قوقل أندرويد
Google Android User Interface

تختلف متطلبات تصميم واجهة المُستخدِم للأجهزة المحمولة بشكل كبير عن تلك الخاصة بأجهزة الحاسب المكتبية، حيث يفرض حجم الشاشة الصغير وأدوات التحكم في الشاشة التي تعمل باللمس اعتبارات خاصة عند تصميم واجهة المُستخدِم لضمان سهولة الاستخدام والقراءة والاتساق.

يُمكن استخدام الأيقونات في واجهة الهاتف المحمول على نطاق واسع، وقد يتم إخفاء عناصر التحكم تلقائيًا عند عدم استخدامها، ويجب أن تكون الأيقونات نفسها أصغر حجمًا، وقد يكون من الضروري إخفاء وصف الأيقونات لتفادي إرباك المُستخدِم. يجب أن يكون المُستخدِمون قادرين على معرفة ما تشير إليه كل أيقونة سواء من خلال النصوص أو التمثيل الرسومي الذي يُمكن فهمه بسهولة.

شكل 2.19: واجهة المُستخدِم لأنواع مختلفة من الأجهزة

وزارة التعليم
Ministry of Education
2023 - 1445

تمرينات

1 أجب عن الأسئلة التالية بناءً على ما تعلمته في هذا الدرس:

1. ما المقصود بالتفاعل بين الإنسان والحاسب (HCI)؟ اذكر مكوناته.

2. ما المقصود بتصميم واجهة المُستخدِم (UI)؟

2 صِف بإيجاز الفرق بين تجربة المُستخدِم (UX) وواجهة المُستخدِم (UI)؟

تجربة المُستخدِم (UX)	واجهة المُستخدِم (UI)

3

ضع علامة (✓) أمام نوع الجهاز المناسب لكل وصف من الأوصاف التالية:	أجهزة الحاسب المكتبية	الأجهزة المحمولة
1. أجهزة قليلة التكلفة وبمواصفات عالية.	○	○
2. يُمكن أن يصل حجم شاشتها إلى 30 بوصة.	○	○
3. تحتوي على شاشة ذات دقة أعلى.	○	○
4. خفيفة الوزن ويمكن وضعها داخل الجيب.	○	○
5. يتم توصيلها عادةً بالفأرة ولوحة المفاتيح.	○	○

4 قارن بين الوسائل التي تتصل بها أجهزة الحاسب المكتبية والأجهزة المحمولة بالإنترنت.

5

حدِّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. يهتم تفاعل الإنسان والحاسب (HCI) حصريًا بتطوير منطق الأعمال للتطبيقات.	○	○
2. من أهم أوجه التشابه بين تجربة الهاتف المحمول وجهاز الحاسب المكتبي استخدام الأشخاص لها بالطريقة نفسها، وللمهام نفسها.	○	○
3. يشمل تفاعل الإنسان والحاسب (HCI) المجال العلمي لعلم المعرفة.	○	○
4. لا يؤثر نظام التشغيل على سرعة الهاتف المحمول أو الحاسب المكتبي.	○	○
5. تمنحك الأجهزة المحمولة وصولًا كاملًا إلى جميع موارد الجهاز.	○	○
6. تُعدُّ شبكات بيانات الهاتف المحمول أرخص وسيلة للاتصال بالإنترنت.	○	○
7. يستخدم مايكروسوفت ويندوز وقوقل أندرويد مكونات واجهة المُستخدِم الرسومية نفسها.	○	○
8. يتمتع مُستخدِمو الأجهزة المحمولة بإمكانية تصفح الإنترنت أثناء تنقلهم أو أثناء استخدام وسائل النقل العام.	○	○
9. يَستخدِم الأشخاصُ في البيئات المكتبيّة الأجهزةَ المحمولة أكثر من استخدام أجهزة الحاسب.	○	○
10. يؤثر الاختلاف في استخدام الهواتف الذكية وأجهزة الحاسب المكتبية على طريقة عمل مواقع الويب والتطبيقات التي يجب أن تعمل بشكل جيد على كل جهاز.	○	○

6 أدرج بعض ميزات واجهة المُستخدِم (UI) وتجربة المُستخدِم (UX) الرئيسة لنظامي تشغيل مايكروسوفت ويندوز وقوقل أندرويد.

نظام تشغيل ويندوز	نظام تشغيل قوقل أندرويد

7 وضِّح الاختلاف بين أنظمة تشغيل أجهزة الحاسب المكتبية والأجهزة المحمولة.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

إنشاء نموذج أولي

تصميم النظام System Design

مرحلة تصميم النظام تلي مرحلة التحليل، حيث يتم خلال هذه المرحلة تحديد عناصر النظام ومكوناته وواجهات النظام، وتشمل هذه المرحلة أيضًا العمل على تحديد أو تصميم هيكلية النظام ومكونات الأجهزة وأنظمة التشغيل والبرمجة والتكامل مع الأنظمة الأخرى ومسائل الأمان في النظام.

العمليات الرئيسة لمرحلة التصميم The Main Operations of the Design Phase

يتمحور العمل في مرحلة التصميم حول الشكل الذي ستبدو عليه واجهات النظام، وكيفية عمل وظائف النظام. تُركِّز بعض أجزاء هذه المرحلة على الميزات الفنية للنظام بينما تُركِّز الأجزاء الأخرى على كيفية استجابة النظام وتفاعله مع المُستخدِم.

العمليات الأخرى المتضمنة أثناء مرحلة التصميم

Other Operations Involved During the Design Phase

- تصميم المُدخَلات المعتمدة على الشاشة، والتي تشمل تصميم طريقة إدخال البيانات إلى النظام، وذلك من خلال صناديق النص والقوائم المُنسدلة والنماذج وغيرها.
- تصميم مُخطَّطات واجهة المُستخدِم، والتي تتضمن مظهر قوائم النظام أو صفحات الويب أو التطبيقات، وتُعدُّ المُخطَّطات الهيكليةَ أداةً رائعة لهذه العملية.
- تصميم تقارير النظام، والتي تشمل عملية تصميم مُخرَجات النظام مثل: تقارير الاستخدام، أو الملخصات، أو البيانات الإحصائية والفواتير، والأنواع الأخرى من التقارير المطبوعة.
- تصميم المُخرَجات المعتمدة على الشاشة، والتي تشمل ما يتم عرضه على الشاشة، وتقارير النظام كنتائج البحث أو رسائل الخطأ أو أي نوع من التقارير التي تظهر على الشاشة فقط.
- تصميم الهياكل الخاصة بتخزين البيانات، بما في ذلك تصميم كيفية تخزين البيانات في قواعد البيانات والجداول.
- تصميم قواعد التحقق من صحة المُدخَلات وقواعد التحقق من البيانات بحيث تتضمن كيفية منع إدخال البيانات غير الصحيحة أو أخطاء النظام وكيفية التحقق من صحتها.

بعد أن تعرَّفت على مفهوم واجهة المُستخدِم وتجربة المُستخدِم، وعلى أنظمة تشغيل الهواتف الذكية، أصبح لديك القدرة لإنشاء تطبيق للمُستخدِمين ذوي الاحتياجات الخاصة. تحتاج أولًا إلى تصميم النموذج الأولي للتطبيق، ومن أجل هذا الغرض ستستخدم برنامج **بنسل بروجكت** (Pencil Project).

النموذج الأولي Prototype

النموذج الأولي هو نموذج يحاكي المُنتَج الذي ترغب بإنشائه، حيث يُنشئ المصممون النماذج الأولية لتحديد كيفية تفاعل المُستخدِمين مع المنتَج، واختباره قبل إنتاجه، فمن غير المنطقي اختبار تصميم المُنتَج النهائي بعد الانتهاء من بنائه.

يتم تصميم النماذج الأولية لتمكين المصممين من التفكير في الحلول بشكل إبداعي، وذلك للتقليل من احتمالات الفشل، وتجنب إضاعة الوقت والمال في تطوير الأفكار غير القابلة للتطبيق.

أهمية النموذج الأولي The Importance of the Prototype

تساعد النماذج الأولية في التركيز على الوظائف الأساسية للتطبيق، وتعطي العميل فكرة واضحة عن شكل المُنتَج ومظهره، مما يساعد في اتخاذ القرار الصحيح بشأن مدى ملاءمته لتوقُّعات العميل.

جدول 2.4: مدى أهمية النمذجة الأولية

فهم أفضل لمحتوى التصميم	تُوفِّر النماذج الأولية تصورًا قويًا للتصميم لفهم شكل المُنتَج النهائي ومظهره، وتساعد الفريق أيضًا على فهم خلفيات التصميم المطلوب وطبيعة المُنتَج والفئة المستهدفة.
تسهيل عملية الحصول على التغذية الراجعة	يُمكن باستخدام النماذج الأولية جمع التعليقات والملاحظات من الأشخاص المَعنيين في كل مرحلة من مراحل تطوير المُنتَج، وذلك لإضافة ميزات جديدة أو لإعادة تصميم أجزاء من المُنتَج، وأيضًا لاختبار ما يناسبهم وما لا يناسبهم وفقًا للأهداف المحددة للتطبيق قيد الإنشاء.
التحقق من صحة التعديلات قبل التطوير	تسمح النماذج الأولية بإجراء مناقشات متعددة بشأن التغييرات في العمل قبل الدخول في مرحلة التطوير النهائية، وتسهل هذه العملية اعتماد التغييرات المناسبة وتضمن بِناء متطلبات واقعية تلبي هدف التطبيق.
التغييرات المبكرة توفر الوقت والتكلفة	تساعد التغييرات المبكرة على تحقيق الأهداف بشكل أسرع، ويُعدُّ إجراء التعديلات في المراحل النهائية من المشروع أمرًا مكلفًا للغاية، وقد يتطلب إعادة هيكلة جذرية للمُنتَج ومزيدًا من التفكير وإعادة الصياغة. يتيح امتلاك نموذج أولي جاهز إجراء التغييرات اللازمة مبكرًا قبل استثمار كثير من الوقت والجهد في إنشاء المُنتَج النهائي.

فئات النماذج الأولية Prototyping Categories

هناك طرق مختلفة للنمذجة، ويجب دائمًا تحديد الطريقة الصحيحة التي تُناسب المُنتَج والموارد المتاحة للعمل.

يتم تصنيف فئات النماذج الأولية بشكل عام بناءً على دقتها، وتنقسم إلى ثلاث فئات هي: فئة الدقة المنخفضة، وفئة الدقة المتوسطة، وفئة الدقة العالية.

النموذج الأولي منخفض الدقة
Low-Fidelity Prototype

> عادة ما يتم إنشاء هذا النموذج على الورق في مراحل التصميم الأولية، ويتم تنقيحه باستمرار طوال هذه المراحل.

> يساعد هذا النموذج على إجراء التغييرات بسهولة وسرعة، حيث يُركِّز بشكل أكبر على كيفية استخدام النظام بدلًا من الشكل الذي يبدو عليه.

> يصعُب الحفاظ على النموذج منخفض الدقة في دورة التطوير نظرًا لأنّ المُنتَج يصبح أكثر تعقيدًا مع مرور الوقت، مما يجعل النماذج الأولية الورقية غير فعّالة في مواكبة العمق المطلوب للتصميم في تلك المرحلة.

شكل 2.20: نماذج أولية منخفضة الدقة

النموذج الأولي متوسط الدقة
Medium-Fidelity Prototype

> هو نموذج يتم إنشاؤه لمحاكاة وظائف النظام وتمثيلها، مهما كانت تلك الوظائف صغيرة أو محدودة، ويتم إنشاء النموذج بناءً على سيناريوهات استخدام محددة.

> يُعدُّ هذا النموذج الخيار الأفضل للمراحل المتوسطة لتطوير المُنتَج، حيث يتم الانتقال من النموذج الأولي منخفض الدقة إلى النموذج الأولي متوسط الدقة.

شكل 2.21: نماذج أولية متوسطة الدقة

النموذج الأولي عالي الدقة
High-Fidelity Prototype

> غالبًا ما يتم الخلط بين هذا النموذج والمُنتَج النهائي نظرا لتشابههما في المظهر وفعالية بعض وظائف النظام في هذا النموذج، وتُعدُّ النماذج عالية الدقة الأفضل في إعطاء تجربة واقعية مماثلة للمُنتَج ووظائفه الفعلية.

> يتميز بالدقة من حيث عملية تقدير التكلفة والوقت المطلوبين.

> يدعم تحليل الأجزاء الأكثر تعقيدًا من المُنتَج في المراحل المتقدمة، حيث إن إظهار هذا النموذج في المراحل الأولية من النمذجة قد يُربك أصحاب المصلحة ولا يوفِّر المعرفة الأولية اللازمة.

شكل 2.22: نماذج أولية عالية الدقة

تعليمات النمذجة Modeling Instructions

تُعدُّ النماذج الأولية المناسبة أمرًا مهمًا للتحقق من صحة حلول التصميم للمشروع، ولذلك ستتعرف على بعض النصائح التي يجب أخذها في الاعتبار عند العمل مع النماذج الأولية:

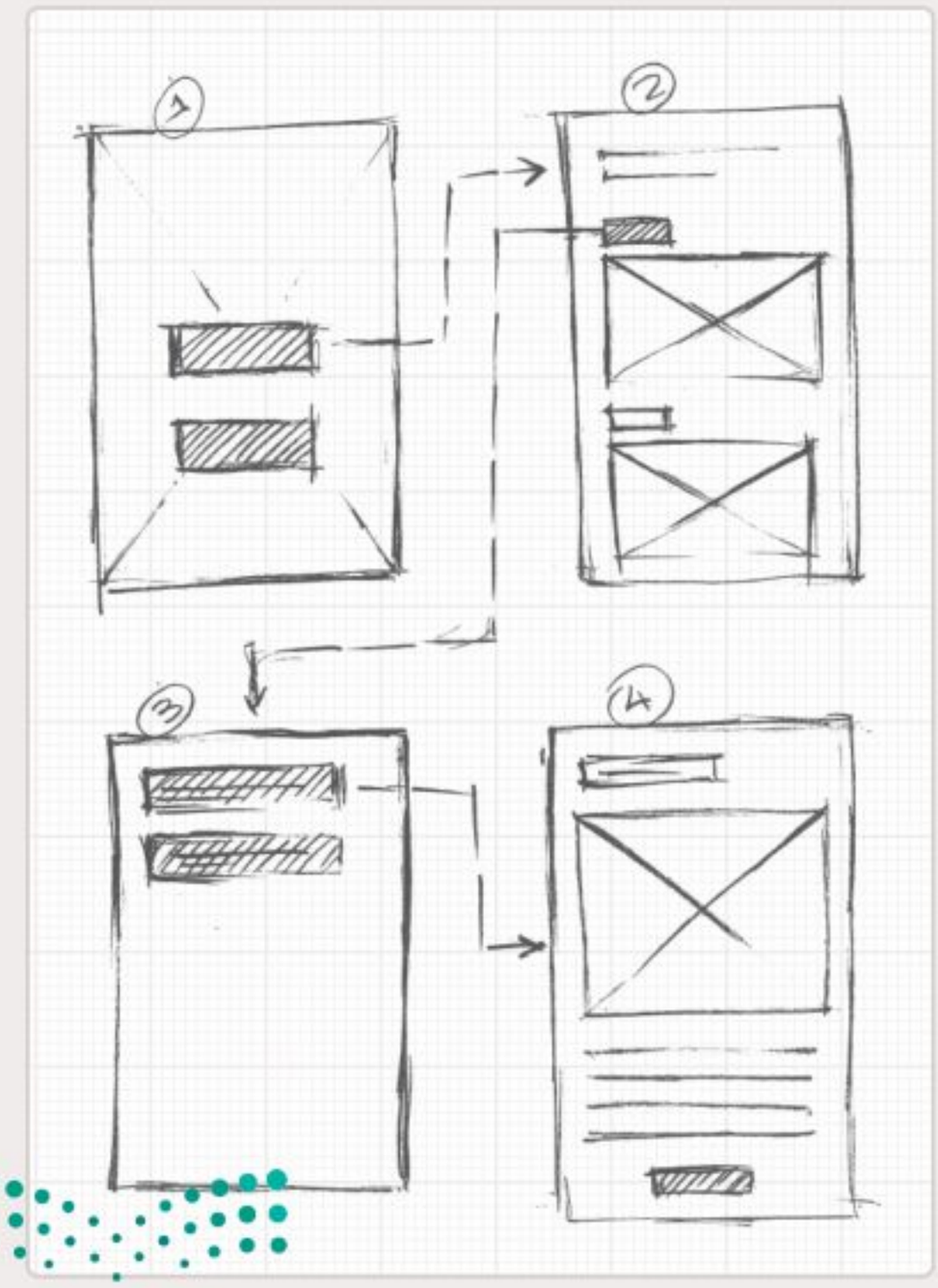
- استثمر الوقت في إنشاء النموذج دون الدخول في تفاصيل كثيرة.
- ضَعْ أهدافَ المُنتَج نصب عينيك أثناء العمل.
- ضَعْ المُستخدِم في اعتبارك أولًا.

سيناريو التطبيق Application Scenario

تتفاوت احتياجات المُستخدِمين المختلفين للتطبيقات، ولذلك فإن هذه التطبيقات يجب أن تأخذ في الاعتبار هذه الاختلافات، وأن تقدم واجهات المُستخدِم والوظائف الخاصة بها بما يناسب حاجات هؤلاء المُستخدِمين.

ستنشئ تطبيقًا لمساعدة السائحين على التنقل من خلال الشاشة ليتمكنوا من قراءة المعلومات حول المواقع السياحية المختلفة التي يمكنهم زيارتها في مدينتي الرياض وجدة.

سيبدو النموذج الأولي منخفض الدقة لتطبيق السياحة الذي يقدم قابلية الوصول كما يلي:



1. تتكون شاشة التطبيق الأولى من صورة وزرَّين، بحيث يمكن للمُستخدِم الضغط على الزر الأول الذي سينقل المُستخدِم للشاشة التالية، أما الزر الثاني فهو لتغيير اللغة من الإنجليزية إلى العربية.

2. تتكون شاشة التطبيق الثانية من صورتين لتمكين المُستخدِم من اختيار المدينة التي يريدها، وتعمل الصور أيضًا كزر ينقل المُستخدِم للشاشة التالية.

3. تعرض الشاشة الثالثة قائمة باثنين من المعالم المميزة لكل مدينة، ويعمل كل منهما كزرّ للانتقال إلى الشاشة التالية.

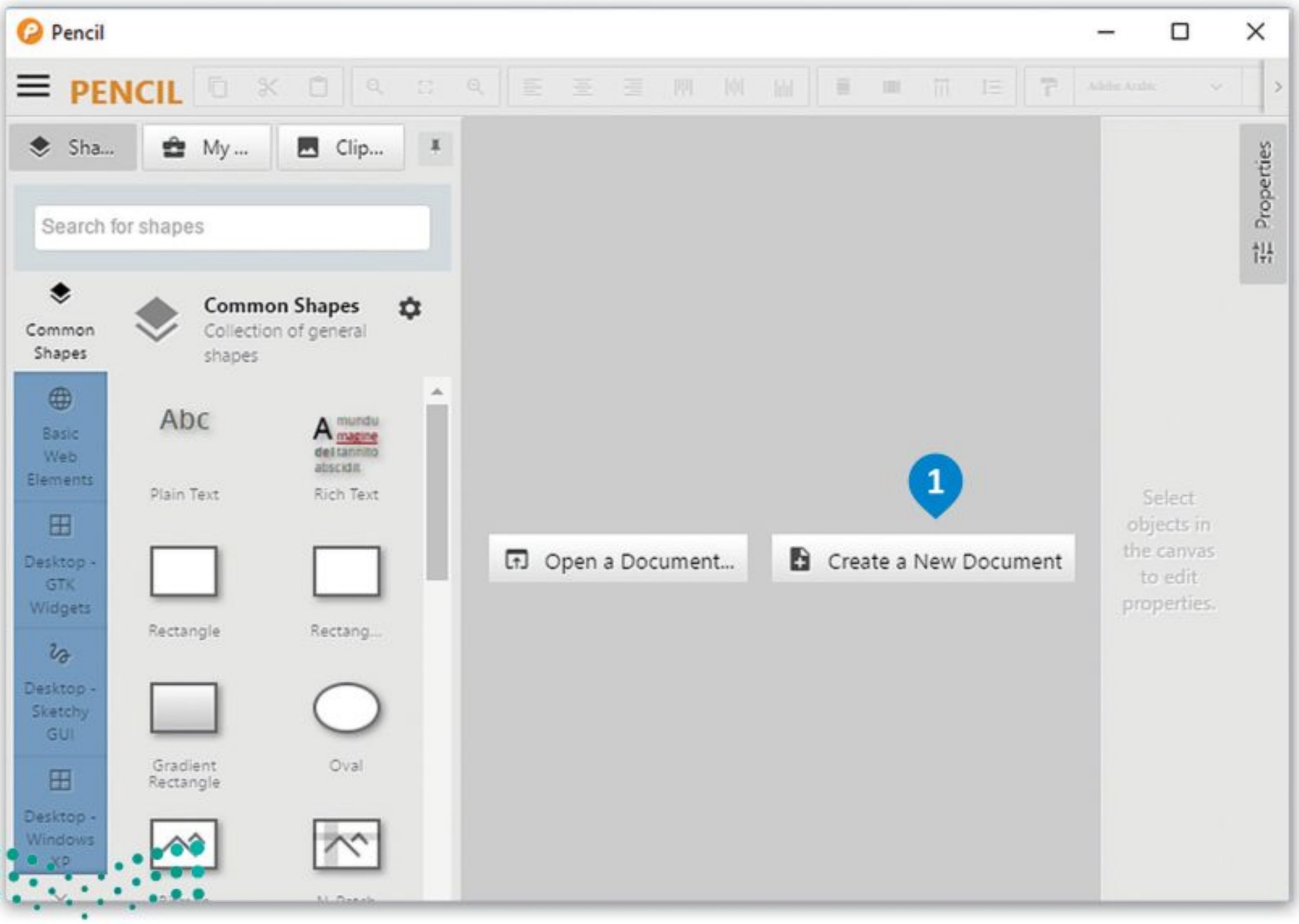
4. تعرض الشاشة الأخيرة صورة ووصفًا بسيطًا حول المَعلم المميز.

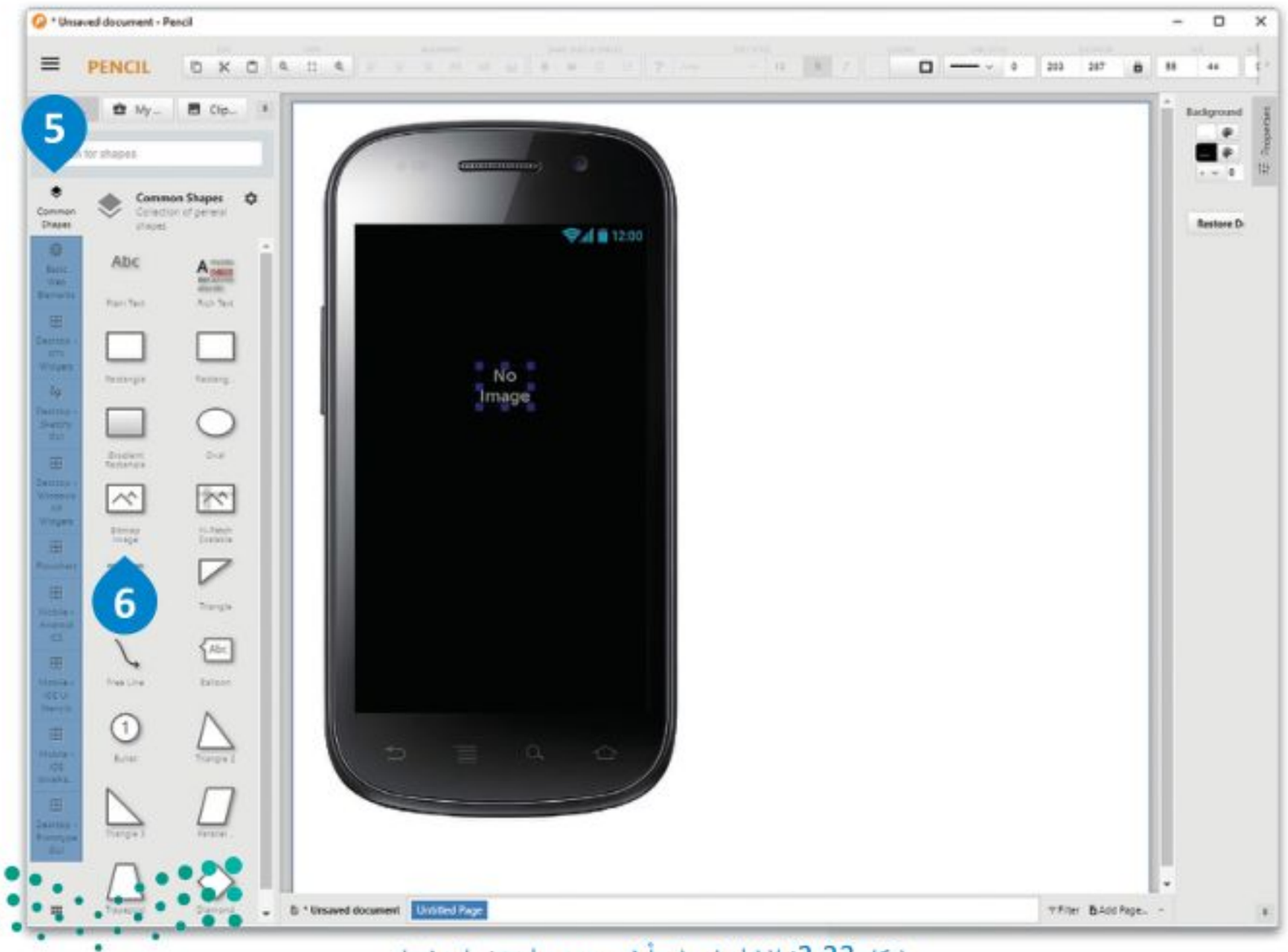
إنشاء النموذج الأولي باستخدام برنامج بنسل بروجكت
Creating the Prototype With the Pencil Project Software

يوفر برنامج بنسل بروجكت **واجهة مُستخدِم رسومية** (GUI) مفتوحة المصدر للنماذج الأولية لجميع الأنظمة الأساسية. استخدمت في الدرس السابق برنامج بنسل بروجكت لإنشاء المُخطَّط الانسيابي، وستُنشئ في هذا الدرس نموذجًا أوليًا متوسط الدقة للتّطبيق على الهاتف المحمول.

لإنشاء نموذج أولي جديد:

> افتح Pencil Project (بنسل بروجكت)، ثم اضغط على Create a New Document (إنشاء مستند جديد). 1
> من لوحة Shapes (الأشكال)، اضغط على Mobile – Android ICS (الهاتف المحمول- أندرويد ICS) لإضافة شكل. 2
> اسحب شكل Phone (هاتف) وأفلته في لوحة التصميم. 3
> اسحب شكل Status Bar (شريط الإشعارات) وأفلته في الجزء العلوي من شاشة الهاتف بالشّكل الذي يبدو عليه في شاشة الهاتف الحقيقي. 4
> من لوحة Shapes (الأشكال)، اضغط على قسم Common Shapes (الأشكال الشائعة) لإضافة الشكل. 5
> اسحب وأفلت Bitmap Image (صورة نقطية) إلى لوحة التصميم لتحميل صورة. 6





شكل 2.23: إنشاء نموذج أولي جديد باستخدام بنسل

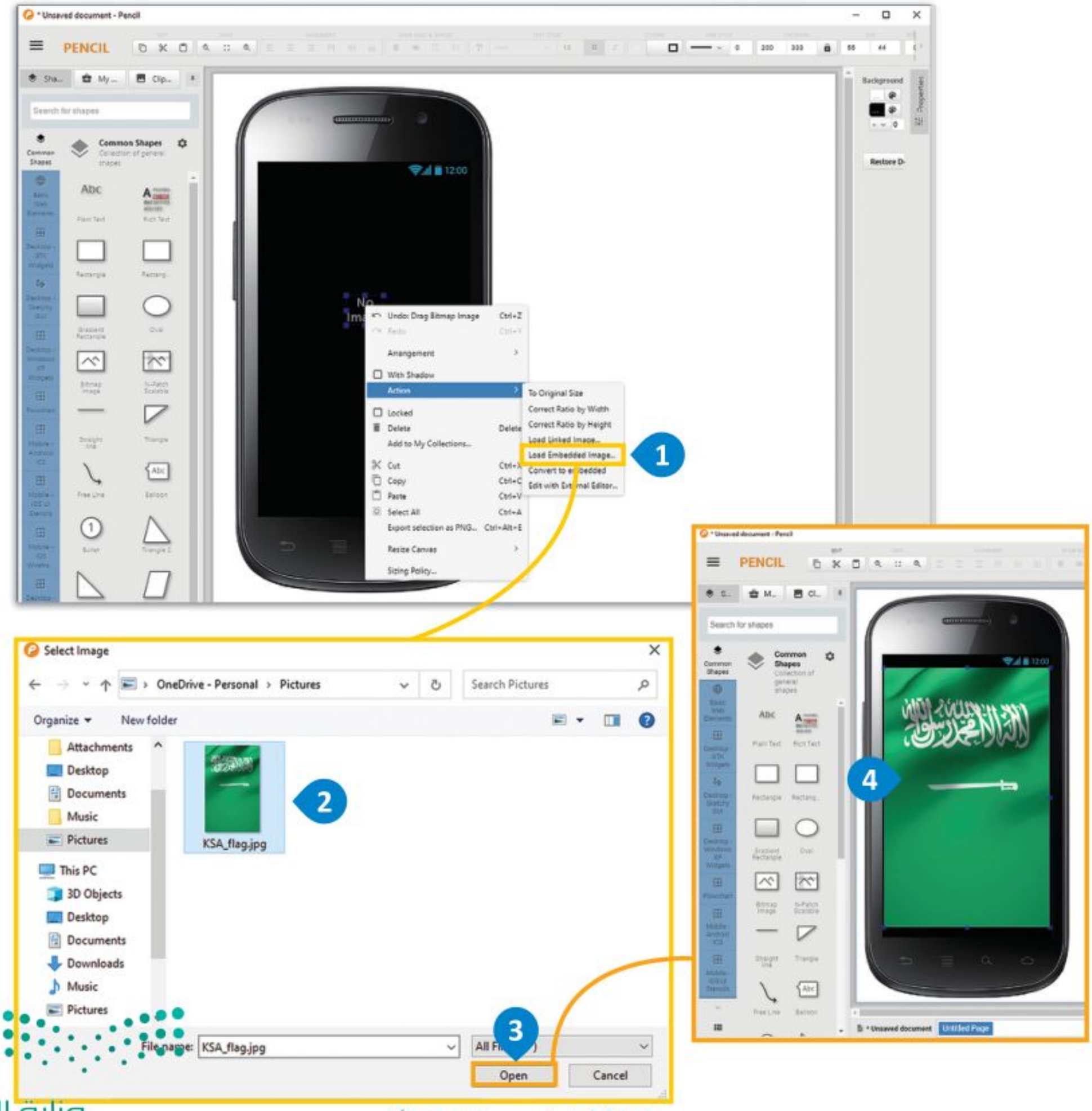
لإدراج صورة:

> اضغط بزر الفأرة الأيمن على شكل Bitmap Image (صورة نقطية)، ثم اختر Action (إجراء)، ثم Load Embeded Image (تحميل الصورة المُضمنة). 1

> اختر ملف الصورة. 2

> اضغط لتحميل الصورة من جهازك. 3

> اسحب الصورة وأفلتها في منتصف مُخطَّط الصورة لملاءمة شاشة الهاتف. 4



شكل 2.24: إدراج صورة في النموذج الأولي

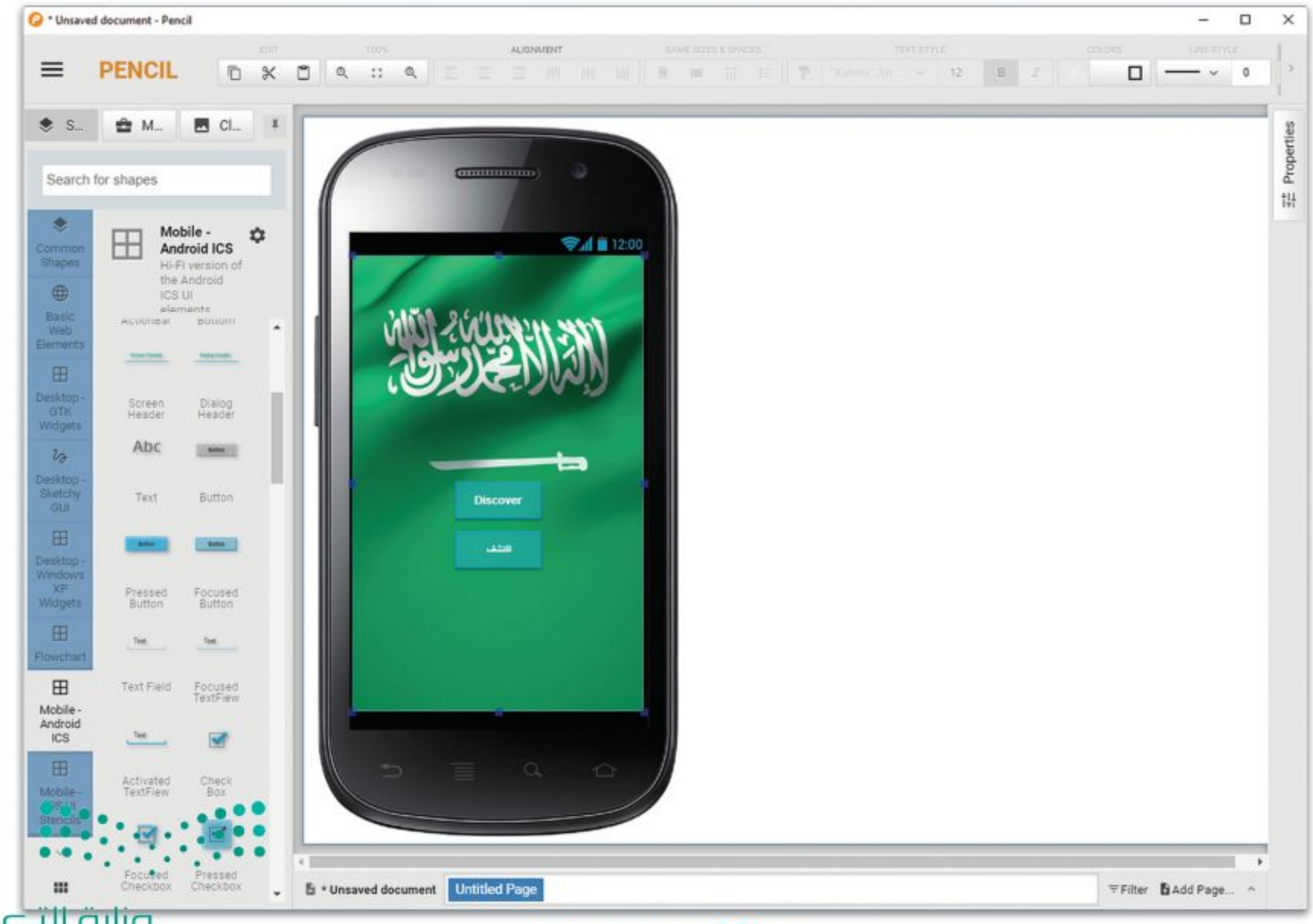
لإضافة زر:

> من قسم Mobile – Android ICS (الهاتف المحمول- أندرويد ICS)، اسحب شكل Focused Button (زر مُركّز) وأفلته في منتصف الشاشة. 1

> اضغط ضغطًا مزدوجًا وسَمِّ الزر Discover (اكتشف). 2

شكل 2.25: إضافة زر باللغة الانجليزية

كرِّر الخطوات التي اتبعتها عند اضافة الزر لإضافة زر ثاني عنوانه مكتوب باللغة العربية.



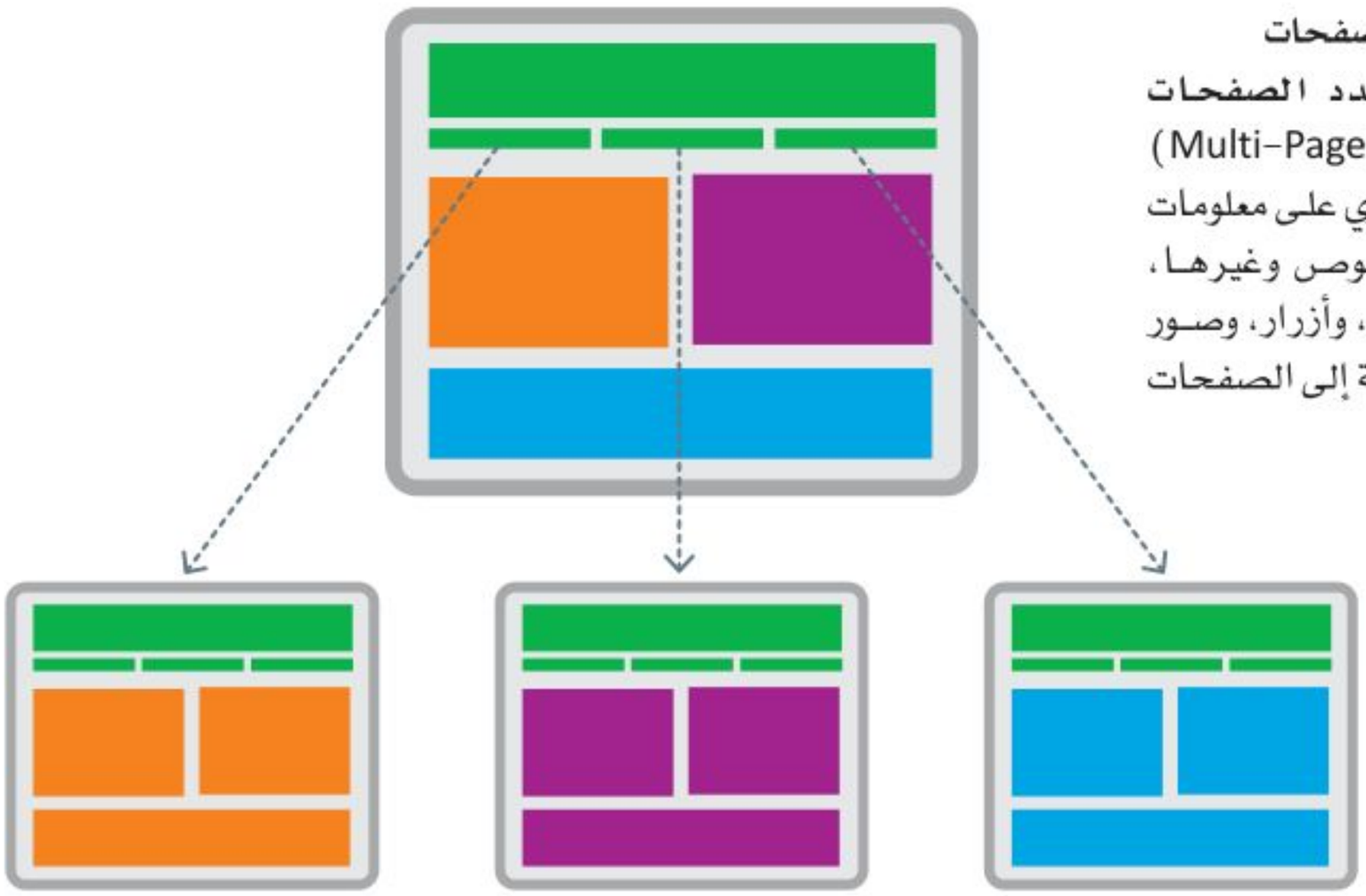
شكل 2.26: إضافة زر باللغة العربية

تطبيق متعدد الصفحات Multi-Page Application

لا يمكن للمُستخدِم التفاعل مع النموذج الأولي في بنسل بروجكت، ولذلك يجب عرض شاشات التطبيق المتعددة جنبًا إلى جنب بنفس الترتيب الذي تظهر به أثناء استخدام التطبيق الفعلي.

إنشاء تطبيق متعدد الصفحات

يتكون **التطبيق متعدد الصفحات** (Multi-Page Application -MPA) من عدة صفحات تحتوي على معلومات ثابتة مثل:صور، ونصوص وغيرها، وعلى روابط لنصوص، وأزرار، وصور وغيرها، وذلك بالإضافة إلى الصفحات الأخرى في التطبيق.



شكل 2.27: تطبيق متعدد الصفحات

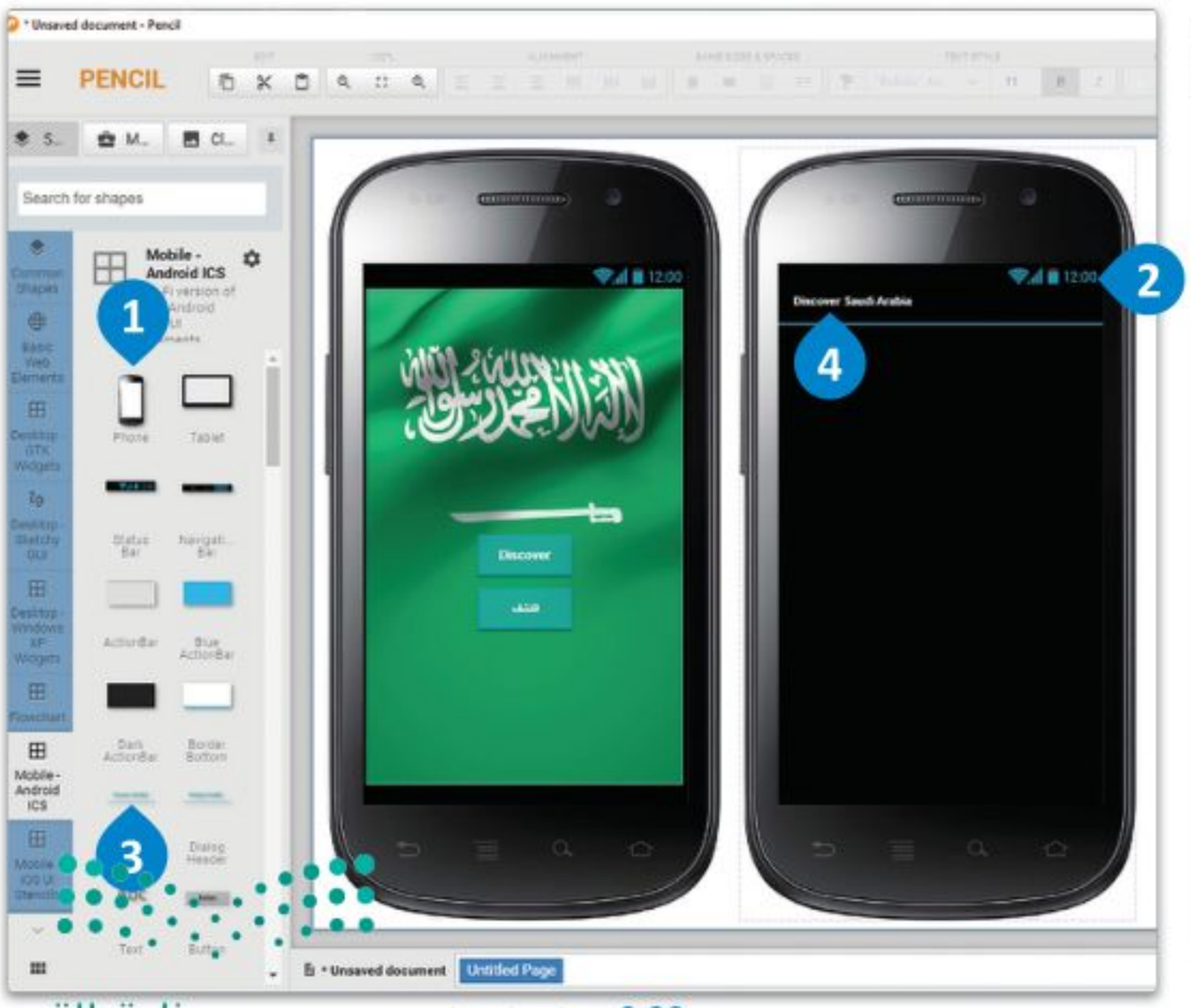
لإنشاء الشاشة الثانية للتطبيق:

> من قسم Mobile – Android ICS (الهاتف المحمول- أندرويد ICS)، اسحب وأفلت شكل Phone (هاتف) على يمين شكل الهاتف الأول. 1

> اسحب شكل Status Bar (شريط الإشعارات) وأفلته في الجزء العلوي من شاشة الهاتف كما يبدو في الهاتف الحقيقي. 2

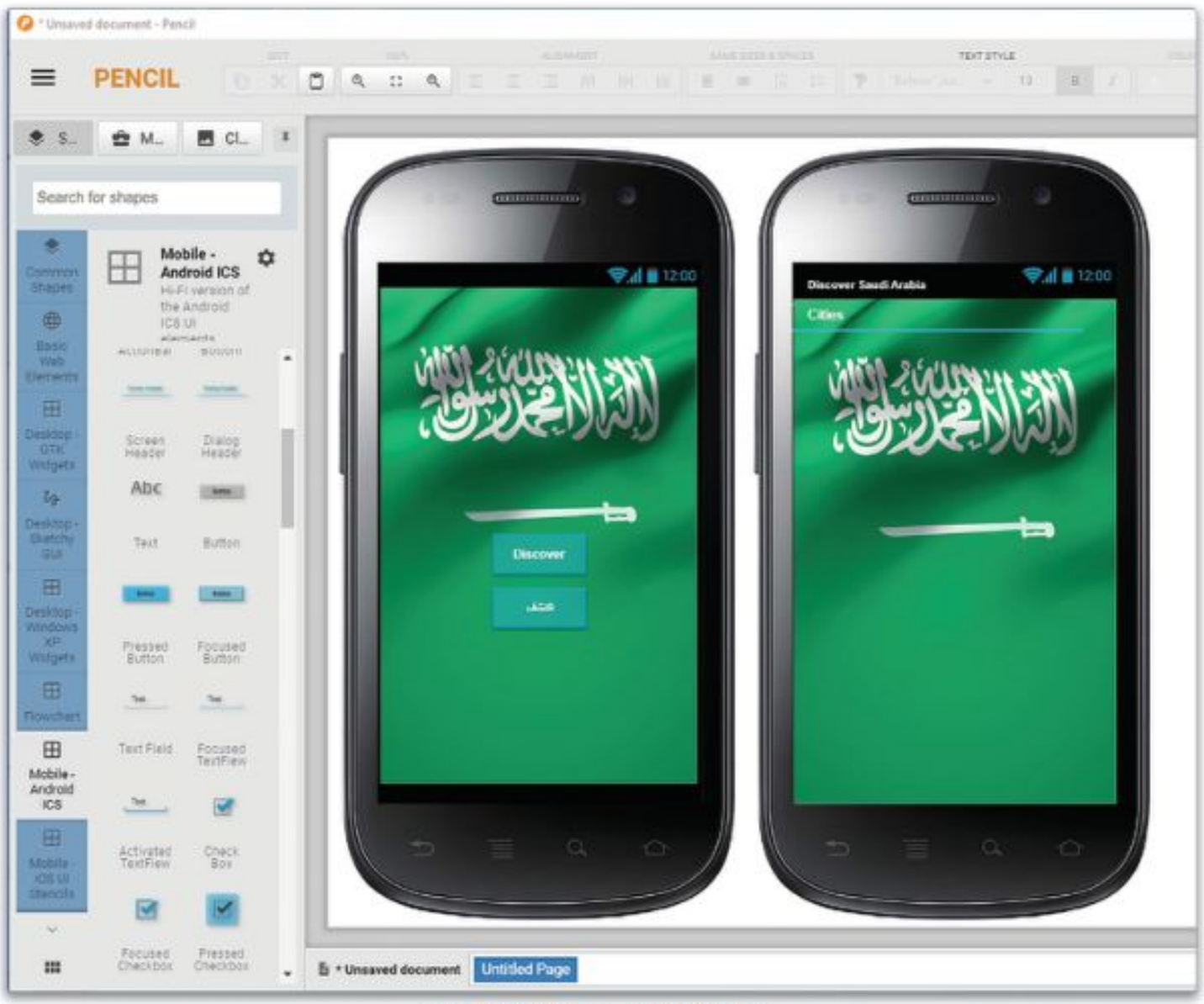
> أسفل شريط الحالة، أضف شكل Screen Header (ترويسة الشاشة)، 3 ثم اضغط ضغطًا مزدوجًا واكتب العنوان Discover Saudi Arabia (اكتشف المملكة العربية السعودية). 4

> غيِّر حجم النص من شريط Font Editor (مُحرِّر الخطوط) إلى 11. 5



شكل 2.28: إنشاء شاشة ثانية

كرِّر الخطوات التي اتبعتها عند إضافة الصورة وترويسة الشاشة لجعل الشاشة تبدو كما تظهر على اليمين.



شكل 2.29: إعداد الشاشة الثانية

عليك إضافة صورتين لمدينتي الرياض وجدَّة كما هو موضح.

شكل 2.30: إضافة صور المُدن

لإدراج شكل نصي:

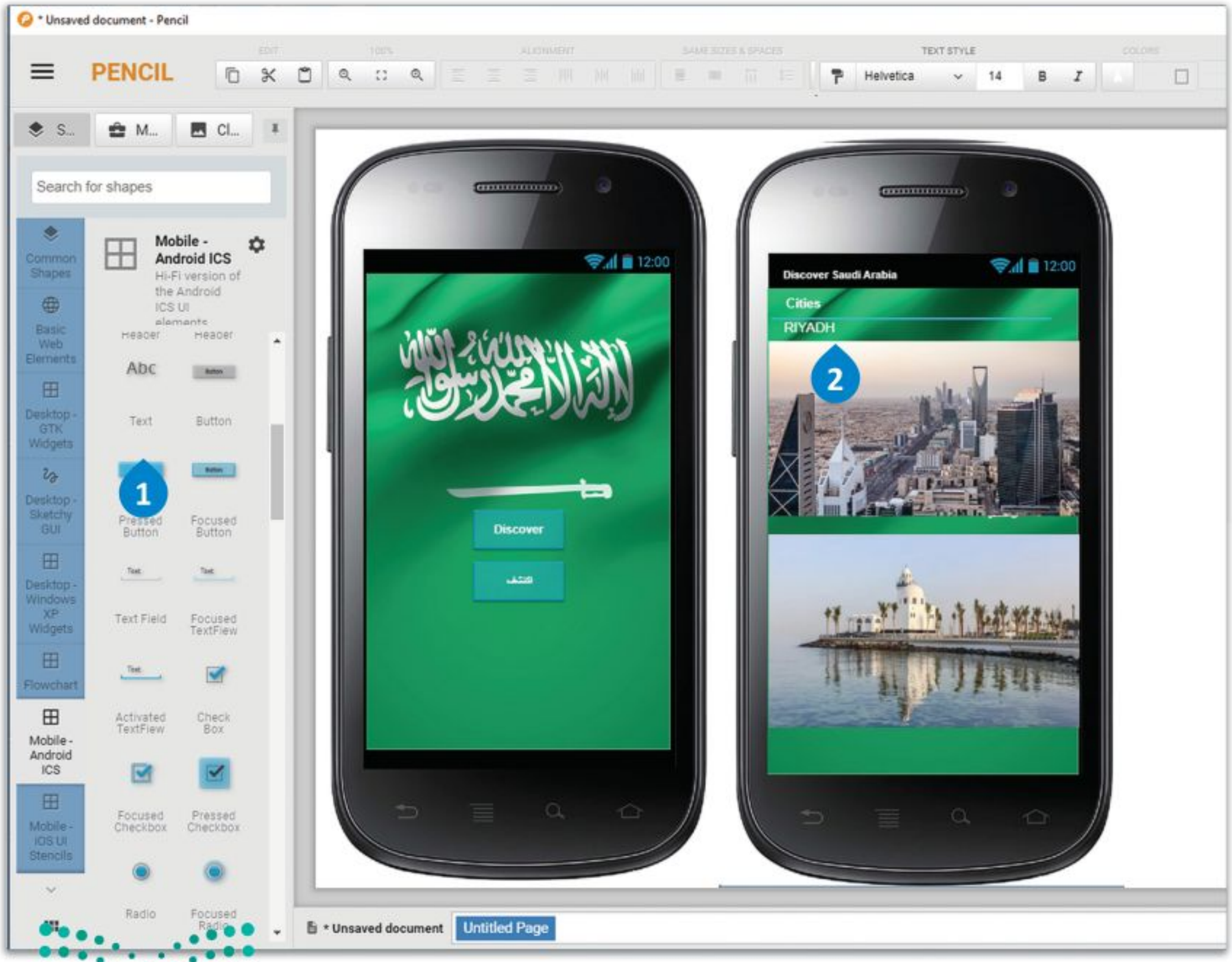
> من قسم Mobile – Android ICS (الهاتف المحمول- أندرويد ICS)، اسحب شكل Text (نص) وأفلته. 1

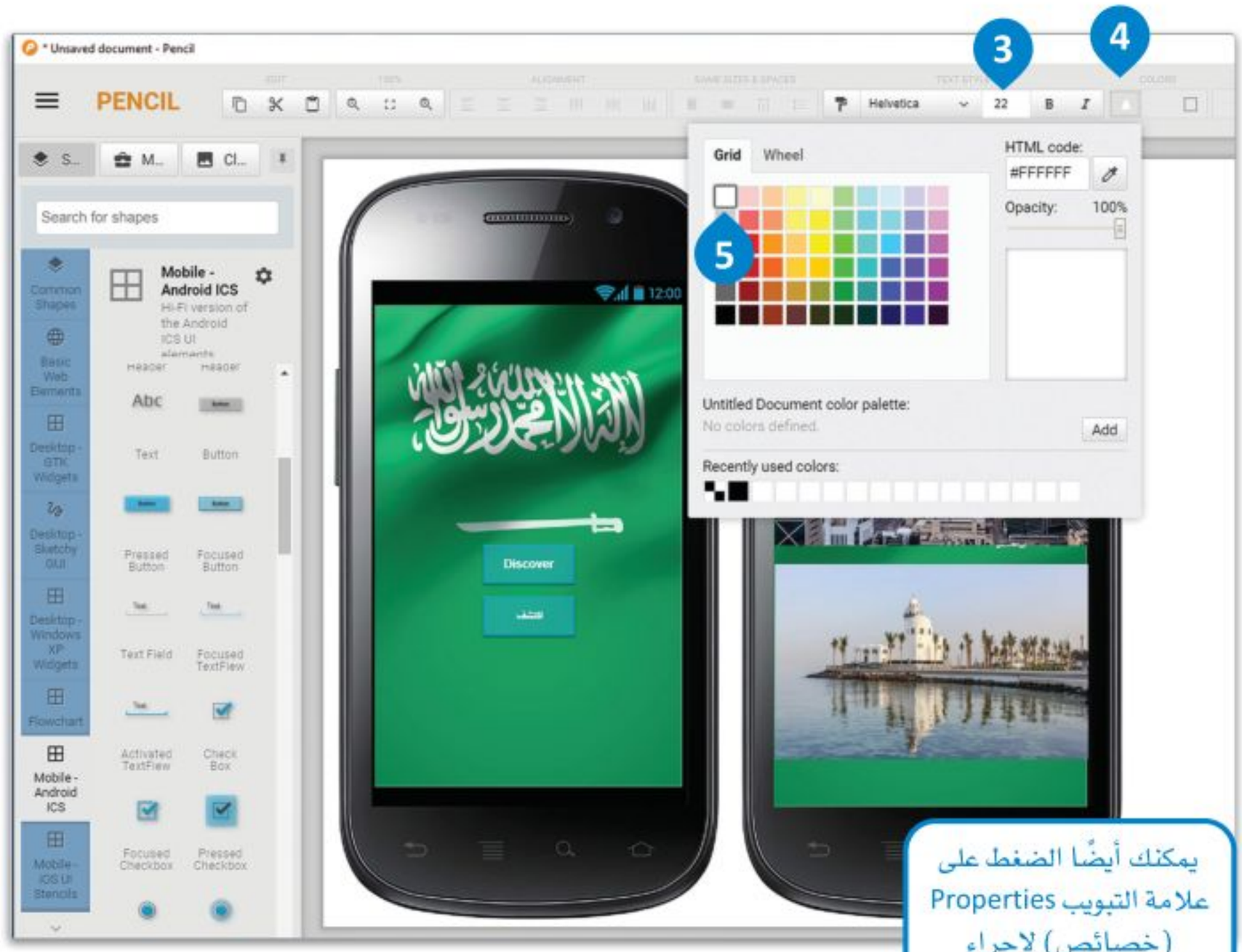
> اضغط ضغطًا مزدوجًا واكتب Discover Riyadh (اكتشف الرياض). 2

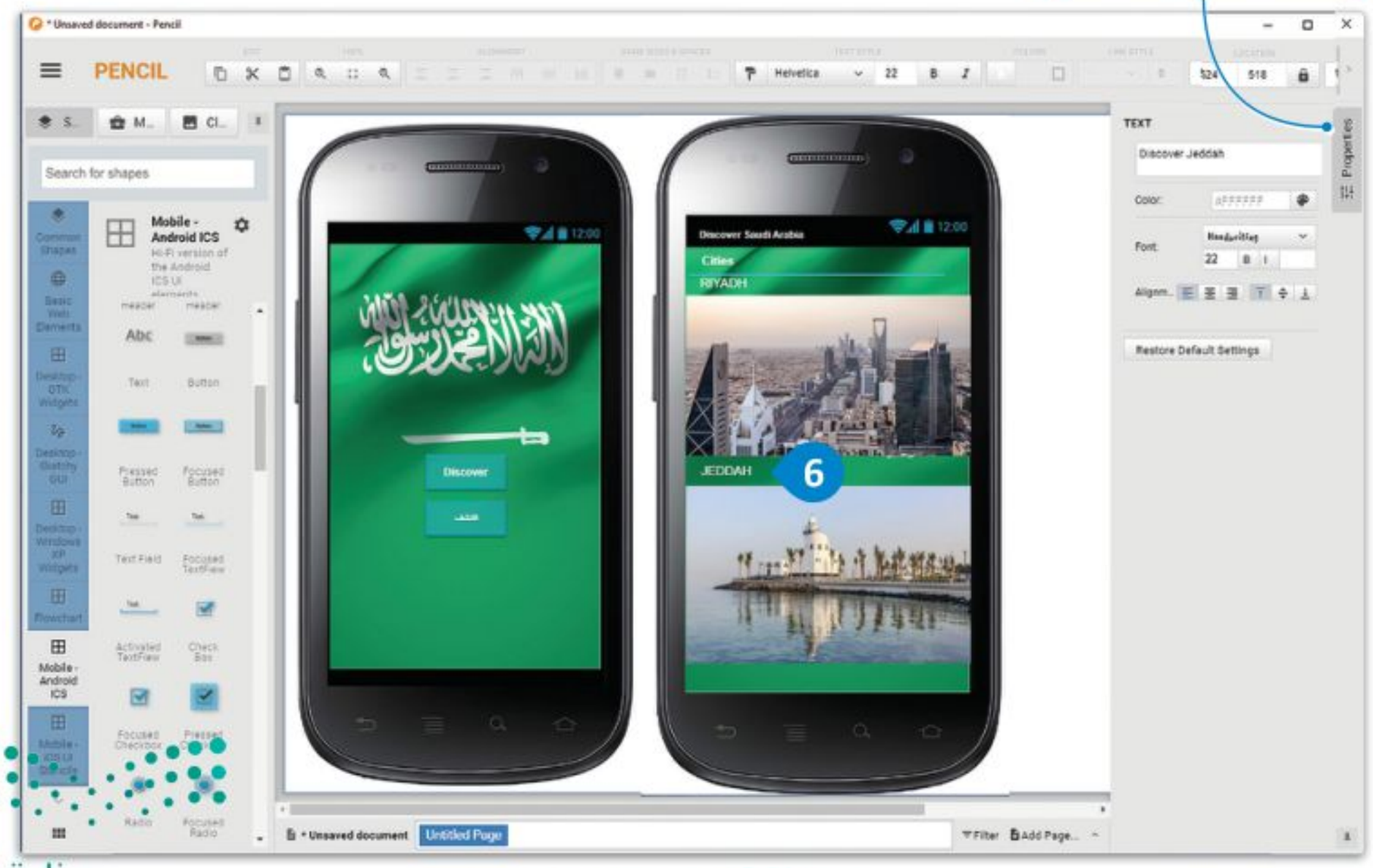
> غيِّر حجم النص من شريط Font Editor (مُحرر الخطوط) إلى 22. 3

> اضغط على Color Palette (لوحة الألوان)، 4 ثم اختر اللون بالرمز: #FFFFFF. 5

> كرِّر الخطوات لإضافة عنوان الصورة الثانية. 6







شكل 2.31: تسمية الصورة الثانية

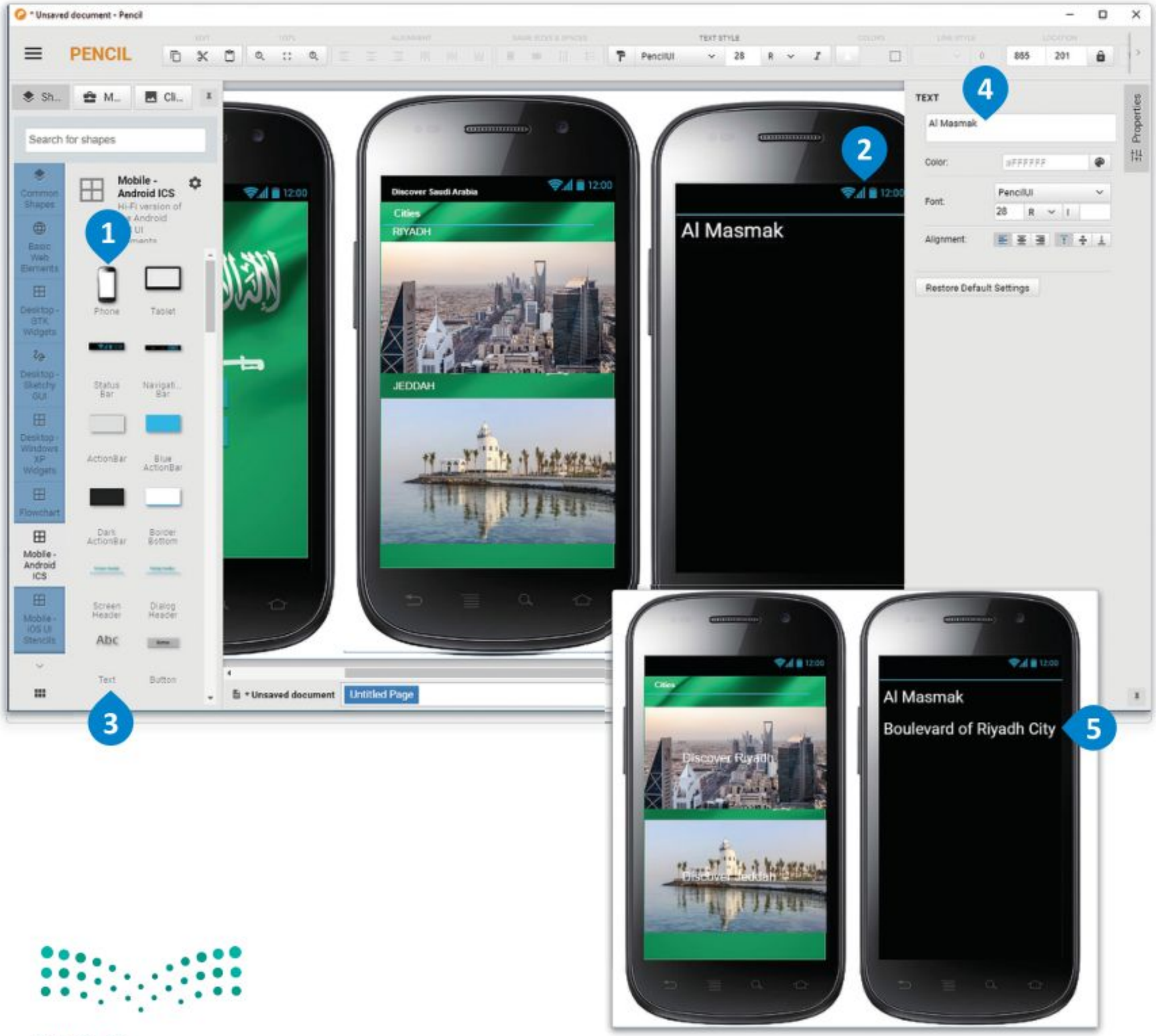
لإنشاء الشاشة الثالثة للتطبيق:

> من قسم Mobile – Android ICS (الهاتف المحمول- أندرويد ICS)، اسحب وأفلت شكل Phone (هاتف) على يمين شكل الهاتف الأول. 1

> اسحب شكل Status Bar (شريط الإشعارات) وأفلته في الجزء العلوي من شاشة الهاتف كما يبدو على شاشة الهاتف الحقيقي. 2

> اسحب وأفلت شكل Text (نص)، 3 ثم من علامة التبويب Properties (خصائص) غيّر لون النص واكتب Al Masmak (المصمك). 4

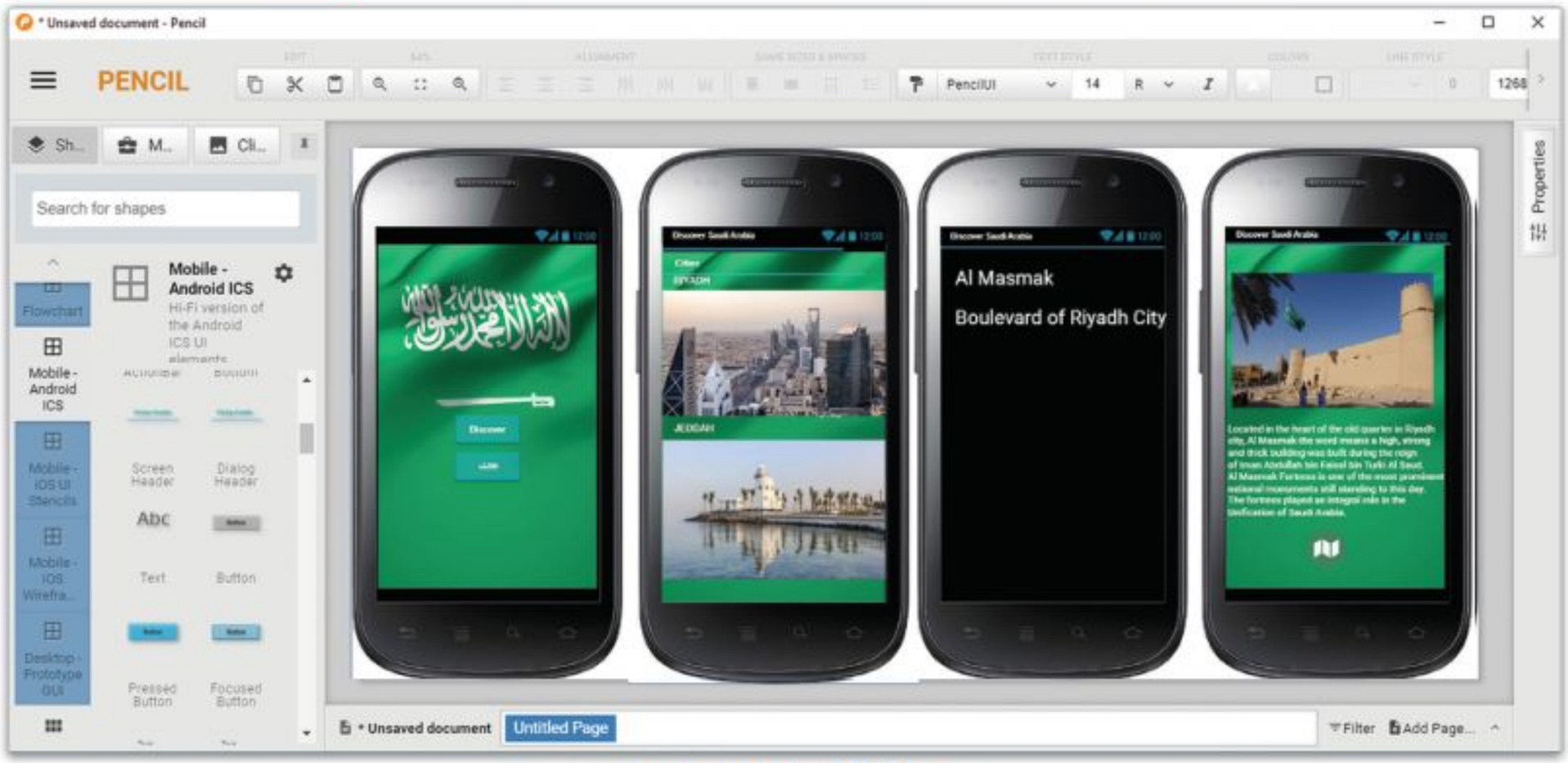
> كرر الخطوتين الأخيرتين لكتابة Boulevard of Riyadh City (بوليفارد مدينة الرياض). 5



شكل 2.32: إنشاء شاشة ثالثة

في الختام، أنشئ الشاشة النهائية التي ستعرض صورة ووصفًا موجزًا حول قصر المصمك.

لا تنسَ حفظ مشروعك عند الانتهاء.



شكل 2.33: إنشاء الشاشة النهائية

دور المُستخدِمين في النمذجة The Role of Users in Prototyping

من المهم أن يقوم المُستخدِمون باستعراض النموذج الأولي بعد الانتهاء من إنشائه، ولتسهيل عملية النمذجة الأولية يجب على مُحلِّل النظام أن ينقلَ الغرض من النموذج الأولي إلى المُستخدِمين بوضوح، مع التركيز على أن أهمية النماذج الأولية تَكمُن في تفاعل المُستخدِم معها وتقديم التغذية الراجعة والملاحظات.

أفضل الاستراتيجيات للحصول على تغذية راجعة على النماذج الأولية: Best Strategies for Getting Feedback on Prototyping:

- ابحث عن طرق متعددة للحصول على تعليقات من المُستخدِمين كإجراء المقابلات الشخصية مثلًا، وذلك للتعرف على آرائهم أثناء استخدام النموذج الأولي.
- اختبر نماذجك الأولية على الأشخاص المناسبين، وإذا كنت في المراحل الأولى لتصميم مشروعك وأردت الحصول على بعض الملاحظات والتغذية الراجعة البسيطة أو المتقدمة، فاختبار نماذجك الأولية على زملائك في الفريق قد يكون كافيًا.
- تأكَّد مما ستختبره واطرح الأسئلة المناسبة.
- كُنْ مُحايدًا عند عرض أفكارك، وتجنب التحيُّز تجاه فكرتك، وحاول التعرف على الخطأ إذا واجهت أي ردود فعلٍ سلبية.
- يمكنك تخطي سيناريو الاختبار الأصلي للتكيف مع بيئة الاختبار، وذلك للحصول على أفضل ردود أفعال من قِبل المُستخدِمين.
- اسمح للمُستخدِم بتقديم كافة الأفكار البناءة والنقد بناءً على ملاحظتهم لنموذجك الأولي، فمن شأن ذلك أن يسهم في تحسين التطبيق.

بعد الحصول على التغذية الراجعة حول التطبيق من المُستخدِمين، يجب على مُحلَّل النظام تعديل تصميمات الشاشة الرئيسة وفقًا لآراء المُستخدِمين حول النموذج الأولي.

تمرينات

1 طابق الأوصاف مع النماذج الأولية المناسبة.

1 النموذج الأولي عالي الدقة	○ يتم استخدامه في المراحل المتوسّطة لتطوير المُنتَج.
2 النموذج الأولي منخفض الدقة	○ يُمثل وظائف النظام ويركز عليها أكثر من المظهر.
3 النموذج الأولي متوسط الدقة	○ هو النموذج الأولي الأقرب لما سيبدو عليه المُنتَج النهائي.
	○ يمكن أن يكون مُكلفًا ويستغرق وقتًا طويلًا.
	○ يمكن إجراء التغييرات عليه بسهولة وبسرعة.
	○ يمكن إنشاؤه على الورق.

2 ما فوائد النمذجة الأولية؟

3 ما النصائح التي يجب اتباعها عند إعداد النموذج الأولي؟

4 عَدِّدْ طرق النمذجة الثلاث؟

5 أكمل النموذج الأولي لتطبيق زيارة المملكة العربية السعودية (Visit Saudi Arabia).

1. افتح النموذج الأولي للتطبيق باستخدام برنامج بنسل بروجكت.
2. أضف صفحةً جديدةً إلى المشروع.
3. أنشئ شاشةً لعرض أحد المعالم المميزة في مدينة الرياض.
4. أنشئ ثلاث شاشات جديدة لمدينة جدة، بالطريقة التي قمت فيها بذلك لمدينة الرياض، وذلك كالتالي:
   - شاشة واحدة لعرض اثنين من المعالم المميزة لمدينة جدة.
   - شاشتين لعرض كلٍ من المَعْلَمَين السابقين لمدينة جدة.

المشروع

1

ستواصل العمل على تطبيق الهاتف المحمول الخاص بالسياحة والذي يتضمن معلومات حول رؤية المملكة العربية السعودية 2030 الذي بدأته في الوحدة السابقة.

2

استخدم برنامج بنسل بروجكت لإنشاء مُخطَّط شجري يمثل كيفية تنظيم مكونات التطبيق الذي ستنشئه، والصفحات التي سيحتويها ومكوّنات كل صفحة.

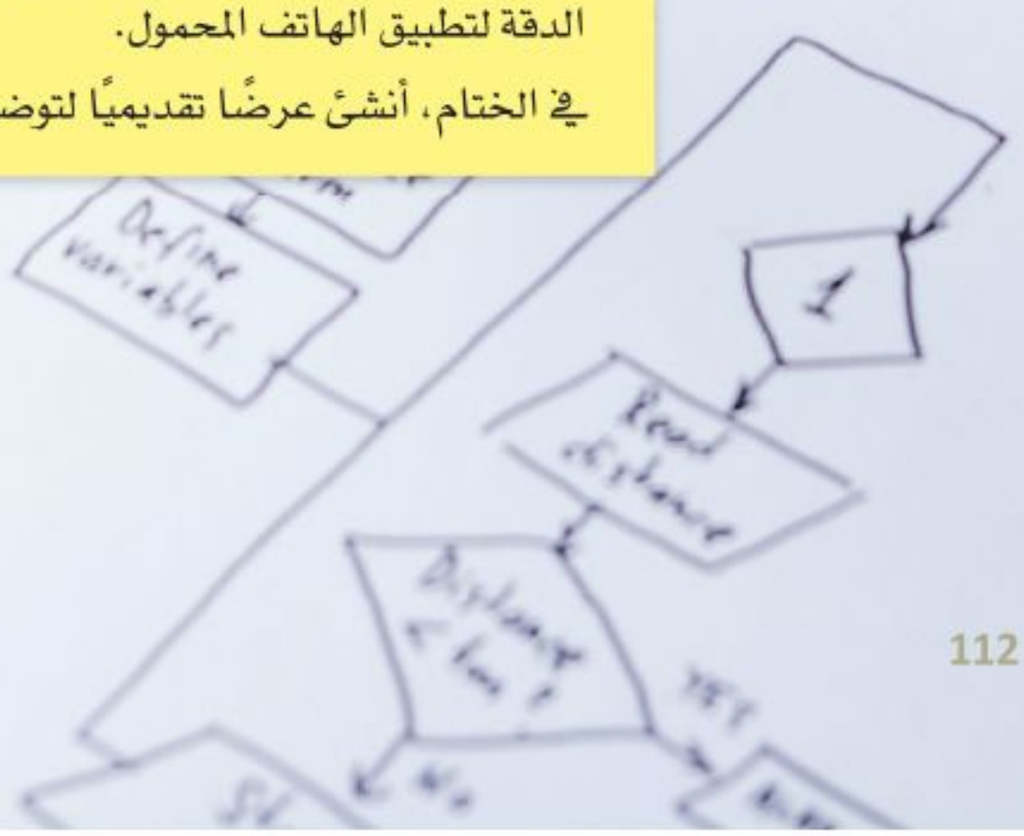
3

بعد ذلك، عليكَ أن تنشئ نموذجًا أوّليًّا منخفض الدقة لتطبيقك باستخدام الورق والقلم الرصاص لرسم شاشات تطبيقك.

4

في المرحلة التالية، استخدم برنامج بنسل بروجكت لتنشئ نموذجًا أوليًا متوسط الدقة لتطبيق الهاتف المحمول.

في الختام، أنشئ عرضًا تقديميًا لتوضيح مشروعك.



ماذا تعلّمت

> التمييز بين المُخطَّطات في مرحلة التحليل.

> إنشاء رسم تخطيطي لسير العمل باستخدام برنامج بنسل بروجكت.

> تصميم نموذج أولي باستخدام برنامج بنسل بروجكت.

المصطلحات الرئيسة

مُتطلبات غير وظيفية	Non-Functional Requirement
عملية	Process
نمذجة أولية	Prototyping
مُخطَّط شجري	Tree Diagram
مُخطَّط حالة الاستخدام	Use Case Diagram
تصميم تجربة المُستخدِم	User Experience (UX) Design
تصميم واجهة المُستخدِم	User Interface (UI) Design
مُخطَّط شبكي	Wireframe Diagram
سير العمل	Workflow

مُخطَّط	Diagram
مُخطَّط انسيابي	Flowchart
مُتطلبات وظيفية	Functional Requirement
نموذج عالي الدقة	High-Fidelity Prototype
التفاعل بين الإنسان والحاسب	Human-Computer Interaction (HCI)
نموذج منخفض الدقة	Low-Fidelity Prototype
نموذج متوسط الدقة	Medium-Fidelity Prototype
تطبيق متعدد الصفحات	Multi-Page Application (MPA)

3. تطوير التطبيقات باستخدام مخترع التطبيقات

سيستخدم الطالب في هذه الوحدة مخترع التطبيقات MIT (MIT App Inventor) لتطوير تطبيق تفاعلي حقيقي للهاتف المحمول يهدف إلى تشجيع السياحة في المملكة العربية السعودية. وسيستخدم النموذج الأولي الذي أنشأه في الوحدة السابقة لتصميم واجهة المُستخدِم ثم يبرمج التطبيق.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يَستخدِم نموذجًا أوّليًّا شبكيًّا لإنشاء واجهة المُستخدِم.
- يُصمِّم واجهة المُستخدِم للتطبيق.
- يجمع ويُطبِّق الملاحظات حول النموذج الأولي لتحسين التطبيق.
- يُطوِّر التَطبيق بناءً على مجموعة من المواصفات.
- يُثري تطبيق الهاتف الذكي بالمحتوى.
- يُعزِّز تطبيق الهاتف بمُكوِّنات تجربة المُستخدِم التفاعلية.

الأدوات

- مخترع التطبيقات MIT (MIT App Inventor)

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

مقدمة إلى مخترع التطبيقات MIT

تطوير تطبيقات الهاتف المحمول
Developing Mobile Applications

تشبه عملية تصميم تطبيقات الهاتف المحمول وتطويرها عملية تطوير تطبيق موقع الويب أو تطبيق حاسب مكتبي.

جدول 3.1: أمثلة على تطبيقات الهواتف الذكية

تطبيقات البريد الإلكتروني.
تطبيقات التواصل الاجتماعي.
تطبيقات التواصل والمراسلة الفورية.
تطبيقات الخرائط.
التطبيقات الإدارية والحكومية كتطبيقات الوزارات أو المستشفيات أو المدارس.
ألعاب الهاتف المحمول.

تطبيق الهاتف المحمول (Mobile Application):

يُعدُّ تطبيق الهاتف المحمول نوعًا من البرامج التطبيقية المُصمَّمة للعمل على الأجهزة المحمولة، كالهواتف الذكية والأجهزة اللوحية.

شكل 3.1: أنواع تطبيقات الأجهزة المحمولة

مراحل إنشاء تطبيقات الهواتف الذكية Stages of Creating Smartphone Applications

التحليل والتصميم:

في البداية، حدِّد فكرة التطبيق والهدف منه، وفئات المُستخدِمين المستهدفة، ثمّ أنشئ له مُخططًا يدويًا يتضمن واجهات المُستخدِم المختلفة، ويوضِّح طريقة ارتباط هذه الواجهات ببعضها.

التطوير:

استخدم أحد برامج تطوير تطبيقات الهواتف الذكية لتنفيذ التصميم الذي أنشأته في المرحلة السابقة.

الاختبار:

اختبر التطبيق وعالج أي مشاكل تظهر في البرمجة أو التصميم، ثم أضف اللمسات النهائية لعملك.

النشر والتسويق:

احصل على الموافقة على نشر التطبيق وقم بتحميله إلى متجر التطبيقات.

وزارة التعليم
Ministry of Education
2023 - 1445

مزايا استخدام مخترع التطبيقات MIT
Advantages of Using MIT App Inventor

تمتاز هذه الأداة بالسهولة والسرعة في عملية التطوير، حيث يُمكنك تطوير التطبيق في أقل من ساعة واحدة.

تُساعد في تطوير المهارات الإبداعية من خلال استخدام اللبنات البرمجية الأساسية، ومن ثَمّ التقليل من فرص ارتكاب الأخطاء البرمجية.

تُتيح سهولة مشاركة التطبيقات التي يتم إنشاؤها في هذا البرنامج.

يمكن من خلالها الوصول إلى العديد من الوظائف الأساسية في الهاتف المحمول، بما فيها المكالمات الهاتفية والرسائل النصية القصيرة ومُستشعرات المواقع والصوت والفيديو وغيرها.

يمكنها حفظ البيانات عبر منصات التخزين السحابية.

مخترع التطبيقات MIT (MIT App Inventor):

تُستخدم أداة مخترع التطبيقات MIT لتطوير تطبيقات الهواتف الذكية، حيث يمكن من خلالها إنشاء التطبيقات دون الحاجة إلى كتابة نصوص برمجية، باستخدام بيئة قائمة على اللبنات البرمجية المشابهة لبرنامج **سكراتش** (Scratch).

يمكن تطوير تطبيقات الأجهزة المحمولة بلغات برمجة خاصة مثل: **جافا** (Java) أو **كوتلن** (Kotlin) أو **سويفت** (Swift). يمكن أيضًا تجهيز التطبيق للتوزيع باستخدام أداة مخترع التطبيقات.

تطوير تطبيق خاص بالسياحة Developing a Tourism Application

ستطوِّر تطبيقًا للسياح الزائرين للمملكة العربية السعودية يسمح لهم بالبحث عن المعالم المميزة والأكثر جاذبية فيها، فعند اختيارهم لوِجْهَة معينة، ستظهر لهم قائمة بالمعالم المميزة في تلك الوِجْهَة، ثمّ يختارون أحدها ليتم عرض صورتها ووصفها. سيتم تطوير هذا التطبيق بناء على النموذج الأولي الشبكي الذي تم إنشاؤه في الوحدة السابقة، وستصمم في هذا الدرس جميع الشاشات الخاصة بهذا التطبيق، ثمَ ستقوم ببرمجتها في الدرس التالي.

الاختلافات بين النمذجة الأولية لواجهة المُستخدِم، وتطوير واجهة المُستخدِم
Differences between Prototyping UI and Developing UI

عند تصميم نماذج شبكية باستخدام أداة تصميم النماذج الأولية، يتم ترتيب ظهور العناصر والمُكوِّنات على الشاشة بشكل مبدئي، ثم يتم ترتيب هذه المُكوِّنات ديناميكيًا وفقًا للأداة المُستخدمة عند تطوير التطبيق الفعلي. يُستخدم مخترع التطبيقات MIT كما في معظم منصات التطوير **مُجمَّعًا للمُكوِّنات** (Container Components) لترتيب ومحاذاة المُكوِّنات الأخرى الموضوعة بداخله. ضع في اعتبارك أن طرق إنشاء واجهة المُستخدِم تختلف بين مرحلة إنشاء النماذج الأولية، ومرحلة التطوير الفعلي للتطبيق.

معلومة

تم تطوير مخترع التطبيقات (App Inventor) بواسطة قوقل (Google) عام 2010 وتتم إدارته الآن بواسطة معهد ماساتشوستس للتقنية (Massachusetts Institute of Technology - MIT).

وزارة التعليم
Ministry of Education
2023 - 1445

الانتقال من النموذج الأولي إلى التطبيق Transitioning from Prototype to Application

نظرًا لأنك أنشأت سابقًا نموذجًا شبكيًا أوليًا، فقد أصبحت تمتلكُ المعرفة بكيفية تصميم واجهة المُستخدِم للتطبيق، ويعني هذا أن وقت التطوير سيكون أقصر؛ لأنّ قرارات تصميم تجربة المُستخدِم وواجهة المُستخدِم قد تم اتخاذها سابقًا.

ستستخدم الأدوات التي يوفرها برنامج مخترع التطبيقات لتصميم التطبيق بشكلٍ مطابق قدر الإمكان للنموذج الأولي الشبكي للتطبيق، وتذكَّر أن النموذج الأولي الشبكي لا يمثل طريقة عرض التطبيق بصورته النهائية.

يتم تنفيذ تغييرات واجهة المُستخدِم والميزات الجديدة أثناء عملية تطوير التطبيق، حيث يمكن الحصول على التغذية الراجعة والملاحظات من خلال اختبار المُستخدِمين للتطبيق؛ فهم يقدّمون ملاحظاتهم حول النموذج الأولي الشبكي الذي أنشأته في الوحدة السابقة أثناء التجهيز لتطوير التطبيق، ويمكن استخدام هذه الملاحظات لإعادة تصميم النموذج الأولي مرةً أخرى، أو يمكن أخذها بالاعتبار أثناء مرحلة التطوير، وتنفيذها مباشرة في تلك المرحلة في برنامج مخترع التطبيقات.

النقاط الرئيسة المُستفادة من الملاحظات والتغذية الراجعة بخصوص النموذج الأولي هي كالتالي:

- يجب وضع مُكوِّنات شاشات المدن والمعالم المميزة داخل مُجمَّع بلون خلفية يختلف قليلًا عن صورة العَلَم.
- يجب أن يكون لشاشة المعالم المميزة مظهر متناسق مع صفحة المدن.
- من المفيد توفير طريقة لعرض الموقع الخاص بكل مَعْلَم يتم عرضه.

ستبدأ الآن في تطوير واجهة مُستخدِم التطبيق باستخدام النموذج الأولي كدليل إرشادي، كما ستأخذ الملاحظات والتعليقات المُستقاة من التغذية الراجعة للمُستخدِمين في عين الاعتبار.

البدء بإنشاء التطبيقات في مخترع التطبيقات
Start Building Apps with App Inventor

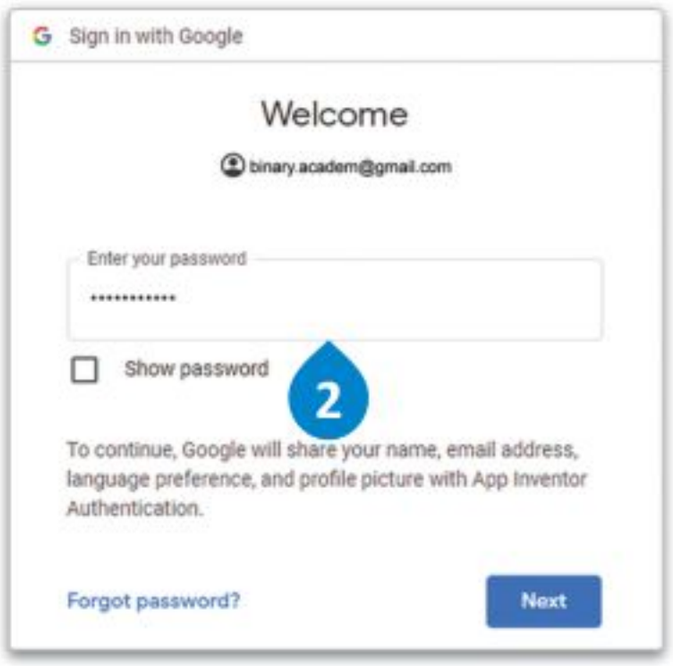
للبدء بإنشاء التطبيقات باستخدام مخترع التطبيقات MIT، يلزمك تسجيل الدخول إلى موقع مخترع التطبيقات باستخدام حسابك في قوقل.

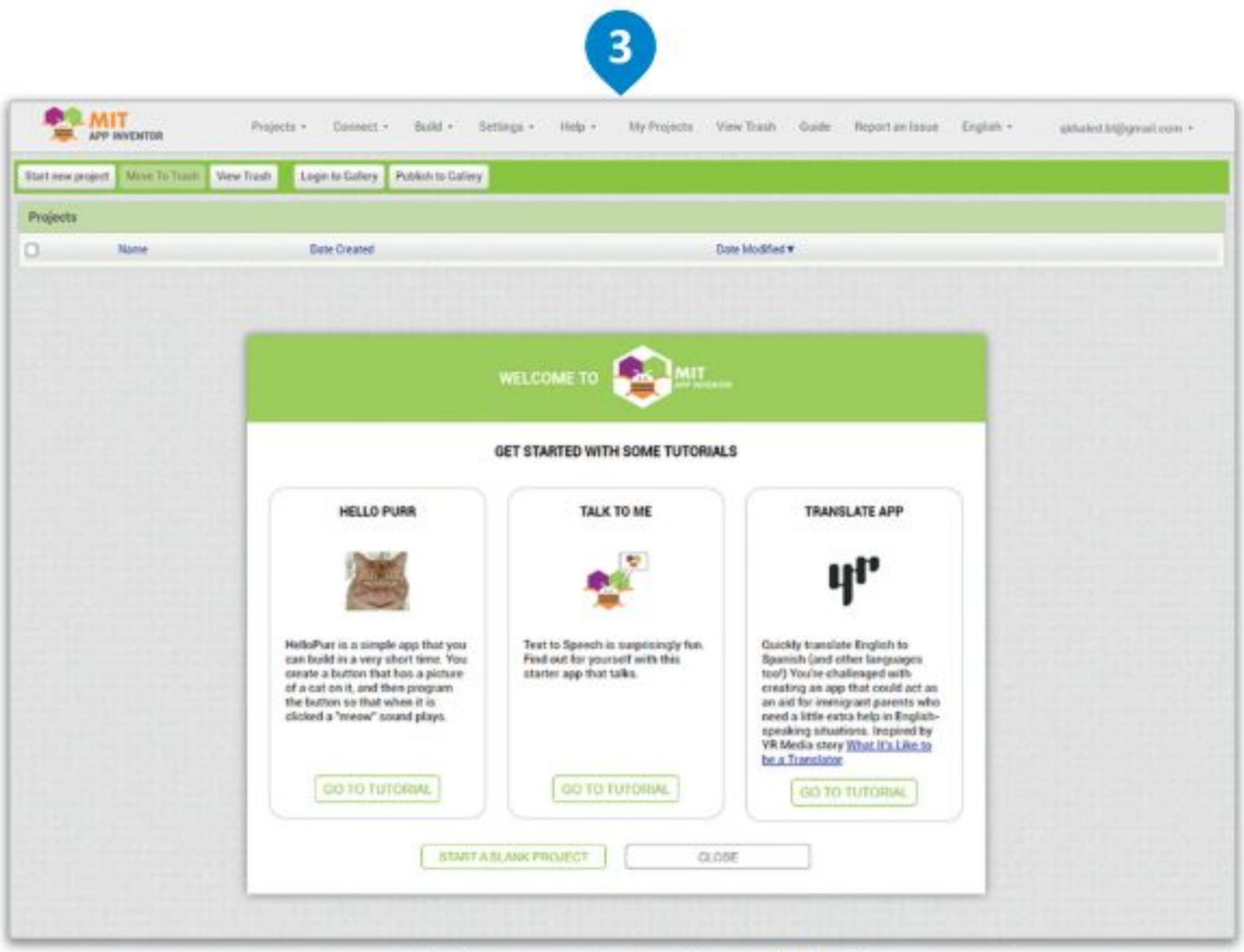
لبدء مخترع التطبيقات MIT:

> انتقل إلى موقع الويب appinventor.mit.edu واضغط على Create Apps (إنشاء تطبيقات). 1

> سجل الدخول باستخدام حساب قوقل الخاص بك. 2

> ستظهر الآن مساحة عمل MIT App Inventor (مخترع التطبيقات MIT). 3



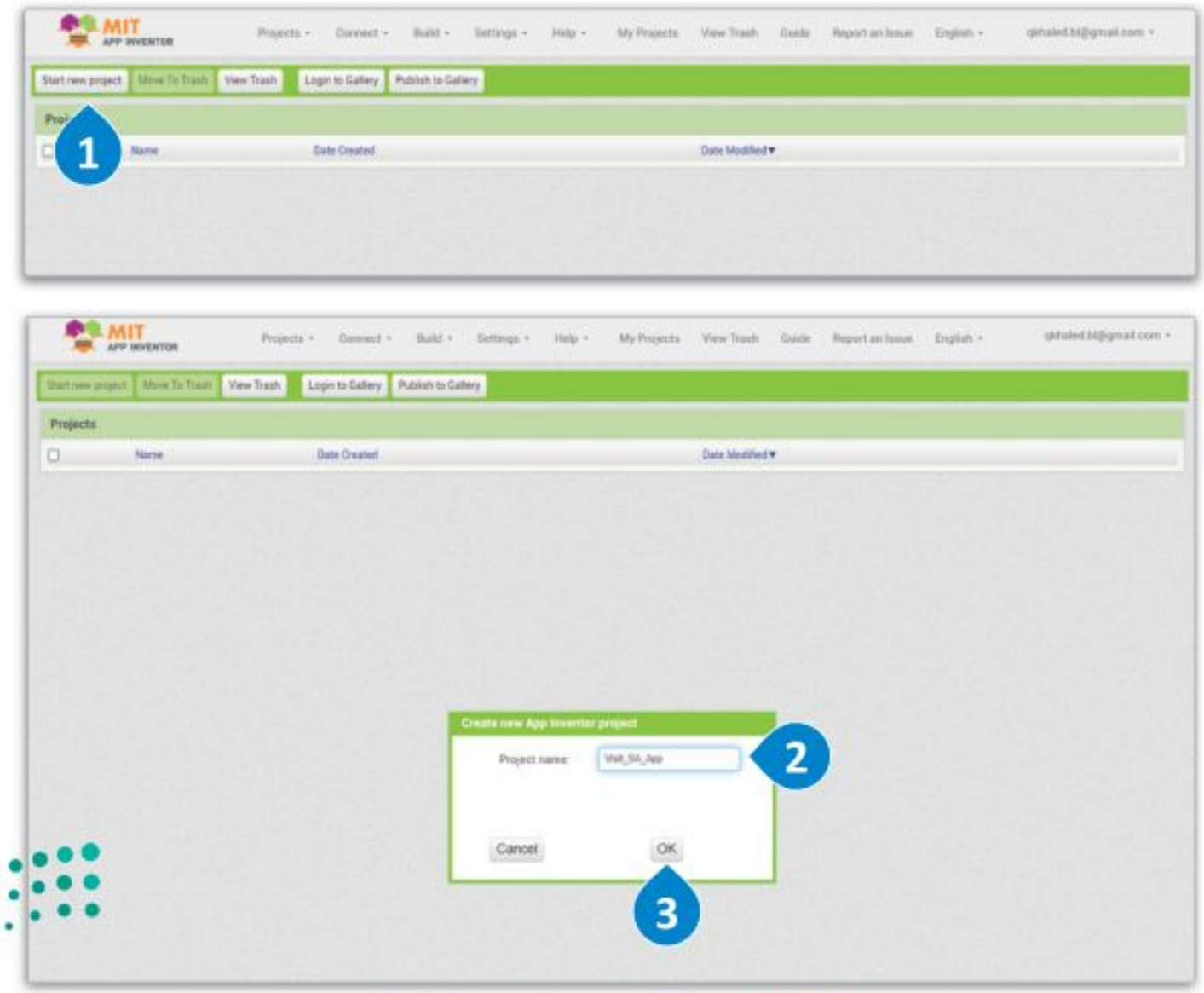


شكل 3.2: تسجيل الدخول إلى مخترع التطبيقات

لبدء مشروع جديد في مخترع التطبيقات MIT:

> اضغط على Start new project (بدء مشروع جديد). 1

> اكتب اسمًا لمشروعك، 2 واضغط على OK (موافق). 3



شكل 3.3: بِدء مشروع جديد في مخترع التطبيقات

واجهة مخترع التطبيقات The App Inventor Interface

يتم تقسيم واجهة مخترع التطبيقات إلى قسمين، القسم الأول خاص بالتصميم، بينما القسم الآخر مُخصص للّبِنات البرمجية. يمكن الوصول إلى هذين القسمين بالتبديل بين خياري **Designer** (المُصمِّم) و**Blocks** (اللبنات البرمجية) من أعلى يمين الشاشة. قسم المُصمِّم هو المكان الذي تُدرج فيه المُكوِّنات في الشاشة وتُغيّر خصائصها الأساسية، أما قسم اللبنات البرمجية فهو المكان الذي تقوم فيه ببرمجة تلك المُكوِّنات.

شكل 3.4: واجهة المُصمِّم في مخترع التطبيقات

جدول 3.2: المُكوِّنات الأكثر استخدامًا في مخترع التطبيقات

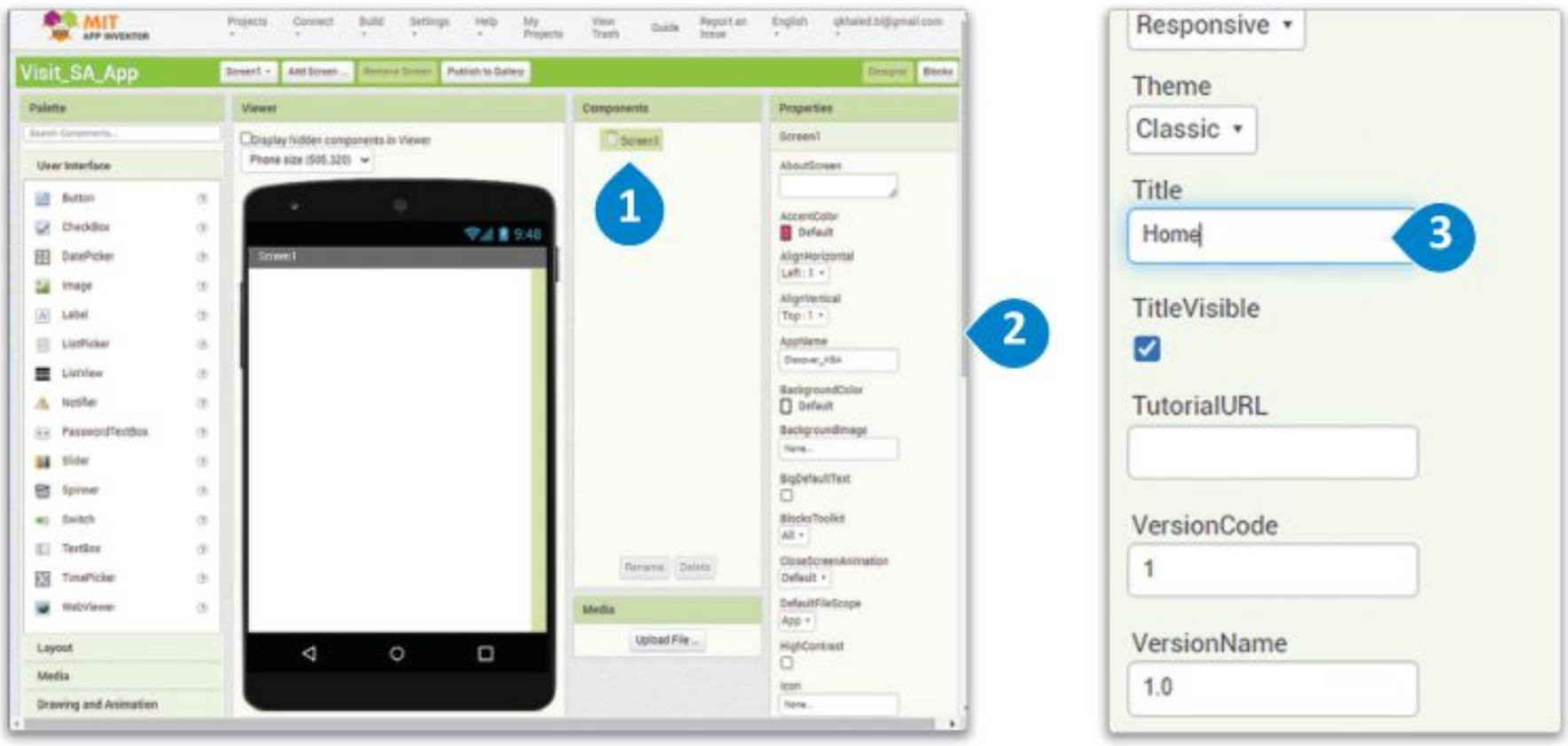
المُكوِّن	الأيقونة	الوظيفة
Button (زرّ)	Button	زرّ أمر لأداء مهمة محددة عند الضغط عليه.
Image (صورة)	Image	مُكوِّن خاص يعرض الصور.
Label (التسمية)	Label	يعرض النص المطلوب تخصيصه في حقل Text (النص) في لوحة Properties (الخصائص).
ListPicker (قائمة الخيارات)	ListPicker	عند الضغط عليه، يعرض هذا المُكوِّن العديد من الخيارات النصية للاختيار من بينها.

تغيير خصائص المُكوِّن Changing the Properties of a Component

ستجعل عنوان الشاشة الأولى Home (الرئيسة) عِوضًا عن Screen1 (شاشة 1)، وستكون تلك بمثابة الشاشة الرئيسة للتطبيق، وستغيّر Title (العنوان) إلى Home (الرئيسة) في Viewer (العارض) كما هو موضح في الشكل أدناه.

لتغيير عنوان الشاشة:

> حدِّد Screen1 (شاشة 1)، من قسم Components (المُكوِّنات). 1
> مرِّر الشريط الجانبي في قسم Properties (الخصائص) لأسفل، 2 وفي حقل Title (العنوان)، اكتب كلمة Home (الرئيسة). 3



شكل 3.5: تغيير عنوان الشاشة

إضافة زرّ إلى الشاشة Adding a Button to the Screen

ستنشئ الآن زرّ أمر يسمى Visit KSA (زيارة المملكة العربية السعودية)، وعند الضغط على هذا الزرّ سيفتح التطبيق شاشة جديدة ويعرض الوجْهَات التي يُمكن زيارتها في المملكة العربية السعودية.

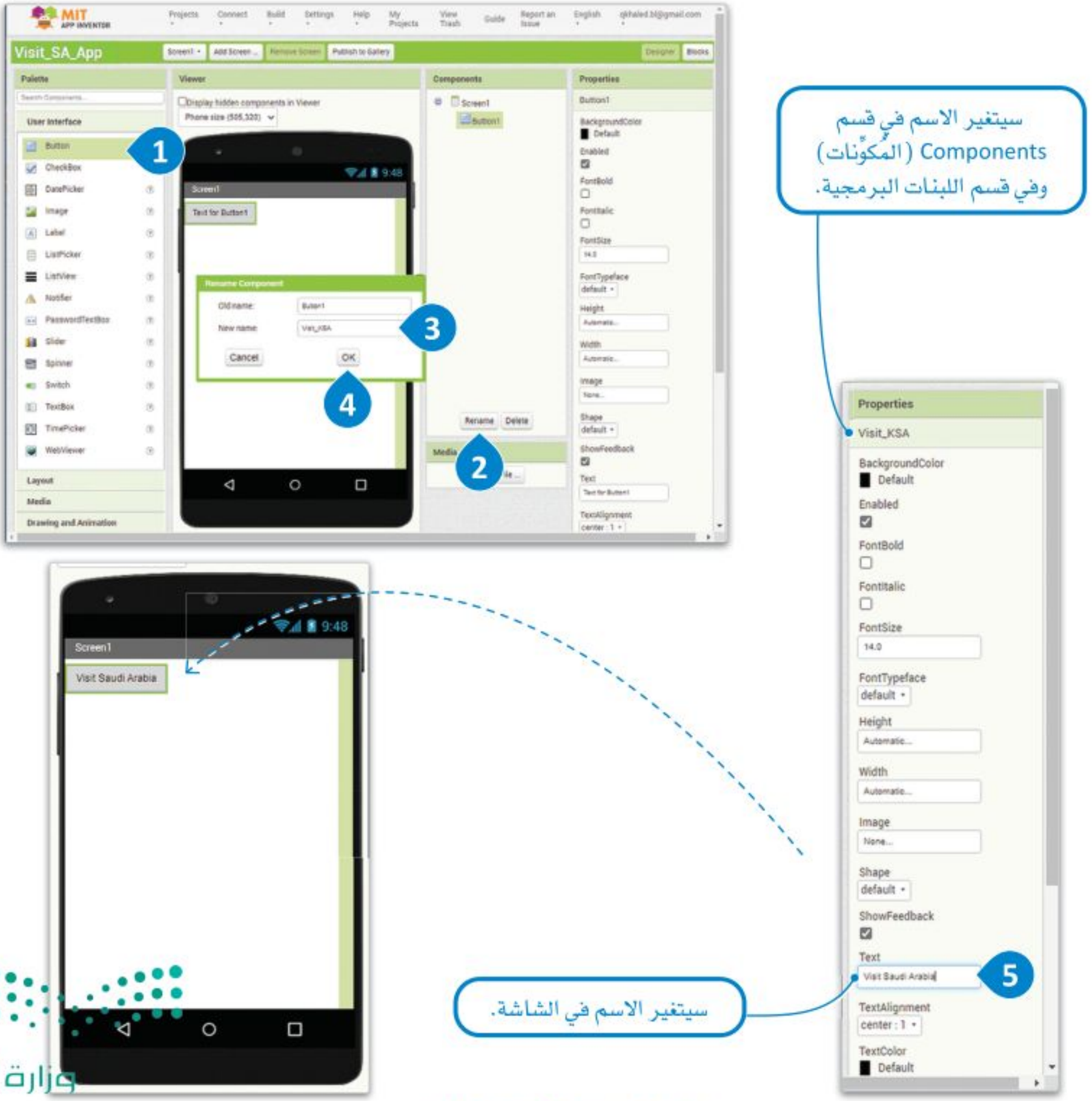
لإضافة المُكوِّن Button (زرّ):

> اسحب المُكون Button (زرّ) وأفلته من لوحة User Interface (واجهة المُستخدِم) إلى الشاشة. 1

> اضغط على Rename (إعادة تسمية). 2

> اكتب Visit KSA (زيارة المملكة العربية السعودية)، 3 ثم اضغط على OK (موافق). 4

> مرِّر الشريط الجانبي لأسفل في قسم Proprties (الخصائص)، ثم اضغط على Text (نص) واكتب Visit Saudi Arabia (زيارة المملكة العربية السعودية). 5



شكل 3.6: إضافة المُكوِّن button (زرّ).

محاذاة مُكوِّنات الشاشة Aligning the Components of the Screen

يجب أن تضع أزرار التطبيق في منتصف الشاشة الرئيسة.

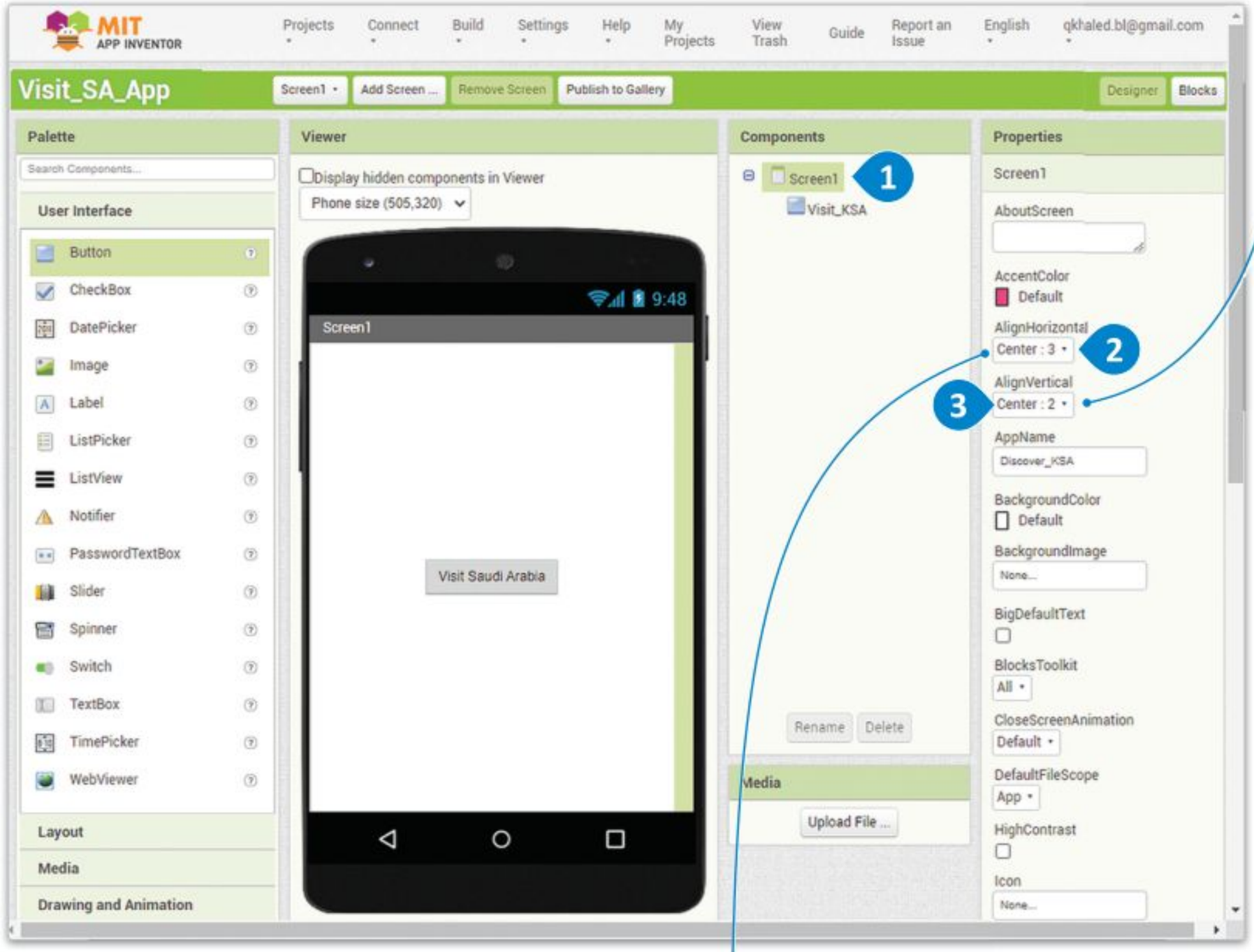
لوضع الزرّ في منتصف الشاشة:

> حدِّد Screen1 (شاشة 1) من قسم components (المُكوِّنات). 1

> من قسم Properties (الخصائص)، ومن AlignHorizontal (محاذاة أفقية) اختر Center: 3 (المنتصف:3). 2

> من AlignVertical (محاذاة عمودية) اختر Center: 2 (المنتصف:2). 3

الرقم 2 هو الرقم المخصص لهذا الإعداد لأداة المحاذاة الرأسية.



الرقم 3 هو الرقم المخصص لهذا الإعداد لأداة المحاذاة الأفقية.

شكل 3.7: توسيط الزرّ

إضافة صورة خلفية إلى الشاشة Adding a Background Image to the Screen

ستجري بعض التحسينات على التطبيق من خلال إضافة صورة خلفية إلى الشاشة.

لإضافة صورة الخلفية:

> من قسم الخصائص في Screen1 (شاشة 1)، حدِّد خاصية BackgroundImage (صورة الخلفية). 1

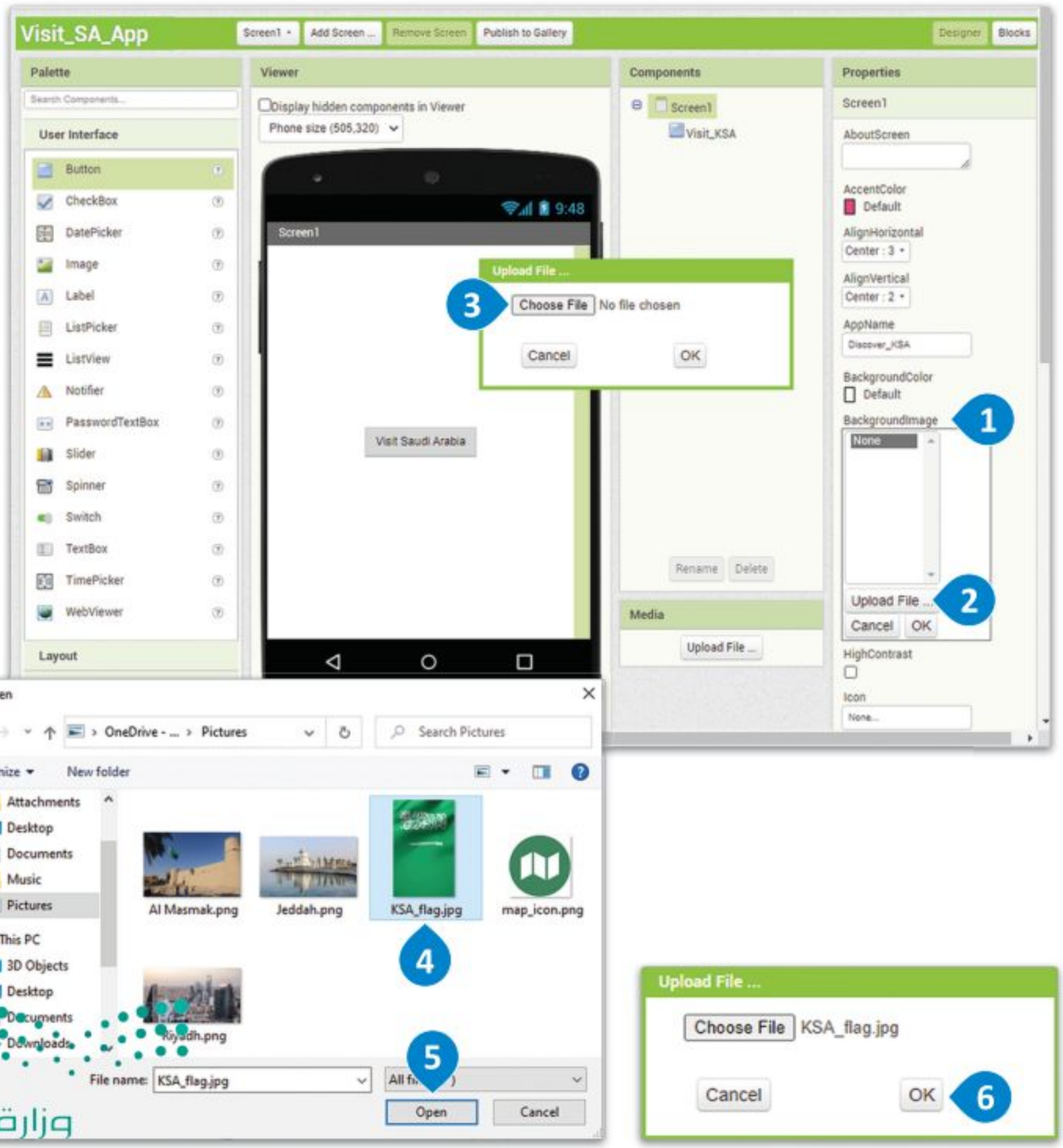
> اضغط على Upload File (تحميل ملف) لتحميل الصورة من حاسبك. 2

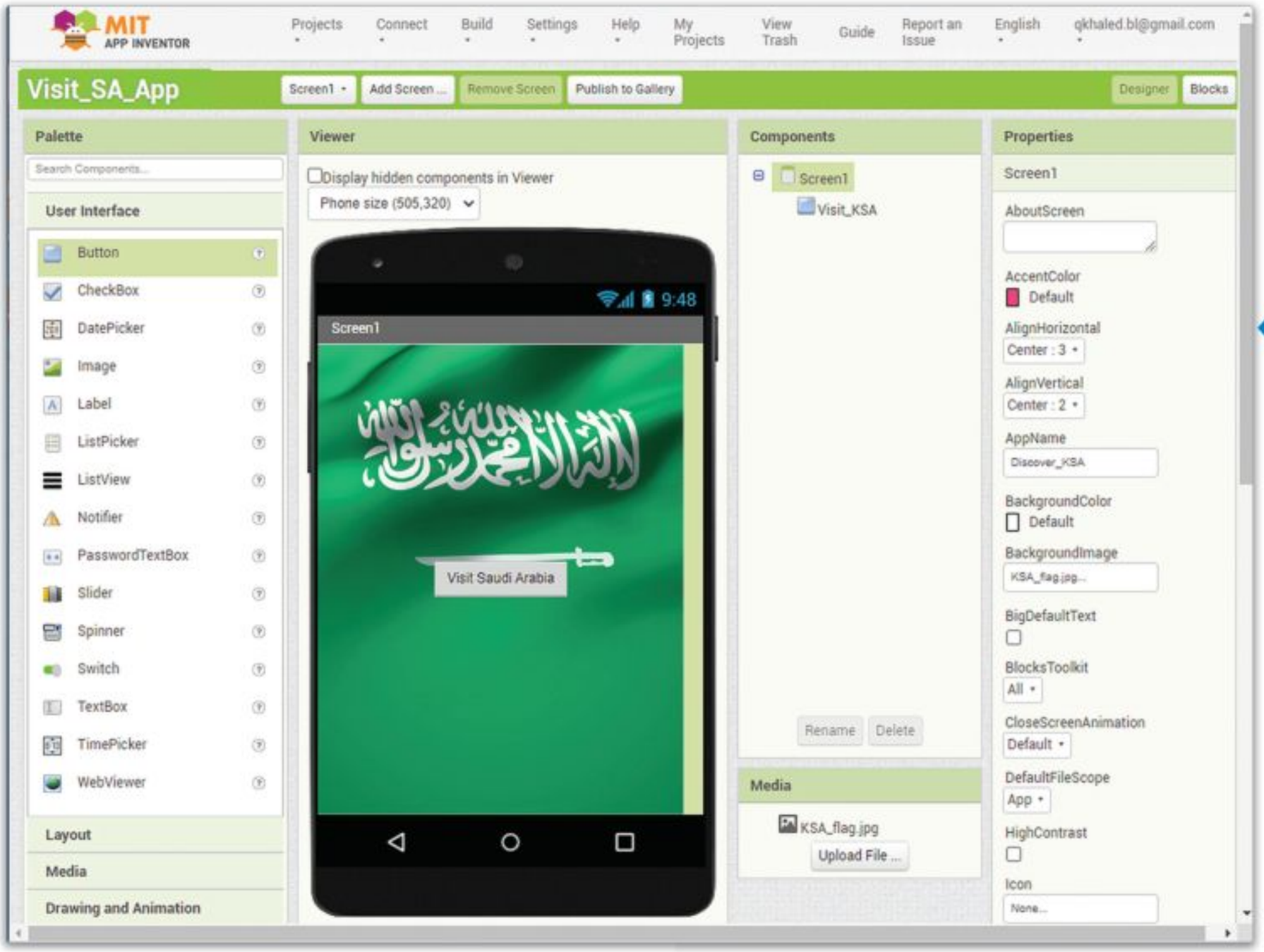
> اضغط على Choose File (اختيار ملف) لاختيار صورة من حاسبك. 3

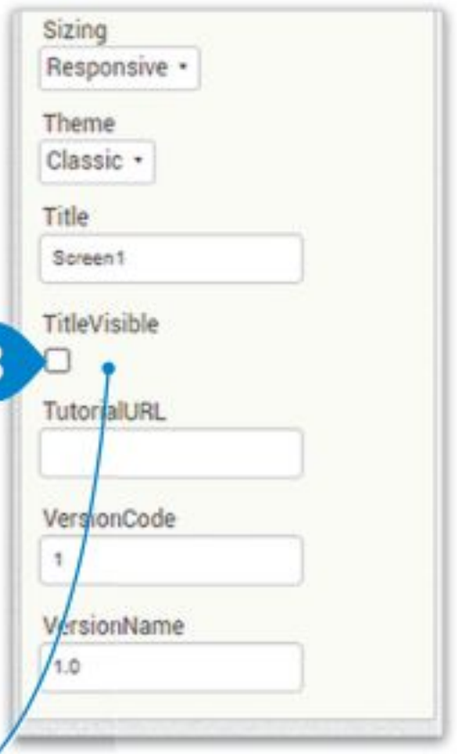
> ستظهر نافذة مفتوحة، اختر منها الصورة التي تريد إضافتها من حاسبك، 4 ثم اضغط على Open (فتح). 5

> ثم اضغط على OK (موافق). 6

> مرِّر الشريط الجانبي لأسفل في قسم Properties (الخصائص) الخاص بشاشة Screen1 (شاشة 1)، 7 وألغِ تحديد خاصية TitleVisible (العنوان مرئي). 8







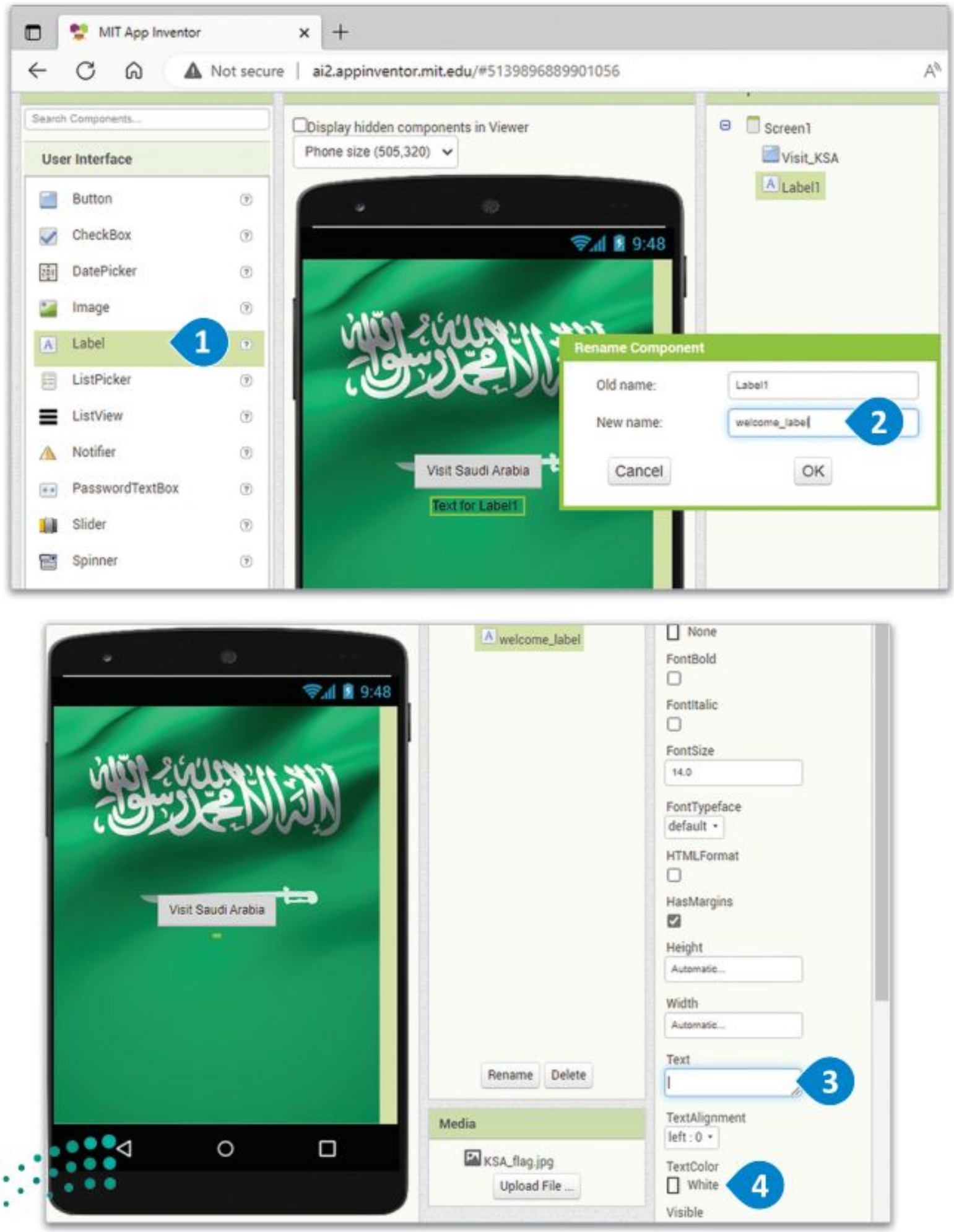
عند إلغاء تحديد هذه الخاصية
لن يتم عرض اسم الشاشة.

شكل 3.8: إضافة صورة الخلفية

لإضافة مُكوِّن Label (التسمية):

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Label (التسمية) إلى الشاشة عن طريق سحبه وإفلاته أسفل Button1 (زرّ 1)، (1) وأعد تسميته إلى welcome_label (التسمية_الترحيبية) بتحديد Rename (إعادة تسمية) من لوحة Components (المُكوِّنات). (2)

> في المُكوِّن welcome_label (التسمية_ الترحيبية)، امسح خاصية Text (النص)، (3) واضبط الخاصية TextColor (لون النص) إلى White (أبيض). (4)



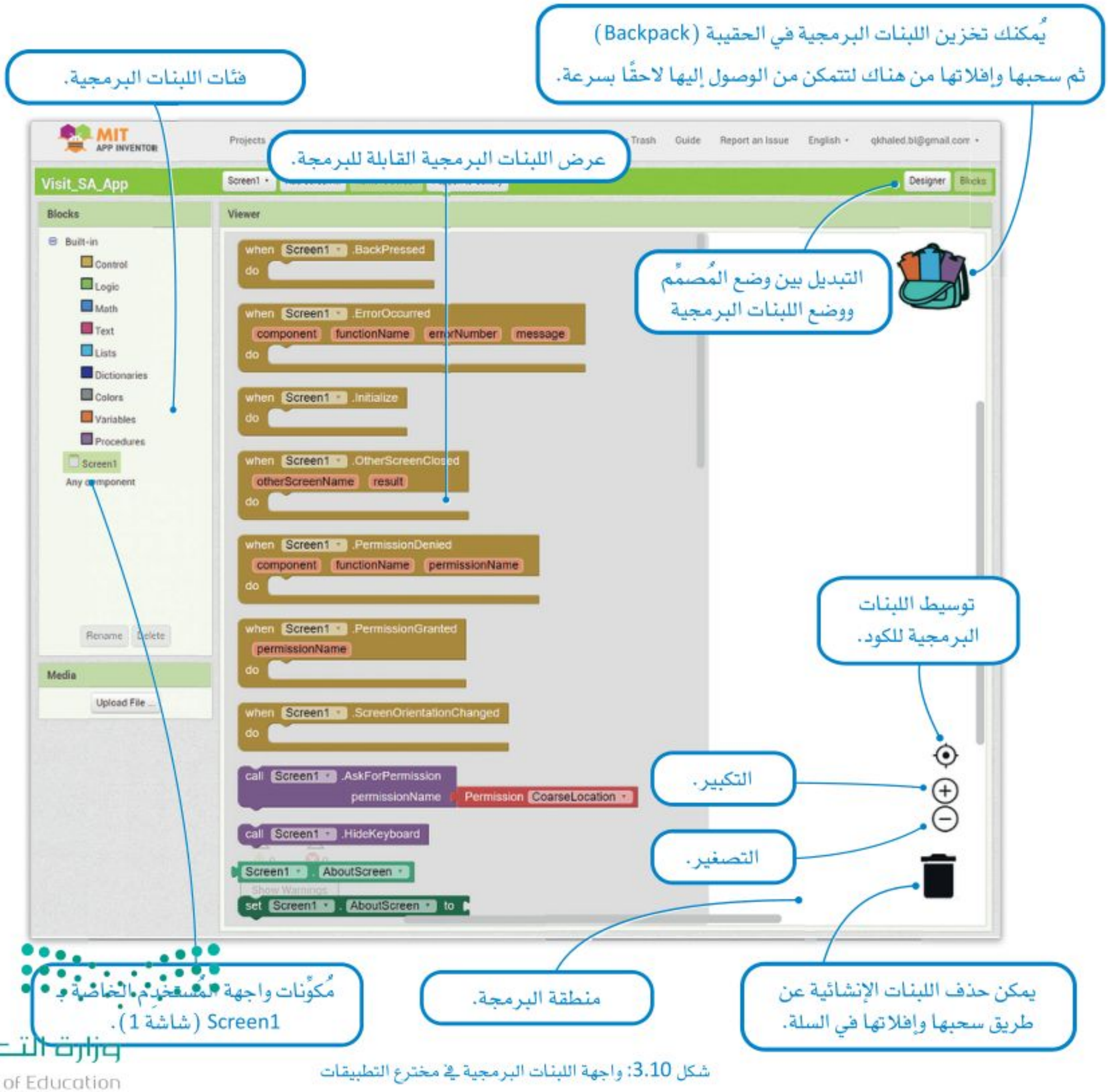
شكل 3.9: إضافة مُكوِّن Label (التسمية)

برمجة الزرّ التفاعلي Programming the Interactive Button

بعد أن أضفت الزرّ، ستضيف الآن بعض اللبنات البرمجية التي ستجعل الزرّ يعرض الجملة Welcome to Saudi Arabia (مرحبًا بكم في المملكة العربية السعودية) عند الضغط عليه. عليك أولًا تغيير العرض من Designer (المُصمِّم) إلى Blocks (اللبنات البرمجية).

صفحة اللبنات البرمجية في مخترع التطبيقات The App Inventor Blocks Page

هذه هي صفحة اللبنات لواجهة مخترع التطبيقات حيث سيتم عرض جميع المُكوِّنات التي أضفتها من صفحة المُصمِّم هنا، ويمكن برمجتها باستخدام لغة برمجة مرئية قائمة على اللبنات البرمجية. هناك أنواع عديدة من اللبنات التي يمكن استخدامها، بما فيها لبنات برمجية **لمنطق البرنامج** (Program Logic)، **ومعالجة الحدث** (Event Handling)، **ومعالجة المتغير** (Variable Manipulation) **وتغيير المُكوِّن** (Component Alteration).



شكل 3.10: واجهة اللبنات البرمجية في مخترع التطبيقات

مجموعات أوامر اللبنات البرمجية

Control	التحكم في سير البرنامج.
Logic	تنفيذ العمليات المنطقية.
Math	إجراء العمليات الحسابية.
Text	إجراء العمليات على النصوص.
Lists	تكوين هياكل بيانات القوائم.
Dictionaries	تكوين هياكل بيانات القاموس.
Colors	إضافة الألوان إلى المُكوِّنات.
Variables	تهيئة المُتغيرات ومعالجتها.
Procedures	تنفيذ إجراءات مخصصة.

لكل مُكوِّن تحدده معالجات أحداث (Event Handlers) وعمليات خاصة به للتحكم في خصائصه.

لتحديد حدث الضغط على الزرّ:

> حدِّد المُكوِّن Visit_KSA (زيارة _المملكة العربية السعودية). 1

> اختر اللبنة البرمجية when Visit_KSA .Click (عند الضغط على زيارة _المملكة العربية السعودية). 2

> اسحبها وأفلتها في منطقة البرمجة. 3

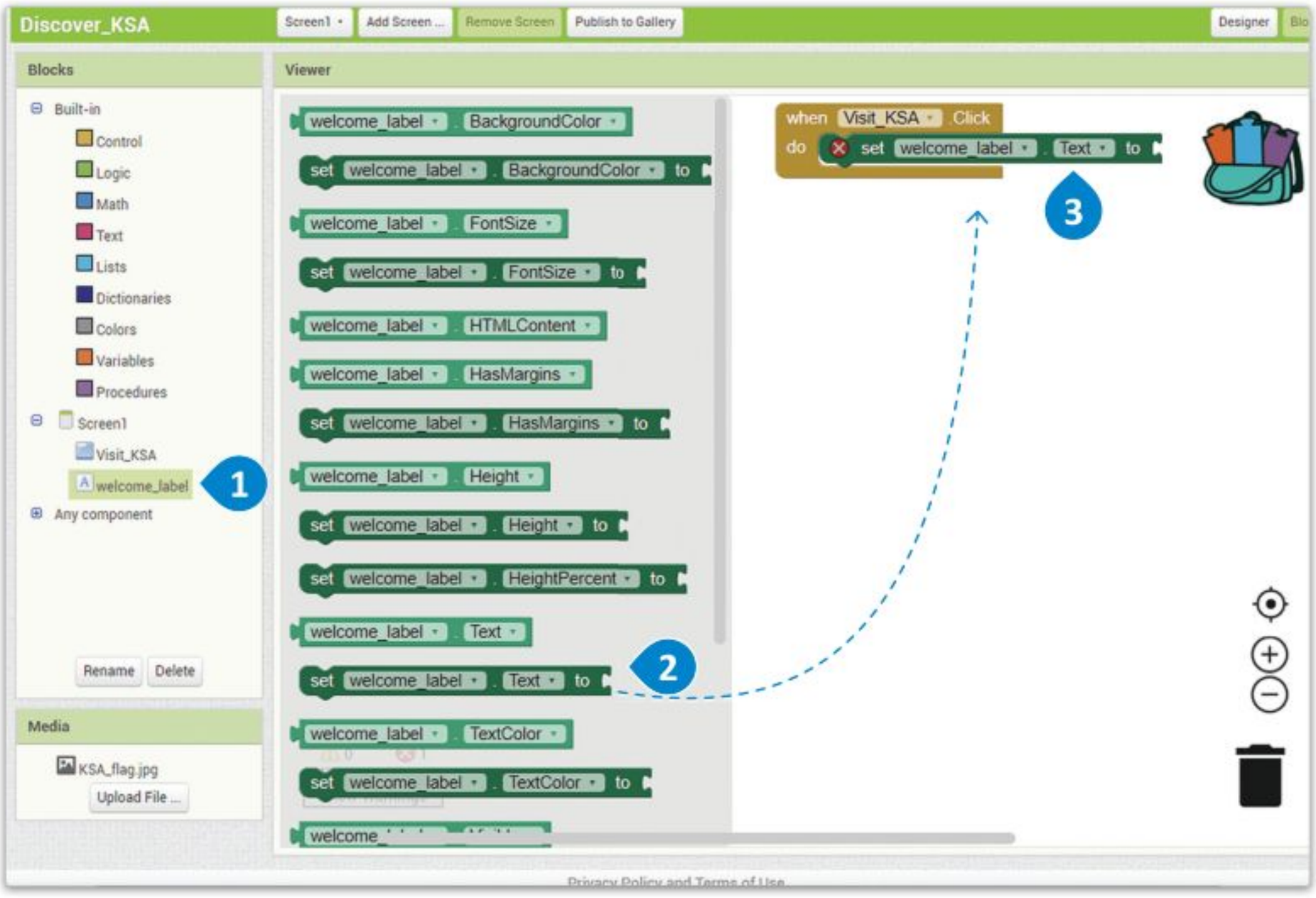
شكل 3.11: تحديد حدث الضغط على الزرّ

للوصول إلى خاصية Text (النص) الخاصة بالتسمية:

> حدِّد المُكوِّن welcome_label (التسمية _الترحيبية). 1

> اسحب وأفلت اللبنة البرمجية set welcome_label .Text to (عيِّن نص التسمية _الترحيبية). 2

> ضع اللبنة السابقة داخل قِسم do (نفِّذ) الخاص باللبنة البرمجية when Visit_KSA.click (عند الضغط على زيارة _المملكة العربية السعودية). 3



شكل 3.12: الوصول إلى خاصية Text (النص) الخاصة بالتسمية

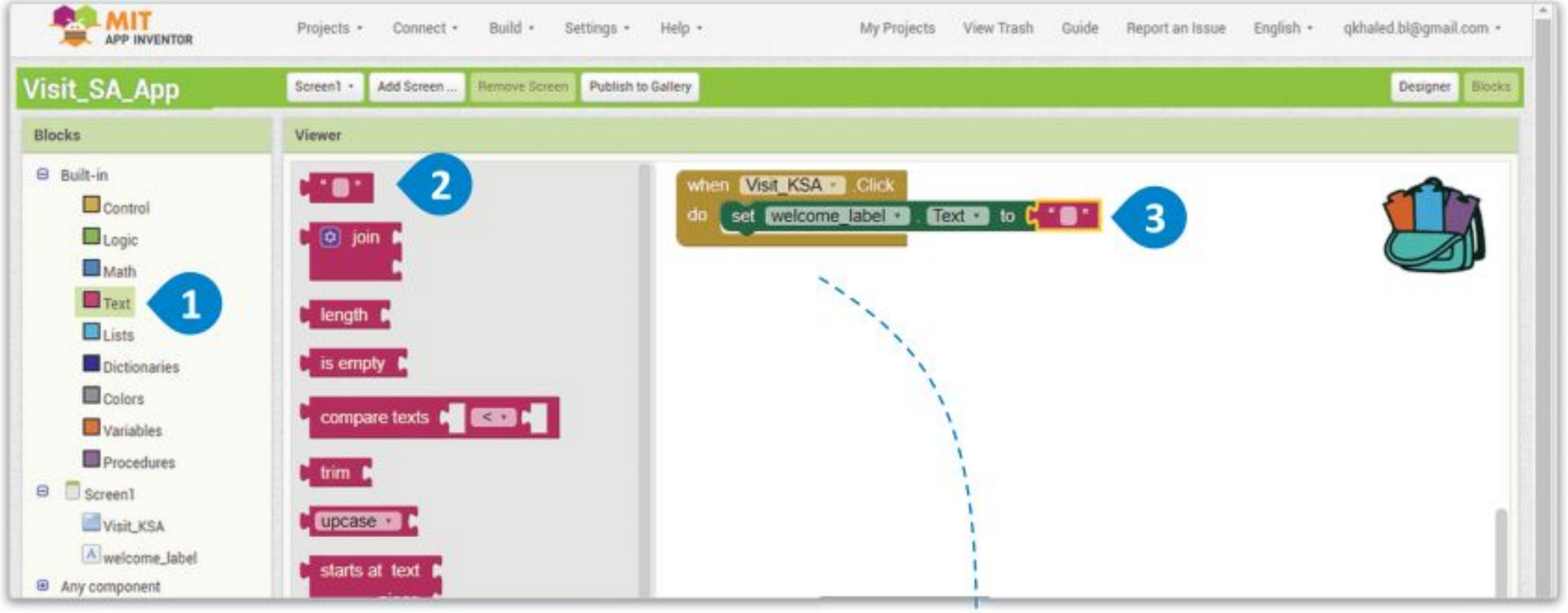
لتعديل خاصية Text (النص) الخاصة بالتسمية :

> حدِّد مجموعة Text (النص). 1

> حدِّد اللبنة البرمجية empty string (سلسلة نصية فارغة). 2

> ضع اللبنة البرمجية empty string (سلسلة نصية فارغة) في اللبنة البرمجية set Text (عيِّن نصًّا). 3

> اكتب Welcome to Saudi Arabia (مرحبًا بكم في المملكة العربية السعودية) في لبنة empty string (سلسلة نصية فارغة). 4



شكل 3.13: تعديل خاصية Text (النص) الخاصة بالتسمية

اختبار التطبيق Testing the Application

ستحتاج في مراحل مختلفة من عملية تطوير التطبيق إلى اختباره للتأكد من جاهزية جميع وظائفه، ويساعد الاختبار المتكرر أثناء التطوير في اكتشاف الأخطاء المحتملة في البرمجة، ويتيح تصحيحها قبل نشر التطبيق وعرضه.

يمكنك اختبار التطبيق في مخترع التطبيقات MIT بطريقتين مختلفتين، تعتمد الأولى منهما على الاختبار من خلال **المحاكي** (Emulator) وهو برنامج تقوم بتثبيته على حاسبك ليحاكي جهاز الهاتف المحمول.

أما الطريقة الثانية فهي باستخدام MIT AI2 Companion، وهو تطبيق تقوم بتثبيته على هاتفك المحمول فعليًا، وبعد تثبيت التطبيق يمكنك الدخول إلى موقع مخترع التطبيقات الإلكتروني ومسح رمز **الاستجابة السريعة** (QR) الخاص بتطبيقك باستخدام تطبيق MIT AI2 Companion، والذي بدوره سيُحمِّل التطبيق الذي أنشأته على المتصفح الموجود على هاتفك. يمكنك تثبيت تطبيق MIT AI2 Companion من الرابط التالي:

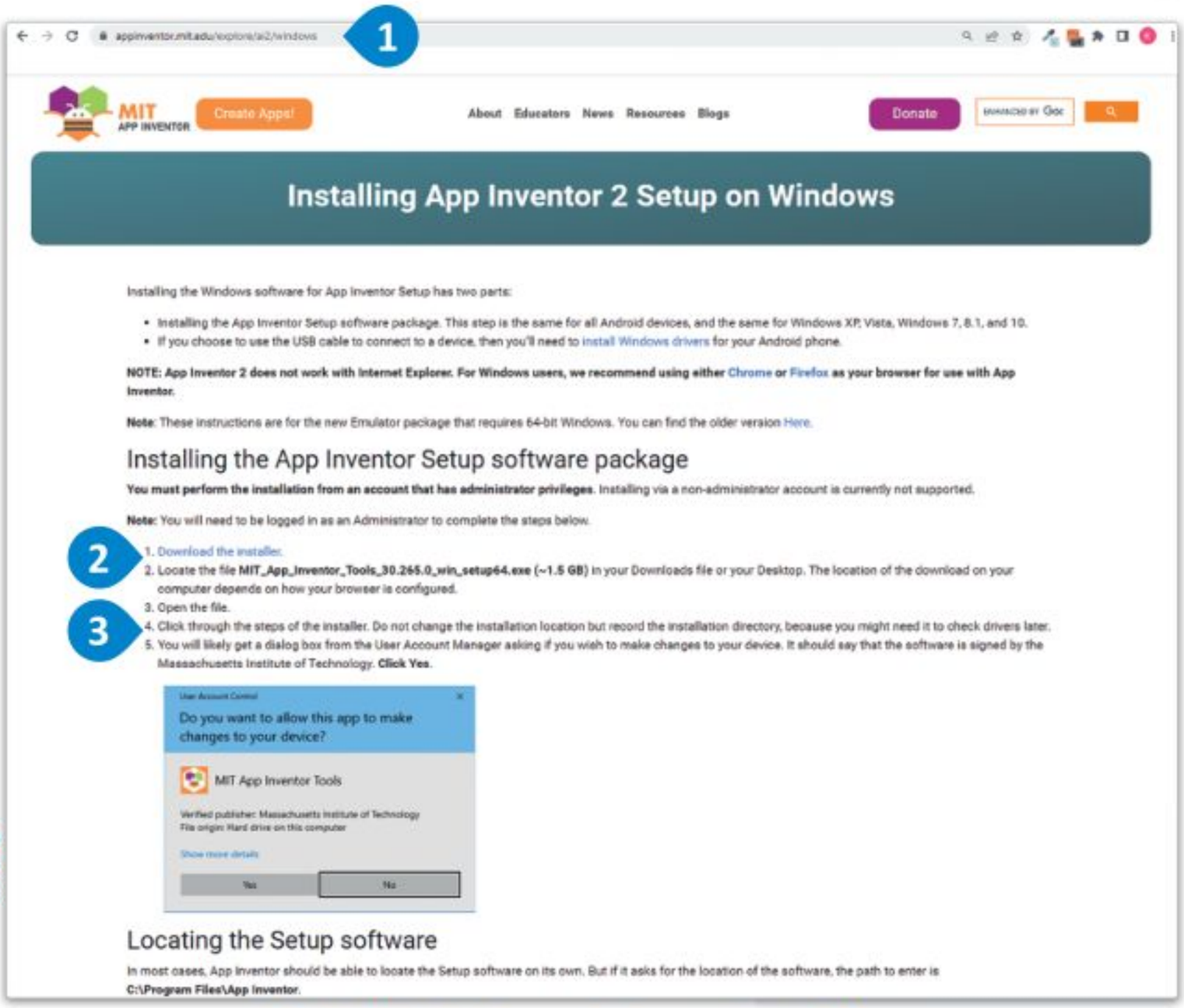
https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&pli=1

إعداد محاكي الأندرويد Setting up the Android Emulator

ستقوم الآن بتثبيت **محاكي الأندرويد** (Android Emulator) لتشغيل تطبيق الهاتف المحمول على حاسبك.

لإعداد تطبيق محاكي الأندرويد (Android Emulator):

> انتقل إلى موقع الويب التالي: https://appinventor.mit.edu/explore/ai2/windows. (1)
> اضغط على رابط Download the installer (تنزيل المُثبت) لتنزيل ملف المُثبت بامتداد ".exe". (2)
> بعد تنزيل المُثبت اتبع الخطوات الموضحة في صفحة الويب. (3)



شكل 3.14: تثبيت تطبيق محاكي الأندرويد (Android Emulator)

1. تشغيل التطبيق باستخدام محاكي الأندرويد

لتشغيل التطبيق:

> اضغط على Connect (اتصال). 1
> اختر Emulator (المحاكي). 2
> اضغط على الزرّ لعرض الرسالة. 3

يجب تشغيل تطبيق المحاكي للحاسب المكتبي قبل بدء الاتصال بموقع مخترع التطبيقات على الويب.

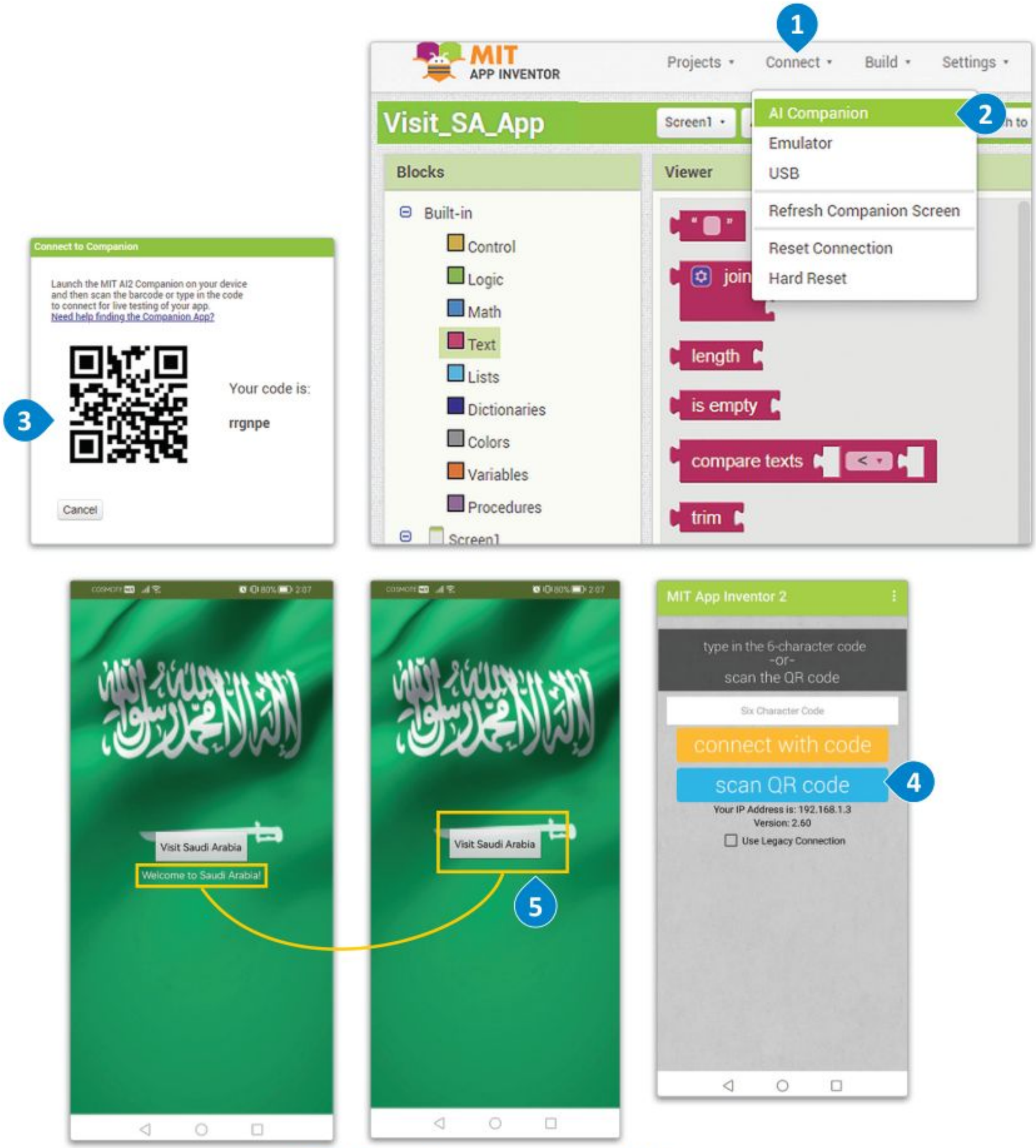
شكل 3.15: اختبار التطبيق باستخدام محاكي مخترع التطبيقات (App Inventor Emulator)

2. تشغيل التطبيق باستخدام AI Companion

لاتصال التطبيق بـ AI Companion:

> اضغط على Connect (اتصال)، 1 ثم اضغط على AI Companion من القائمة العلوية. 2
> سيظهر صندوق حوار برمز QR (الاستجابة السريعة) على شاشة حاسبك. 3
> شغّل تطبيق MIT AI2 Companion من هاتفك المحمول، ثم اضغط على زرّ Scan QR code (مسح رمز الاستجابة السريعة) الموجود في Companion. 4 وقم بمسح الرمز في نافذة App Inventor (مخترع التطبيقات) وسيتم عرض التطبيق الذي تقوم بإنشائه على جهازك.
> اضغط على الزرّ لعرض الرسالة. 5

قبل مسح رمز الاستجابة السريعة، يجب أن يكون الجهازان متصلين بالشبكة اللاسلكية نفسها.



شكل 3.16: اختبار التطبيق باستخدام MIT AI2 Companion

عند إغلاق AI Companion ستتم إزالة التطبيق، ويجب تثبيت التطبيق على الجهاز لكي يبقى بشكلٍ دائم.

تمرينات

1 صِفْ المراحل الأربع لتطوير تطبيقات الهاتف المحمول.

2 قارن بين عملية تطوير تطبيقات الأجهزة المحمولة باستخدام مخترع التطبيقات MIT وتطويرها بالطرق التقليدية.

3 ضعْ قائمة بمزايا تطوير تطبيقات الأجهزة المحمولة باستخدام مخترع التطبيقات MIT.

4 أنشئ تطبيقًا بسيطًا حول الدولة التي تودُّ زيارتها.

- أضف شاشة جديدة باسم Home (الرئيسة)، وأدرج صورة خلفية مع عَلَم تلك الدولة.
- أضف زرّين باسم Sightseeing (مشاهدة المعالم) وUseful Information (معلومات مفيدة).
- أنشئ شاشة جديدة واستخدم أداة Label (التسمية) لكتابة بعض المعلومات المفيدة التي ستظهر عند الضغط على الزرّ.

5 ما الدور الذي يلعبه النموذج الأولي الشبكي في عملية تطوير تطبيق السياحة؟

وزارة التعليم
Ministry of Education
2023 - 1445

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

إضافة المزيد من العناصر إلى التطبيق

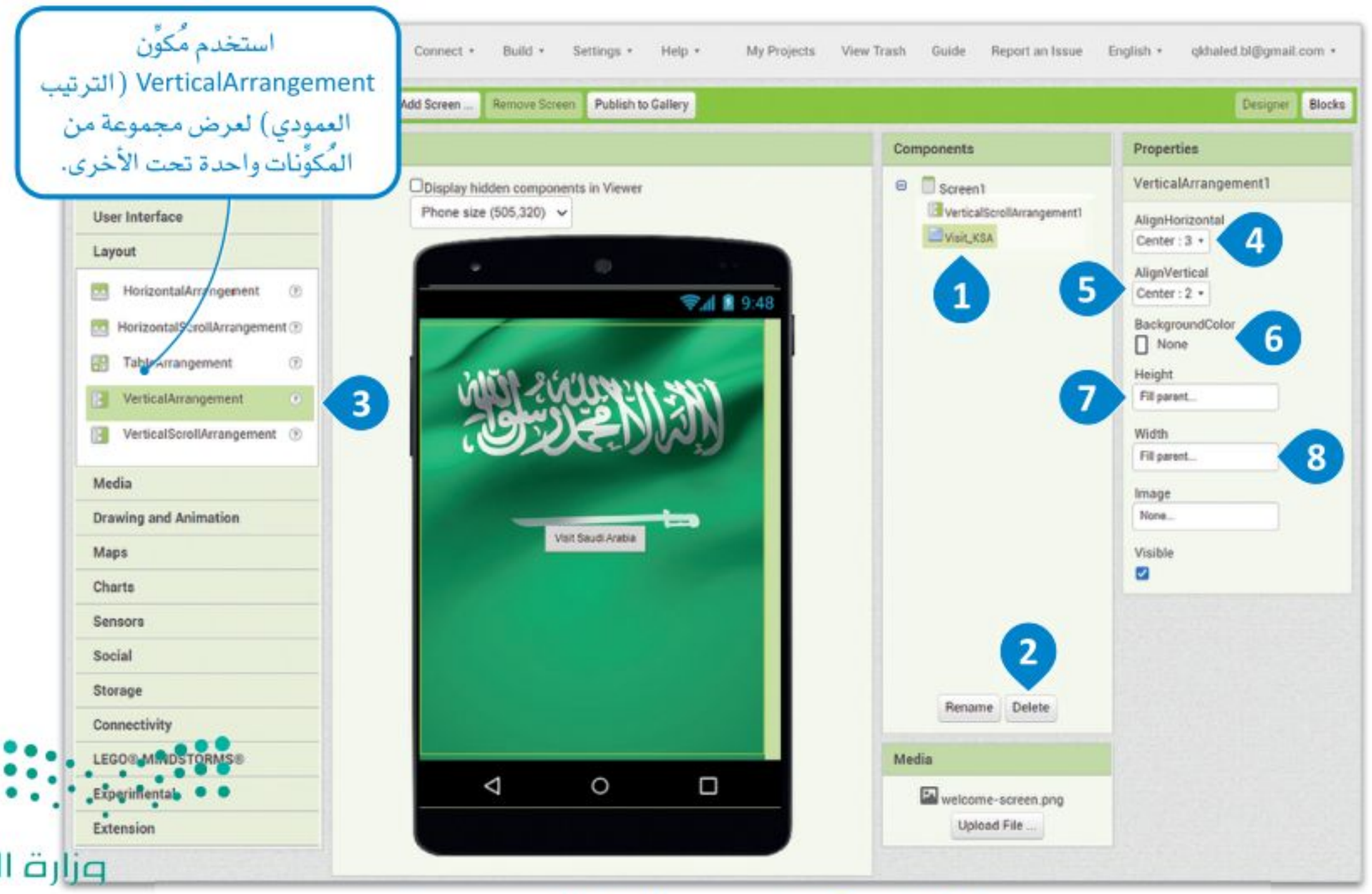
تصميم الصفحة الرئيسة Designing the Home Screen

لقد أنشأت الشاشة الرئيسة لتطبيقك في الدرس السابق، وقدتضمنت زرًّا يعرض رسالة عند الضغط عليه، وفي التدريب الأخير من الدرس السابق حذفت الزرّ وكذلك الرسالة التي كان يعرضها.

ستضيف الآن بعض الأزرار المفيدة إلى الشاشة الرئيسة لتطبيقك، ثم ستُجري بعض التغييرات على مظهرها.

لإضافة مُكوِّن VerticalArrangement (الترتيب العمودي):

- من قسم Components (المكونات) حدِّد زر Visit_KSA (زيارة _المملكة العربية السعودية)، 1 ثم اضغط على Delete (حذف). 2
- من مجموعة Layout (التخطيط)، أضف المُكوِّن VerticalArrangement (الترتيب العمودي) إلى الشاشة عن طريق سحبه وإفلاته في Viewer (العارض). 3
- من مُكوِّن VerticalArrangement1 (الترتيب العمودي)، عيّن خاصية AlignHorizontal (المحاذاة الأفقية) إلى Center:3 (المنتصف : 3)، 4 وخاصية AlignVertical (المحاذاة العمودية) إلى Center:2 (المنتصف:2)، 5 وخاصية BackgroundColor (لون الخلفية) إلى None (بدون). 6
- اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، 7 وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة). 8



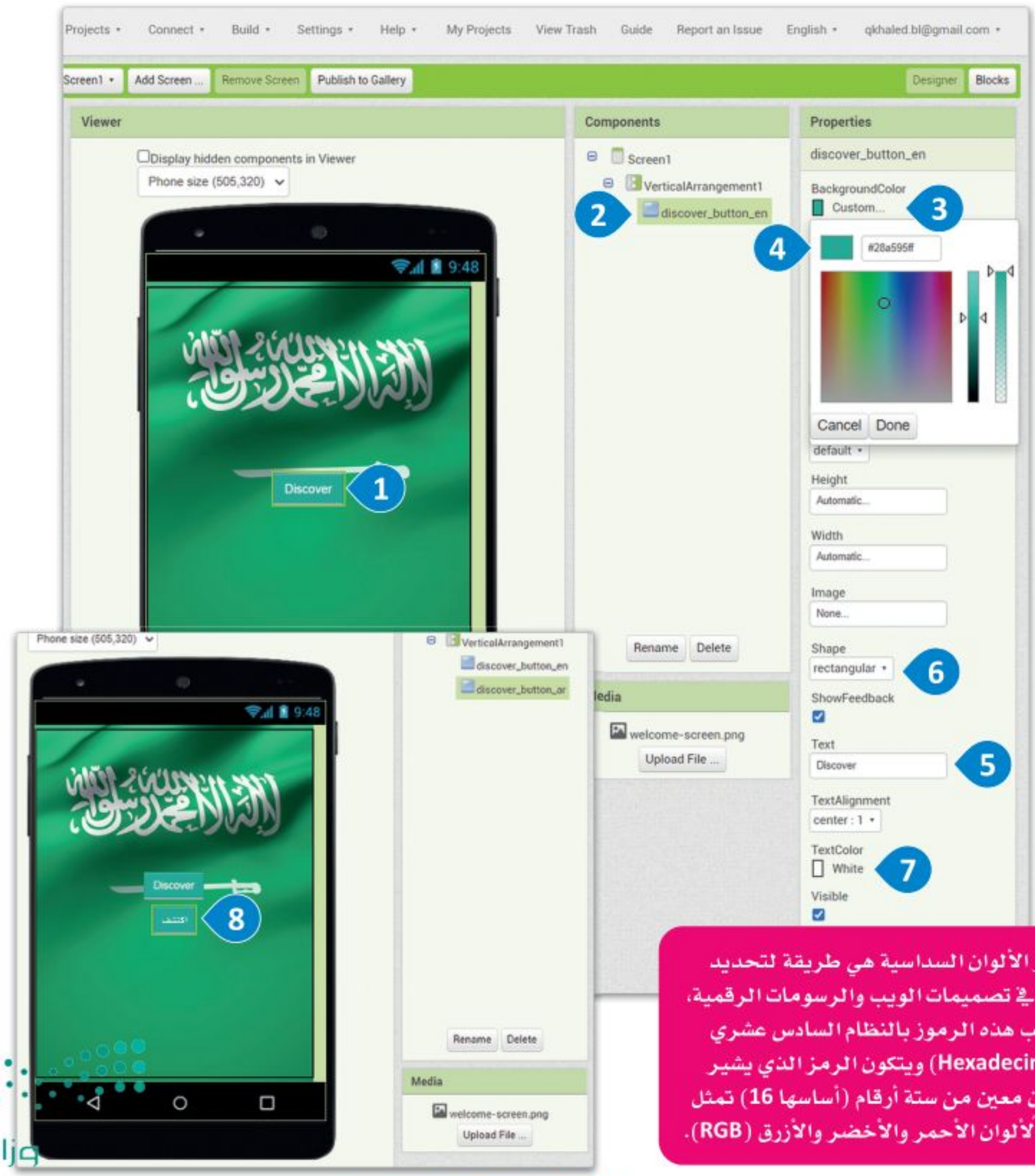
شكل 3.17: إضافة مُكوِّن VerticalArrangement (الترتيب العمودي)

لإضافة زرّ باللغة الإنجليزية:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف المُكوِّن Button (زرّ) إلى الشاشة، (1) وأعد تسميته إلى discover_button_en (زرّ_اكتشف_بالإنجليزية). (2)

> في المُكوِّن Discover_button_en (زرّ_اكتشف_بالإنجليزية)، عيّن خاصية BackgroundColor (لون الخلفية) إلى Custom (مُخصص)، (3) واكتب القيمة #28a595ff، (4) ثم عيّن خاصية Text (نص) إلى Discover (اكتشف)، (5) وعيّن خاصية Shape (الشكل) إلى rectangle (مستطيل)، (6) وأخيرًا عيّن خاصية TextColor (لون النص) إلى White (أبيض). (7)

> كرِّر الخطوات السابقة لإضافة زرّ باللغة العربية. (8)



شكل 3.18: إضافة زرّ باللغة الإنجليزية

رموز الألوان السداسية هي طريقة لتحديد الألوان في تصميمات الويب والرسومات الرقمية، وتكتب هذه الرموز بالنظام السادس عشري (Hexadecimal) ويتكون الرمز الذي يشير إلى لون معين من ستة أرقام (أساسها 16) تمثل درجات الألوان الأحمر والأخضر والأزرق (RGB).

إنشاء الشاشة الثانية للتطبيق Creating the Second Screen of the App

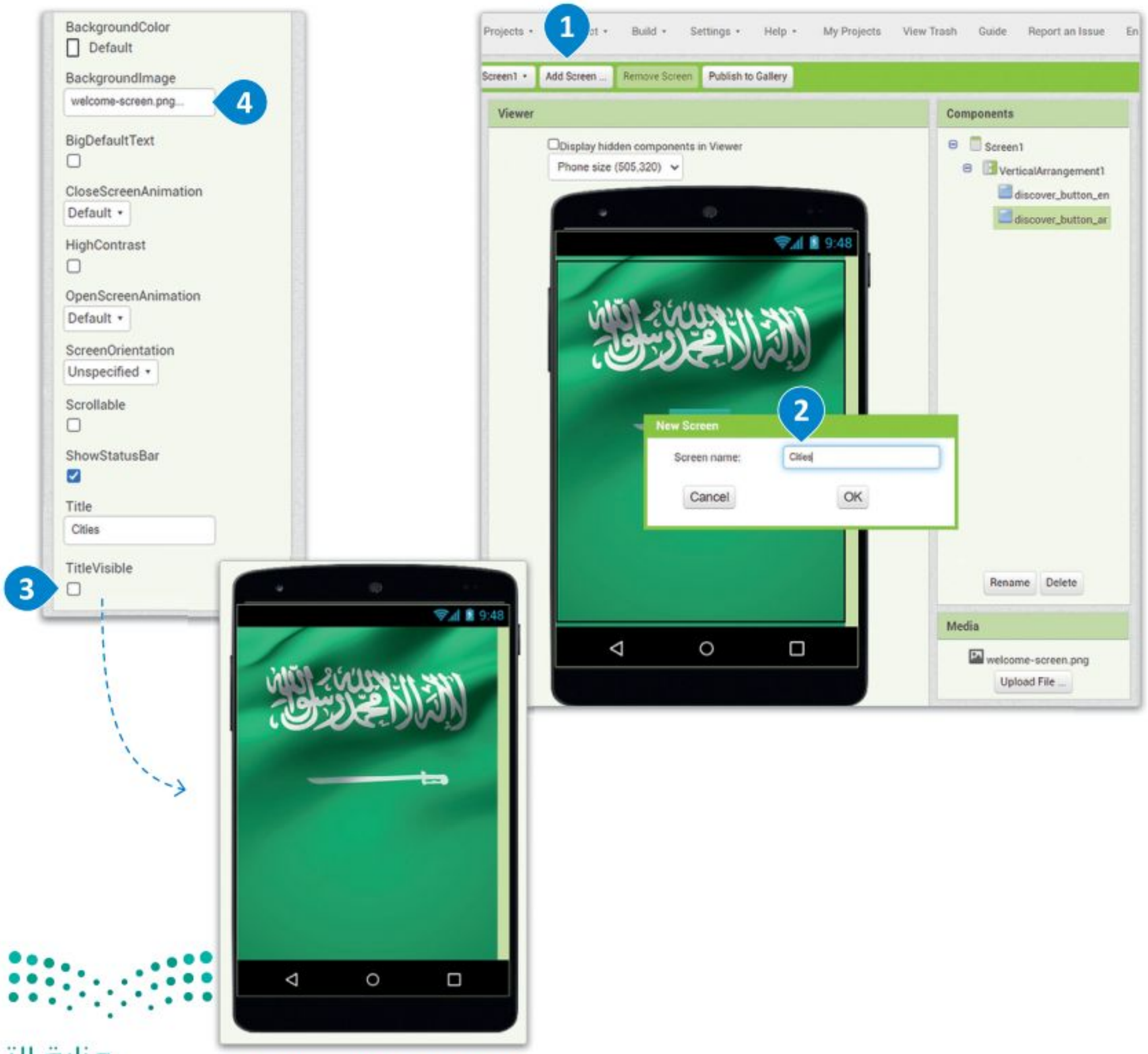
ستعرض الشاشة التالية في التطبيق للمُستخدِم مدينتي الرياض وجِدّة وأبرز معالمهما، وعندما يضغط المُستخدِم على إحدى المدن ستظهر قائمة بالمعالم المميزة المتاحة في التطبيق.

ستُضيف الآن بعض الأزرار المفيدة على الشاشة الرئيسة وتُجري بعض التغييرات على مظهرها.

لإضافة شاشة جديدة:

> اضغط على زرّ Add screen (إضافة شاشة)، 1 ثم أنشئ شاشة جديدة باسم Cities (المدن). 2

> في قسم Properties (الخصائص) لشاشة Cities (المدن)، ألغِ تحديد الخاصية TitleVisible (العنوان مرئي)، 3 وفي خاصية BackgroundImage (صورة الخلفية) ضعْ صورة لعَلَم المملكة العربية السعودية. 4

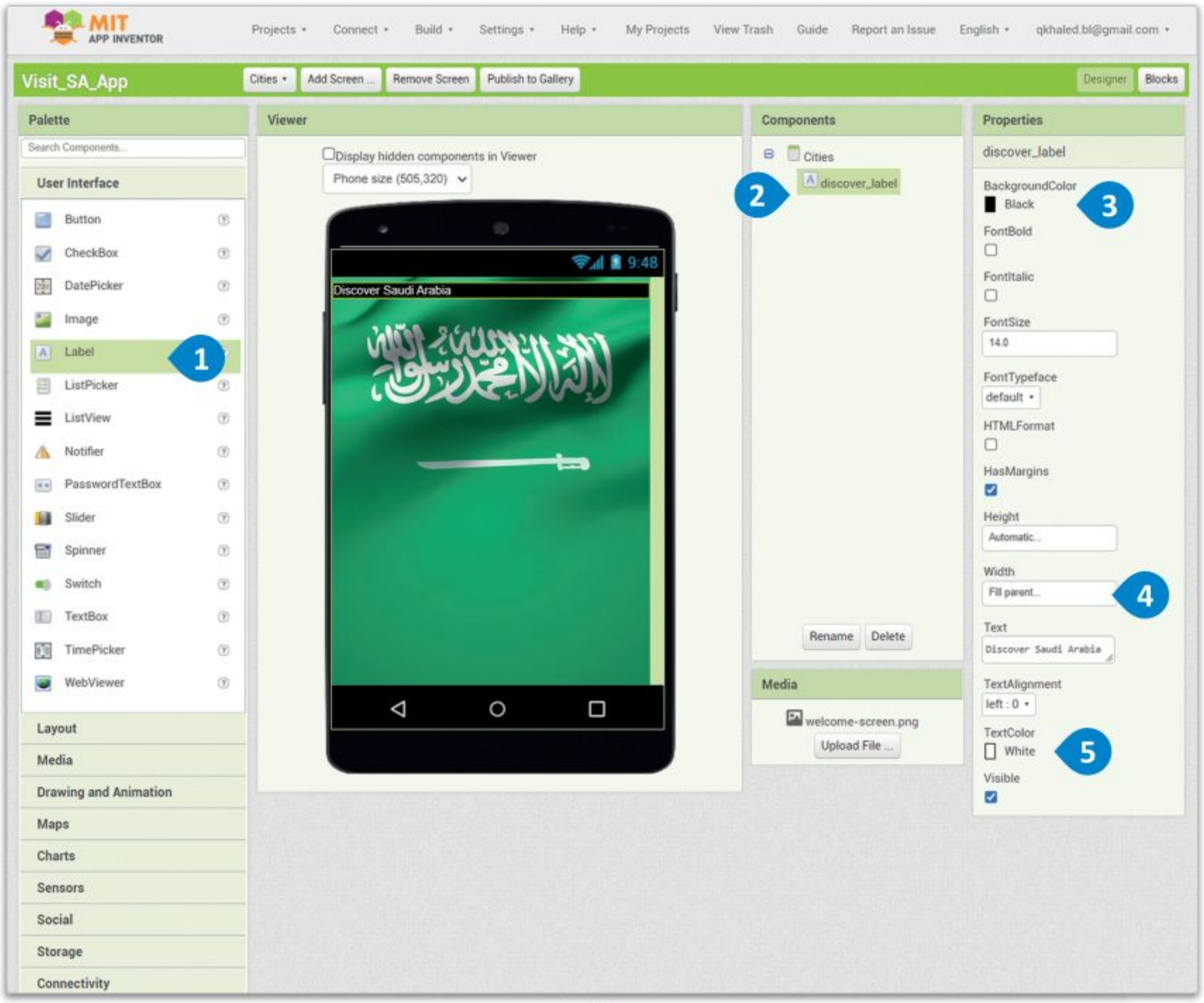


شكل 3.19: إضافة شاشة جديدة إلى التطبيق

لإضافة مُكوِّن Label (التسمية):

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Label (التسمية) إلى الشاشة. 1 ثم أعد تسميته إلى Discover_label (اكتشفِ_ التسمية). 2

> في المُكوِّن Discover_label (اكتشف_ التسمية)، اضبط خاصية BackgroundColor (لون الخلفية) إلى Black (أسود)، 3 واضبط خاصية Width (العرض) إلى Fill Parent (تعبئة المساحة)، 4 واضبط خاصية Text (نص) إلى Discover Saudi Arabia (اكتشف المملكة العربية السعودية)، وأخيرًا اضبط خاصية TextColor (لون النص) إلى White (أبيض). 5



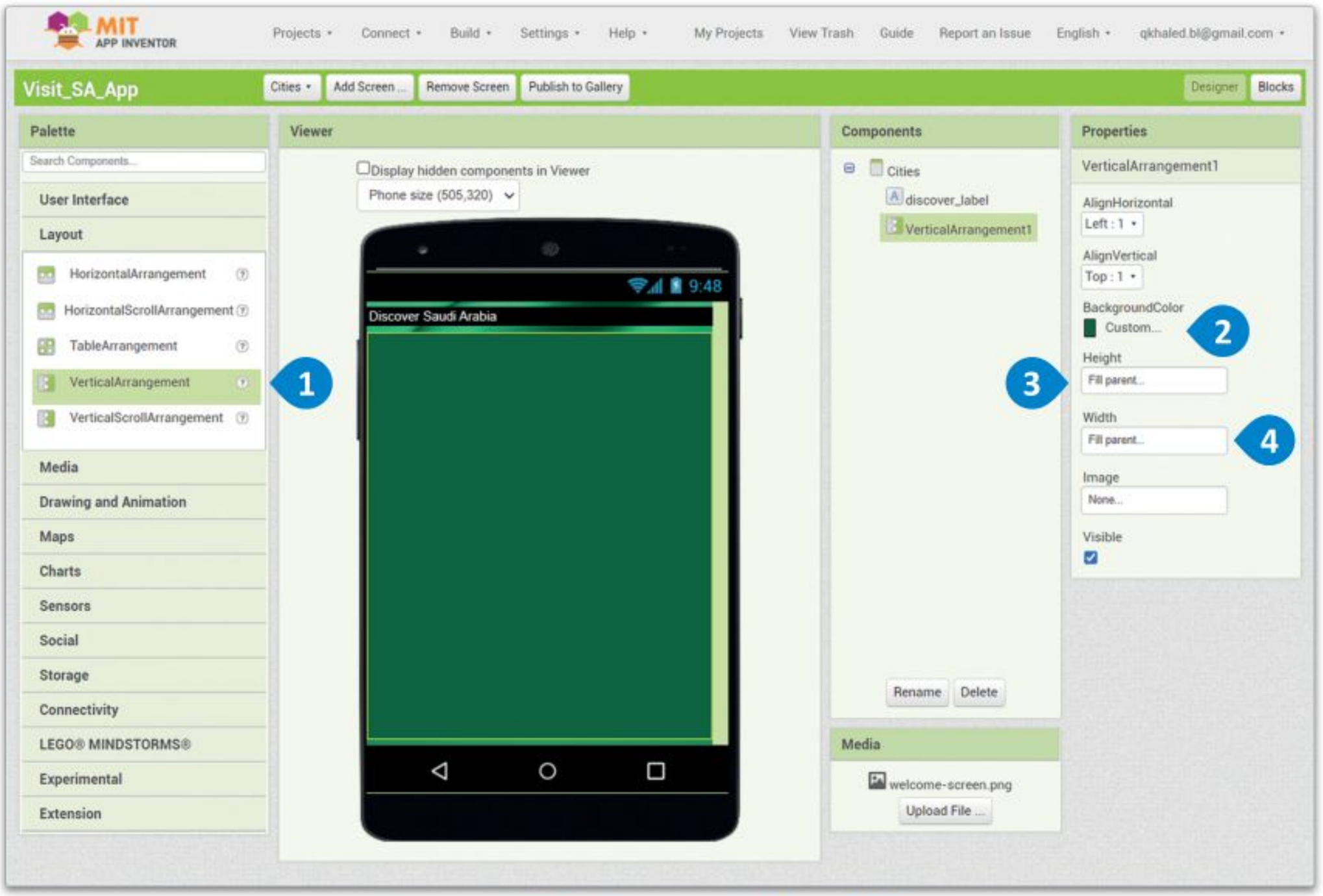
شكل 3.20: إضافة مُكوِّن Label (التسمية)

لإضافة مُكوِّن VerticalArrangement (الترتيب العمودي):

> من مجموعة Layout (تخطيط)، أضف مُكوِّن VerticalArrangement (الترتيب العمودي) إلى الشاشة. 1

> في مُكوِّن VerticalArrangement (الترتيب العمودي)، اضبط خاصية BackgroundColor (لون الخلفية) إلى Custom (مخصص)، واكتب القيمة #11613eff. 2

> في مُكوِّن VerticalArrangement1 (الترتيب العمودي)، اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، 3 وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة). 4

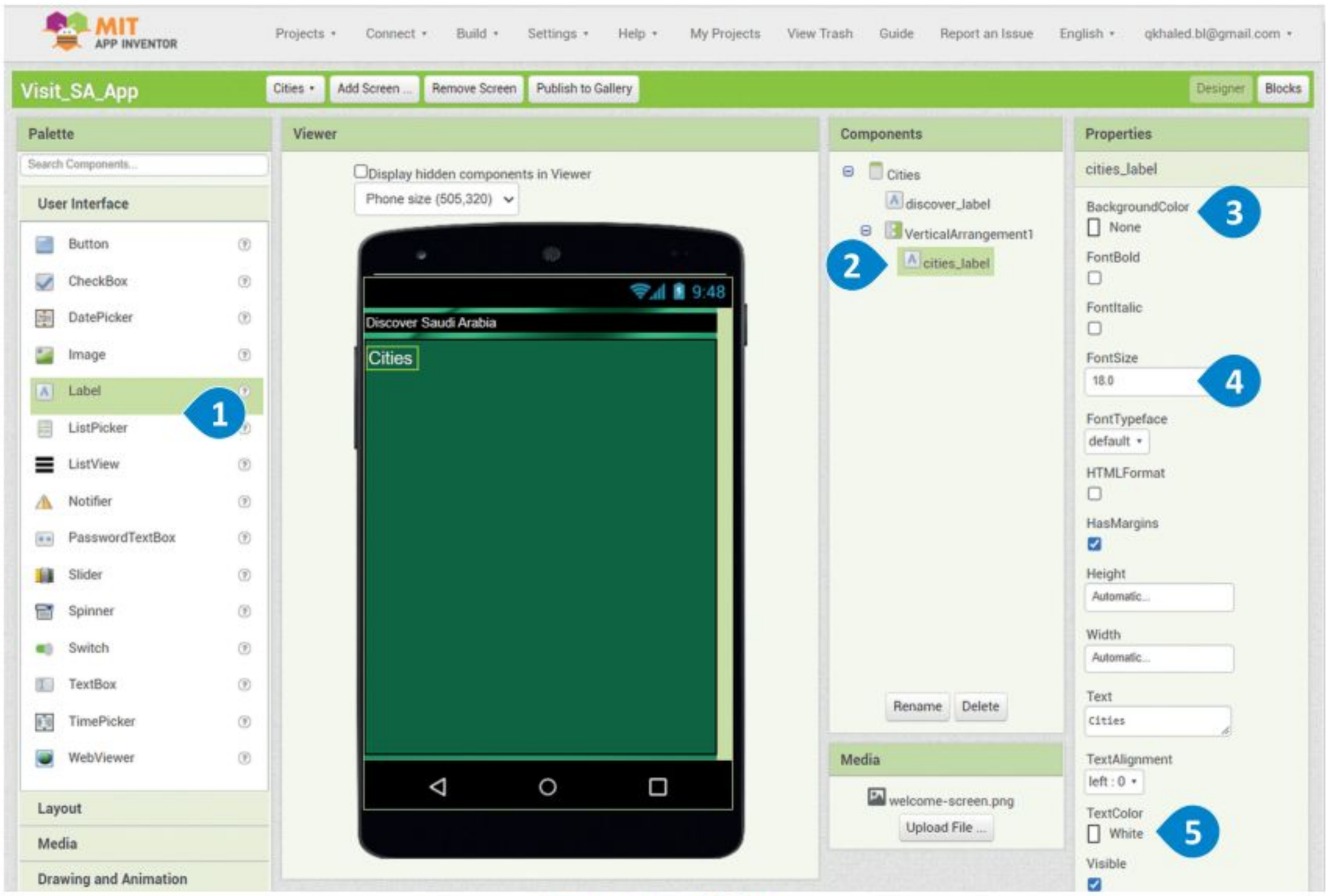


شكل 3.21: إضافة مُكوِّن VerticalArrangement (الترتيب العمودي)

لإضافة تسمية المدن:

> من مجموعة User Interface (واجهة المُستخدم)، أضف مُكوِّن Label (التسمية) إلى الشاشة، 1 وأعد تسميته إلى cities_label (تسمية_ المدن). 2

> في المُكوِّن cities_label (تسمية_ المدن)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، 3 واضبط الخاصية FontSize (حجم الخط) إلى 18.0، 4 واضبط خاصية Text (النص) إلى Cities (المدن) واضبط خاصية TextColor (لون النص) إلى White (أبيض). 5



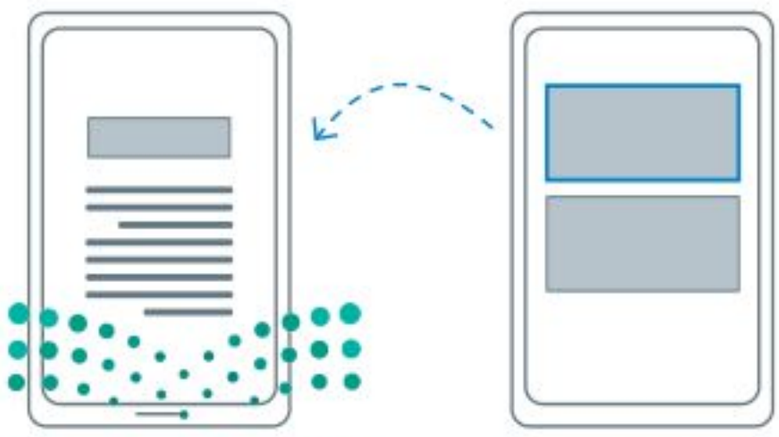
شكل 3.22: إضافة تسمية نصية

إنشاء قائمة Creating a List

القوائم هي نوع من هياكل البيانات التي تُستخدم لإنشاء مجموعات مختلفة من القيم أو العناصر وإدارتها، وستُضيف قائمة مع كل صورة تضيفها إلى التطبيق.

على سبيل المثال، عند الضغط على صورة مدينة الرياض، ستظهر قائمة باثنين من أبرز معالم هذه المدينة على النحو التالي:

- Al Masmak (المصمك).
- Boulevard Riyadh City (بوليفارد مدينة الرياض).



شكل 3.23: فتح صفحة جديدة من ListPicker (قائمة الخيارات)

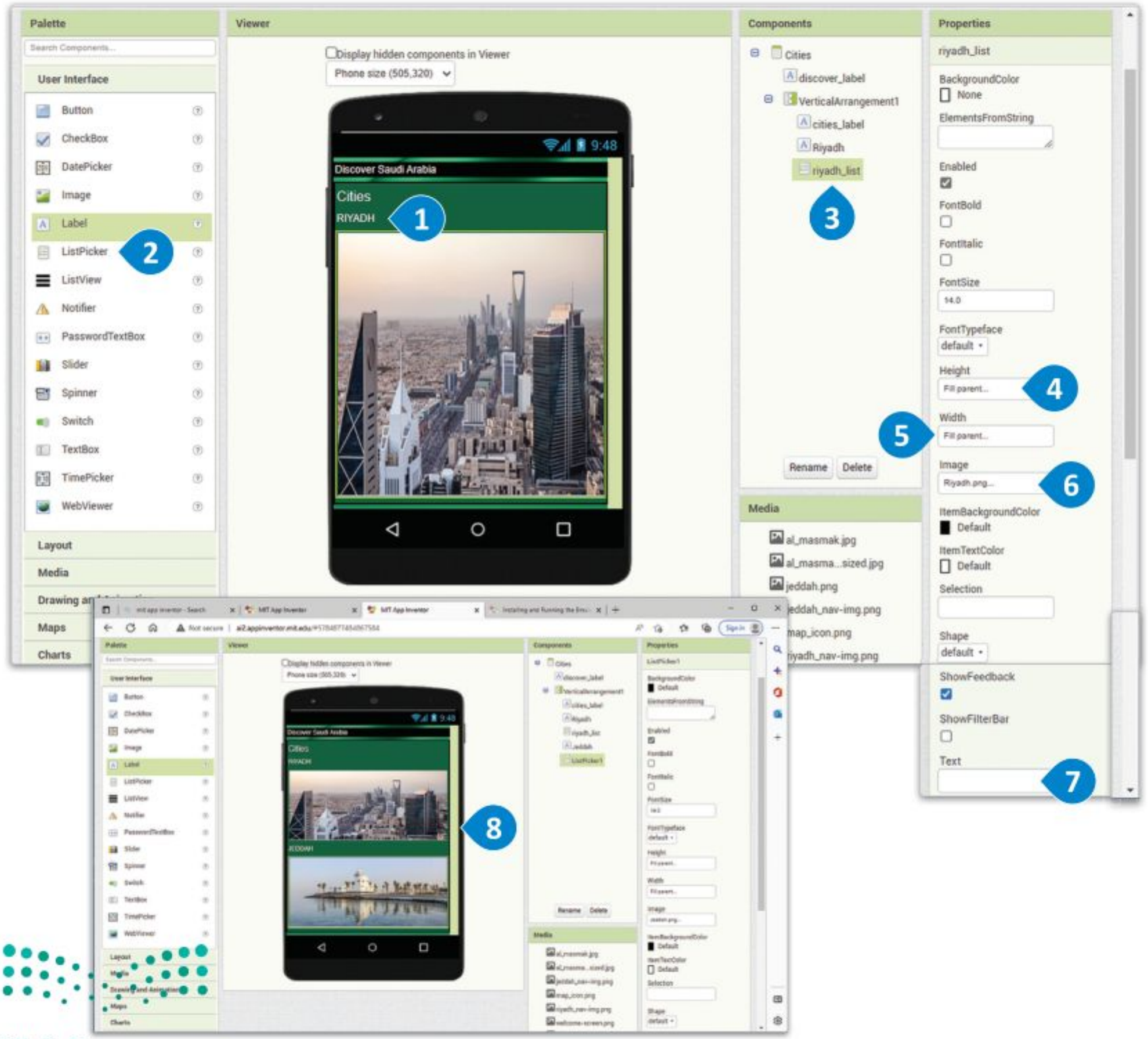
سيتم استخدام المُكوِّن ListPicker (قائمة الخيارات) لتحديد مَعْلَم كل مدينة، وسيتم تمثيل كل مدينة بواسطة القائمة، ولهذا سيكون هناك قائمة لمدينة الرياض وقائمة لمدينة جِدّة .

لإضافة المُكوِّن ListPicker (قائمة الخيارات) للرياض:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف Label (التسمية) باسم RIYADH (الرياض)، 1 ثم أضف مُكوِّن ListPicker (قائمة الخيارات) إلى Screen2 (شاشة 2)، 2 وأعد تسميته إلى riyadh_list (قائمة_الرياض). 3

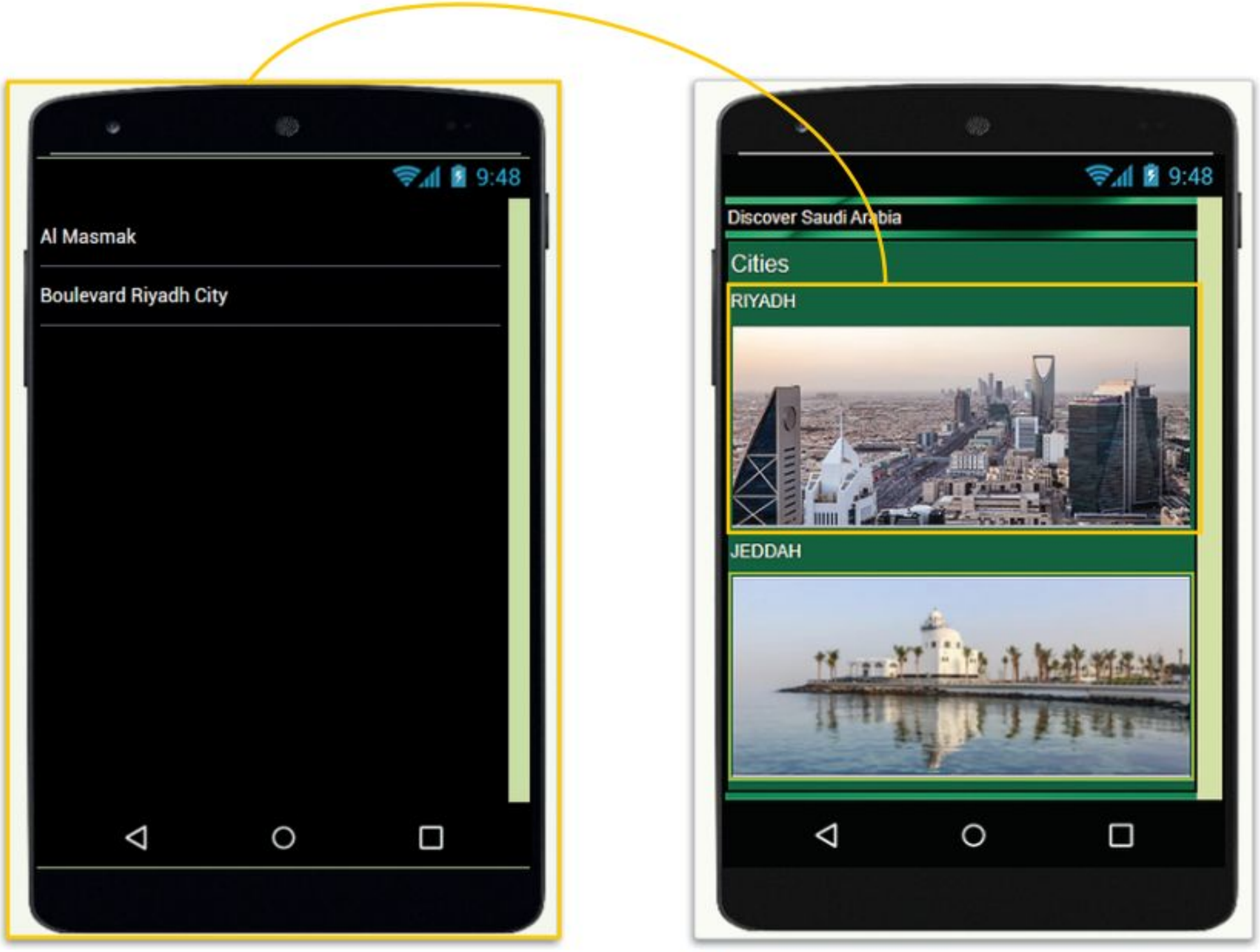
> في المُكوِّن riyadh_list (قائمة_الرياض)، اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، 4 وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة)، 5 ثم حمِّل صورة الرياض من خاصية Image (صورة)، 6 وامسح النص الموجود سابقًا داخل حقل Text (النص). 7

> كرِّر الخطوات لإضافة Label (التسمية) باسم JEDDAH (جِدّة) و ListPicker (قائمة الخيارات) الخاصة بها. 8



شكل 3.24: إضافة مُكوِّن ListPicker (قائمة الخيارات)

عند تشغيل التطبيق النهائي على هاتفك المحمول، سيعمل مُكوِّن ListPicker (قائمة الخيارات) بالطريقة التالية: عند تحديد صورة لكل قائمة، تتغير محتويات الشاشة لإظهار قائمة بالخيارات. فعلى سبيل المثال، عند الضغط على مُكوِّن ListPicker (قائمة الخيارات) الخاص بالرياض، سينفذ التطبيق الإجراء التالي:



شكل 3.25: مُكوِّن ListPicker (قائمة الخيارات)

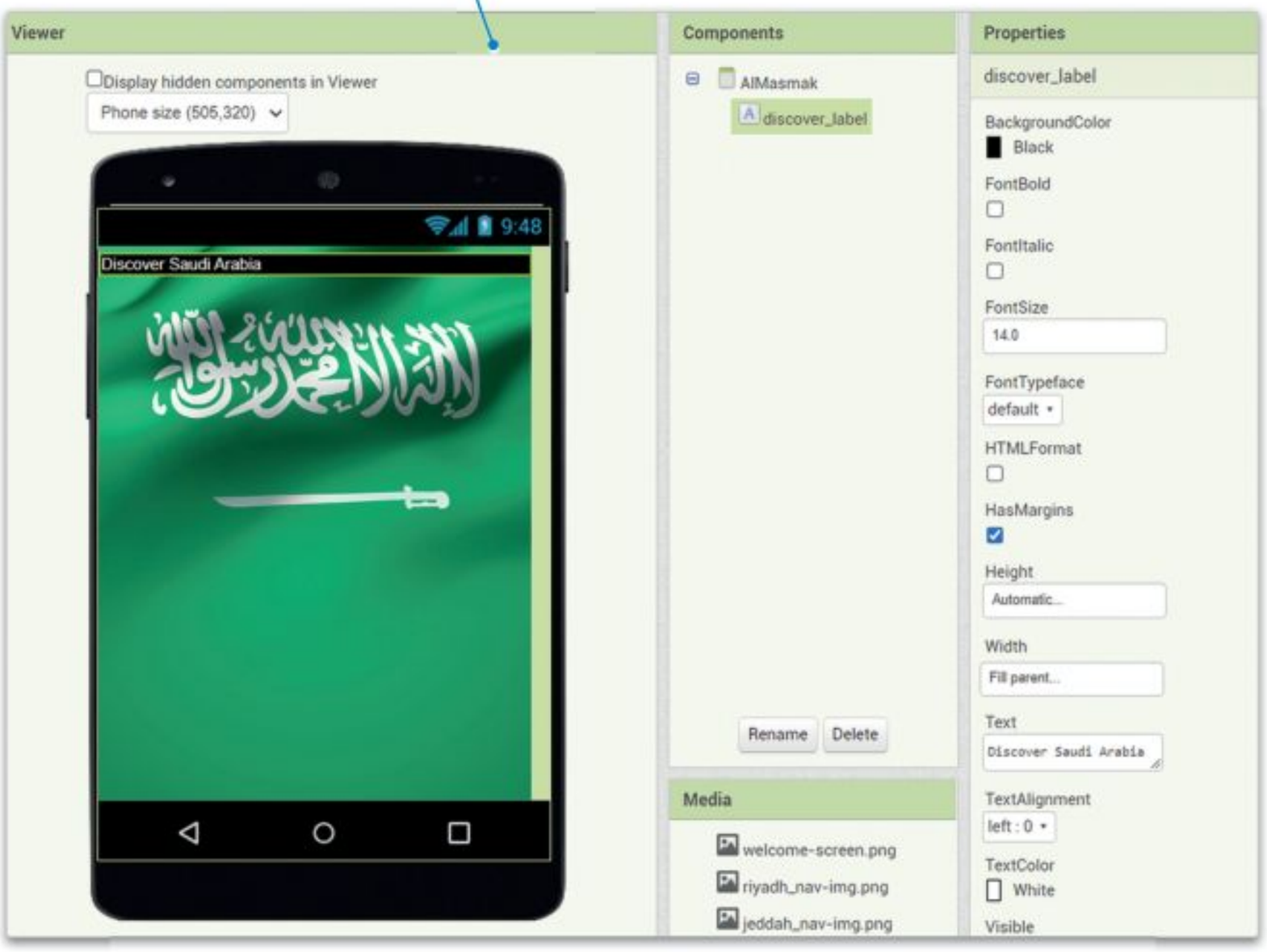
إنشاء الشاشة الثالثة للتطبيق Creating the Third Screen of the App

ستعرض الشاشة الثالثة والأخيرة في التطبيق المعلومات حول المَعْلَم المحدد من الشاشة السابقة، وستعرض هذه الشاشة عنوان المَعْلَم وصورة رمزية له ووصفًا نصيًا، وستعرض أيضًا زرًا للخريطة يتيح عرض خريطة تفاعلية تُظهر موقع المَعْلَم في المدينة المحددة، وسيكون لكل مَعْلَم شاشته المخصصة.

في هذا الدرس ستنشئ الشاشة الخاصة بالمصمك.

كما تعلمت سابقًا، أضف شاشة جديدة وأعد تسميتها إلى Al Masmak (المصمك)، ثم أضف عَلَم المملكة العربية السعودية كخلفية وعنوانًا للشاشة كما في الصورة أدناه.

ستحتاج إلى إزالة تسمية الشاشة الأصلية واستبدالها بالتسمية Discover Saudi Arabia (اكتشف المملكة العربية السعودية).



شكل 3.26: إنشاء الشاشة الثالثة

لإضافة عناصر على الشاشة:

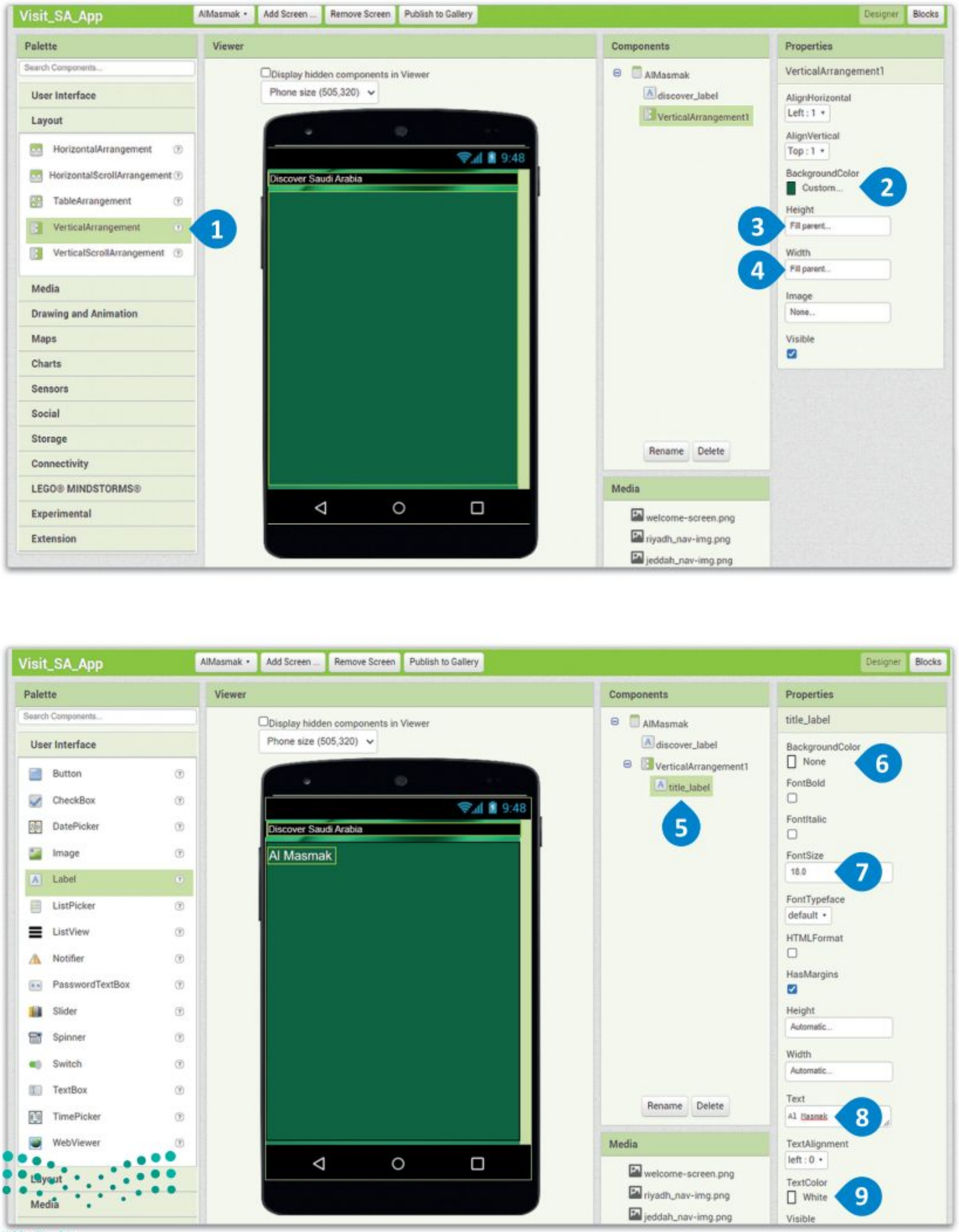
> من مجموعة Layout (التخطيط)، أضف مُكوِّن VerticalArrangement (الترتيب العمودي) إلى الشاشة. (1)

> في مُكوِّن VerticalArrangement1 (الترتيب العمودي1)، اضبط خاصية BackgroundColor (لون الخلفية) إلى Custom (مُخصص)، (2) ثم اكتب القيمة #11613eff.

> في مُكوِّن VerticalArrangement1 (الترتيب العمودي1)، اضبط خاصية Height (الارتفاع) على Fill Parent (تعبئة المساحة)، (3) وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة). (4)

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Label (التسمية) إلى الشاشة، وأعد تسميته إلى title_label (تسمية_ العنوان). (5)

> في مُكوِّن title_label (تسمية_ العنوان)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (6) ثم اضبط خاصية FontSize (حجم الخط) إلى 18.0، (7) واضبط خاصية Text (النص) إلى Al Masmak (المصمك)، (8) واضبط خاصية TextColor (لون النص) إلى White (أبيض). (9)



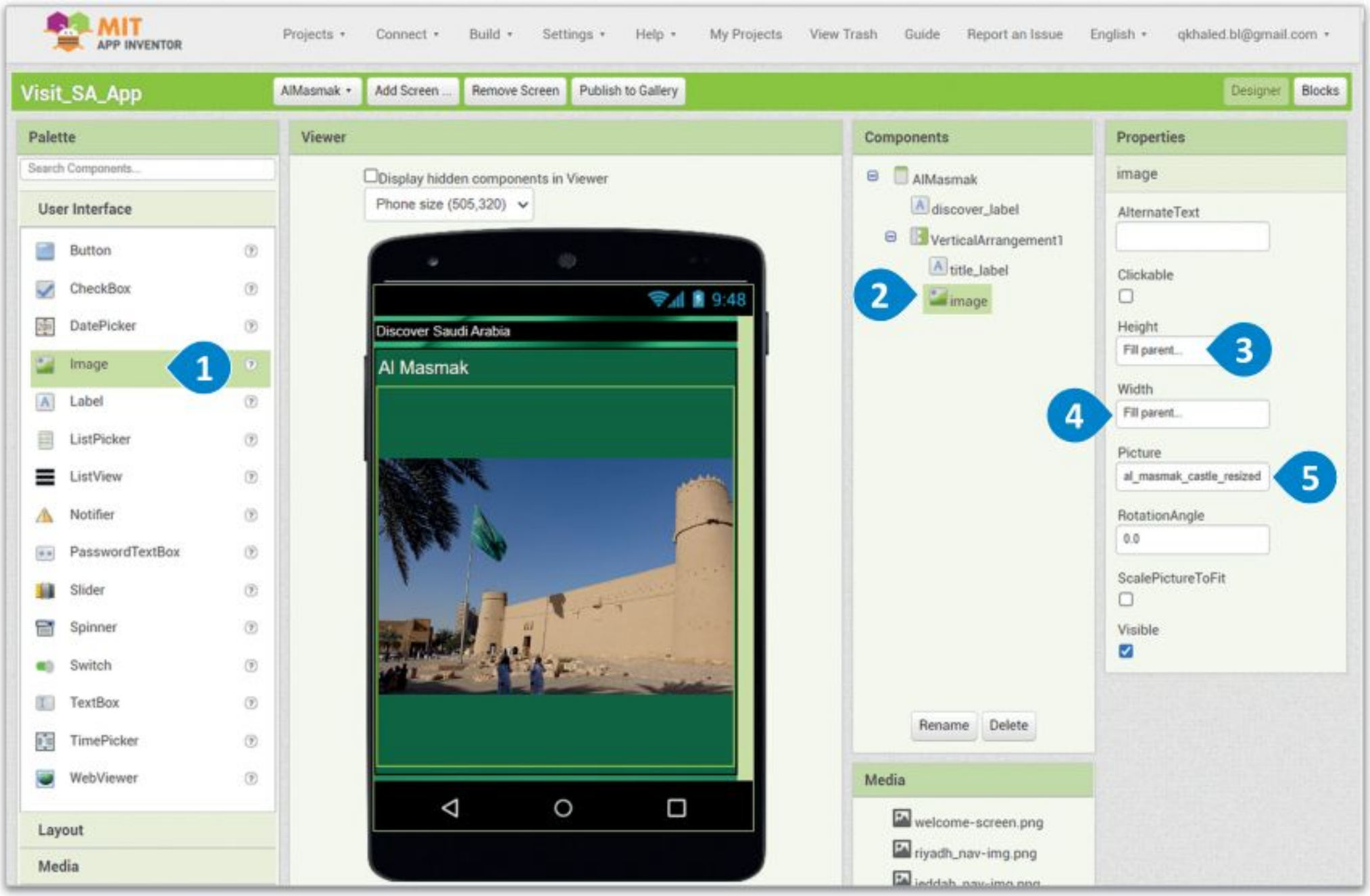
شكل 3.27: إضافة العناصر إلى الشاشة الثالثة

لإضافة المُكوِّن image (صورة):

> من مجموعة User Interface (واجهة المُستخدِم)، أضف المُكوِّن Image (صورة) إلى Screen1 (شاشة 1)، (1) وأعد تسميته إلى image (صورة). (2)

> في المكوِّن image (صورة)، اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، (3) وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة)، (4) واضبط خاصية Picture (صورة) إلى صورة قصر المصمك. (5)

لا يمكن أن يتشابه اسم المُكوِّن مع تسميته، لكن تسمية image هنا تختلف عن المُكوِّن Image.



شكل 3.28: إضافة المُكوِّن image (صورة)

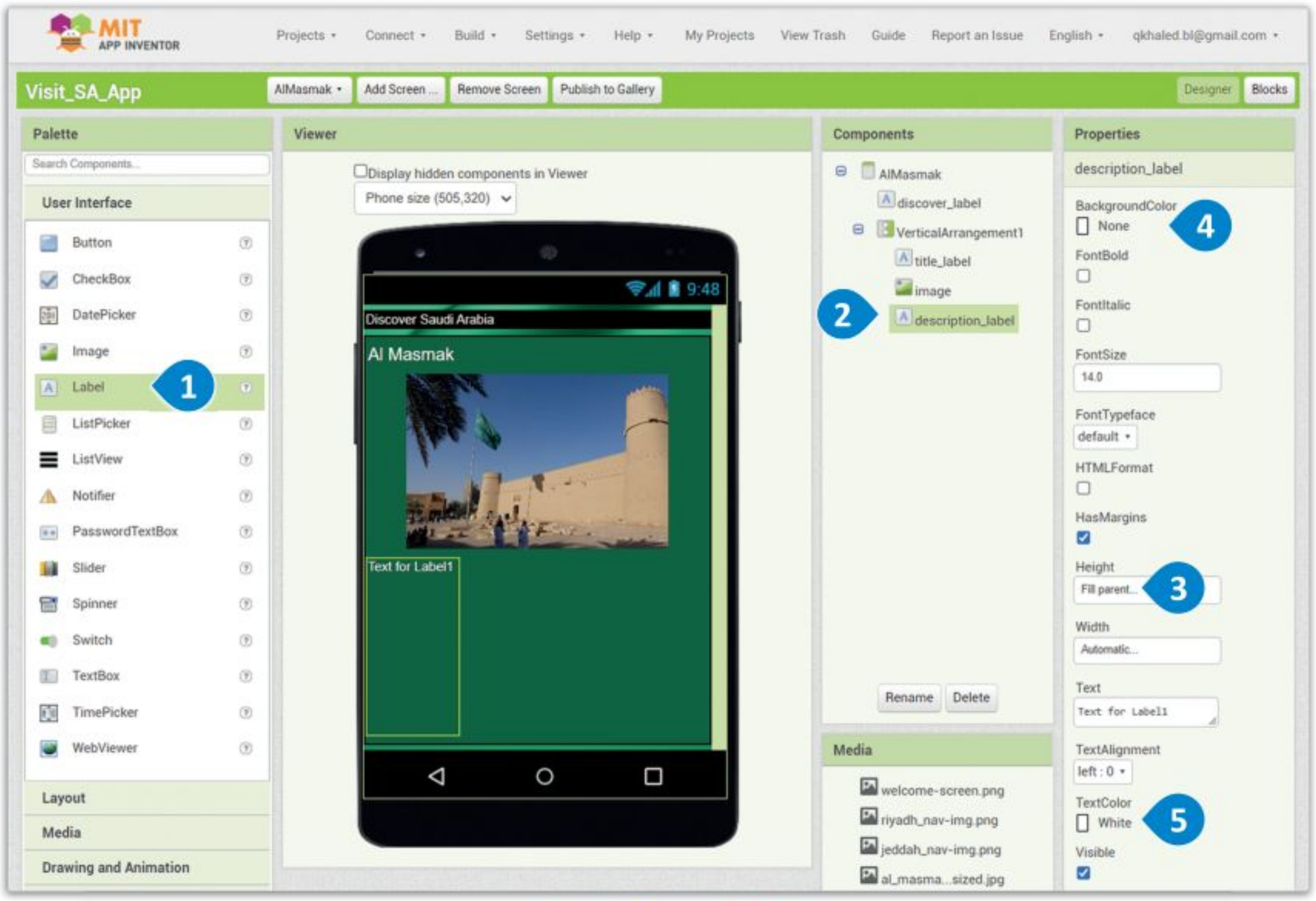
ستضيف الآن Label (تسمية) تحتوي على وصف لقصر المصمك، ولكن في هذه المرحلة ستضيف اختصارًا للنص الذي ستتم إضافته في الدرس التالي.

لإضافة مُكوِّن الوصف النصي:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Label (التسمية) إلى الشاشة، (1) وأعد تسميته إلى description_label (تسمية_ الوصف). (2)

> في مُكوِّن description_label (تسمية_ الوصف)، اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، (3) واضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (4) وخاصية TextColor (لون النص) إلى White (أبيض). (5)

سيتم تغيير حجم الصورة بشكل صحيح عند إضافة المزيد من المُكوِّنات.



شكل 3.29: إضافة مُكوِّن الوصف النصي

إضافة خريطة تفاعلية للتطبيق Adding an Interactive Map to the Application

سيتمكن المُستخدِمون من تصفح خريطة تفاعلية تعرض الموقع الدقيق للمَعلَمْ من الشاشة الخاصة به، وذلك ليتمكنوا من رؤية جميع معالم المدينة. ستنشئ في البداية مُجمَّعًا للزرّ الذي سيعرض الخريطة التفاعلية، ثم ستضيف المُكوِّن.

مُكوِّن الترتيب الأفقي Horizontal Arrangement Component

- باستخدام مُكوِّن الترتيب الأفقي، يتم ترتيب الكائنات أفقيًا على طول المحور الأفقي ومحاذاتها رأسيًا في الوسط.
- إذا تم تعيين خاصية الارتفاع أو العرض إلى Automatic (تلقائي)، سيتمّ تحديد الارتفاع الفعلي للمُكوِّن حسب ارتفاع أطول كائن بداخله.
- إذا كانت خاصية Height (الارتفاع) لمُكوِّن HorizentalArrangment (الترتيب الأفقي) فارغة، فسيكون الارتفاع 100.
- إذا تم تحديد خاصية Height (الارتفاع) أو Width (العرض) لمُكوِّن HorizontalArrangement (الترتيب الأفقي) بواسطة Fill Parent (تعبئة المساحة) أو Pixel (البكسل)، فإن أي خاصية Width (عرض) محددة بواسطة Fill Parent (تعبئة المساحة) ستشغل أيضًا أي مساحة لا تشغلها المُكوِّنات الأخرى.

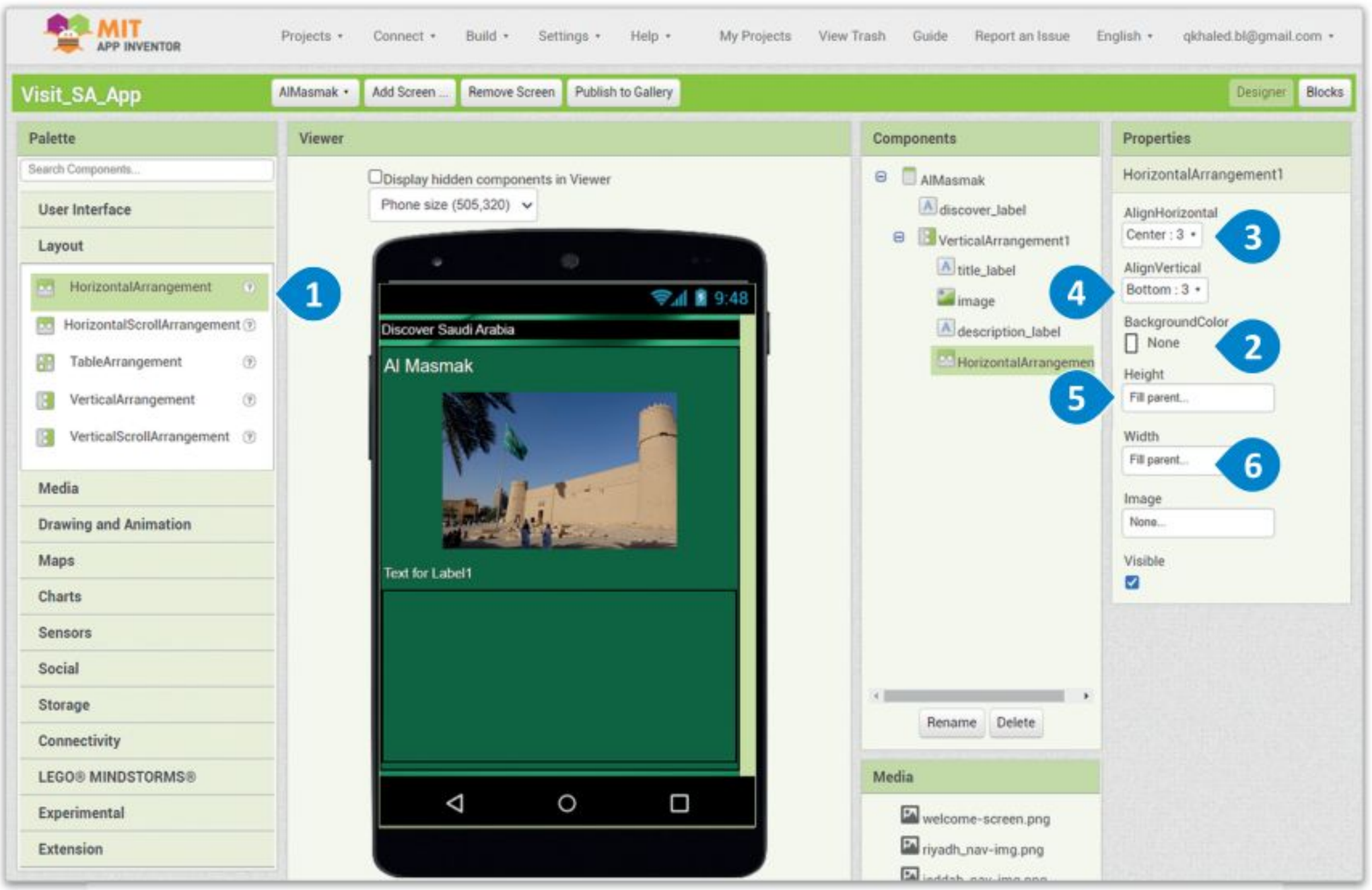
وزارة التعليم
Ministry of Education
2023 - 1445

لإضافة مُكوِّن HorizontalArrangement (الترتيب الأفقي):

> من مجموعة Layout (التخطيط)، أضف مُكوِّن HorizontalArrangement (الترتيب الأفقي) إلى الشاشة. (1)

> في مُكوِّن HorizontalArrangement1 (الترتيب الأفقي1)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (2) واضبط خاصية AlignHorizontal (المحاذاة الأفقية) إلى 3 :Center (المنتصف:3)، (3) واضبط الخاصية AlignVertical (المحاذاة العمودية) إلى 3 : Bottom (الأسفل:3). (4)

> اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، (5) وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة). (6)

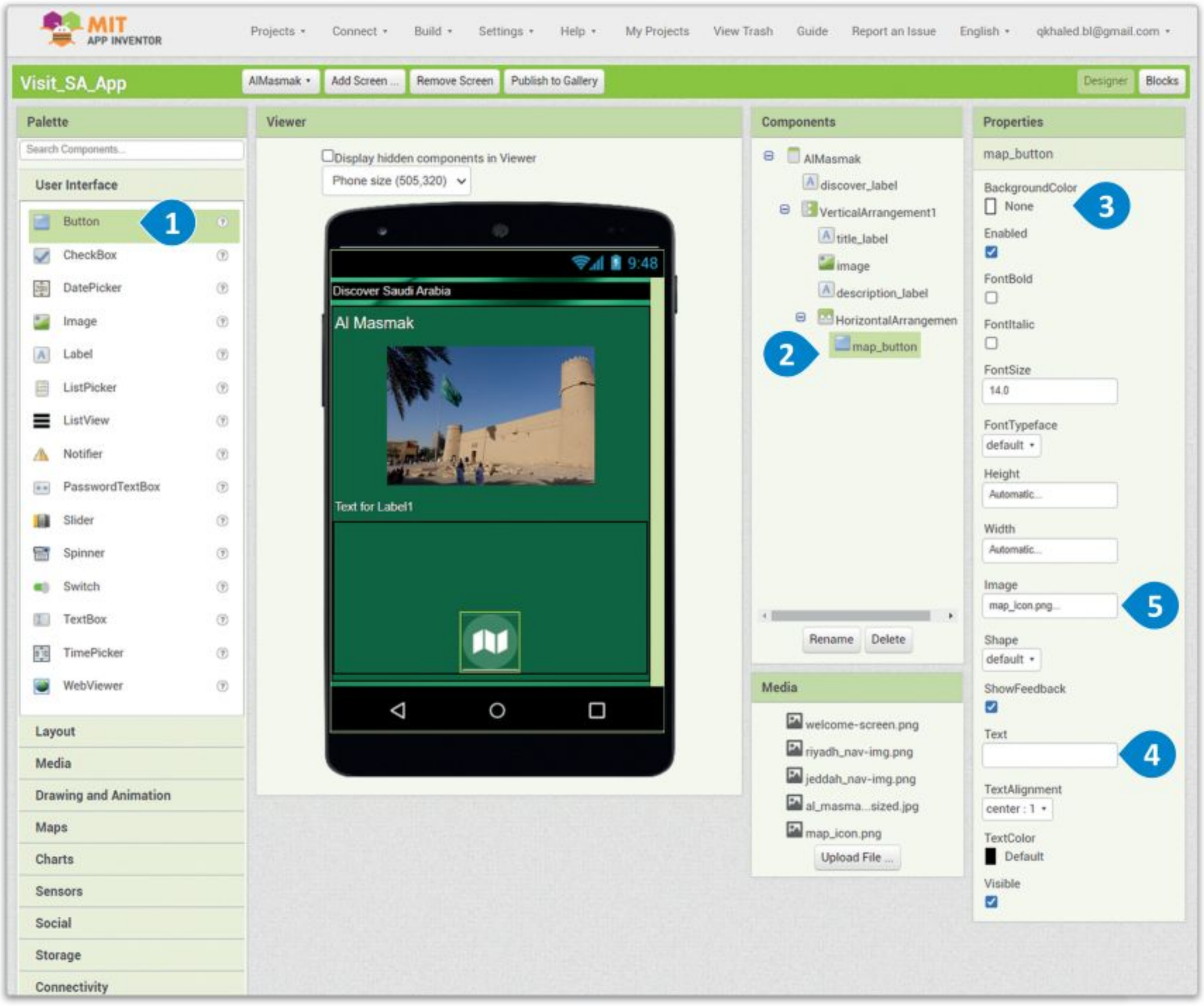


شكل 3.30: إضافة مُكوِّن HorizontalArrangement (الترتيب الأفقي)

لإضافة زرّ Map (الخريطة):

> من مجموعة User Interface (واجهة المُستخدم)، أضف المُكوِّن Button (زرّ) إلى Screen1 (شاشة 1)، (1) وأعد تسميته إلى map_button (زرّ_الخريطة). (2)

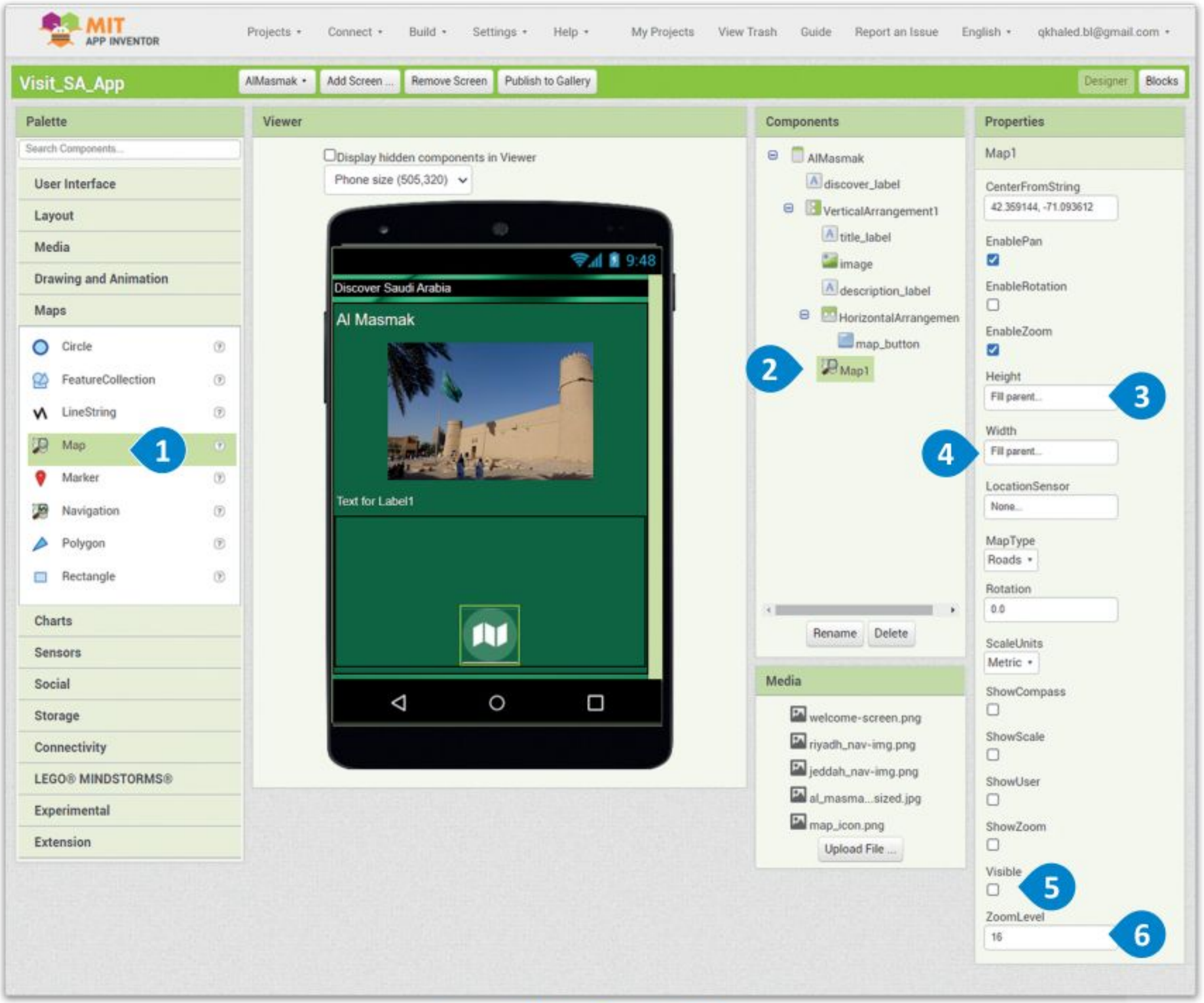
> في المُكوِّن map_button (زرّ_الخريطة)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (3) وامسح النص من حقل Text (النص)، (4) واضبط خاصية Image (صورة) إلى أيقونة خريطة. (5)



شكل 3.31: إضافة زرّ Map (الخريطة)

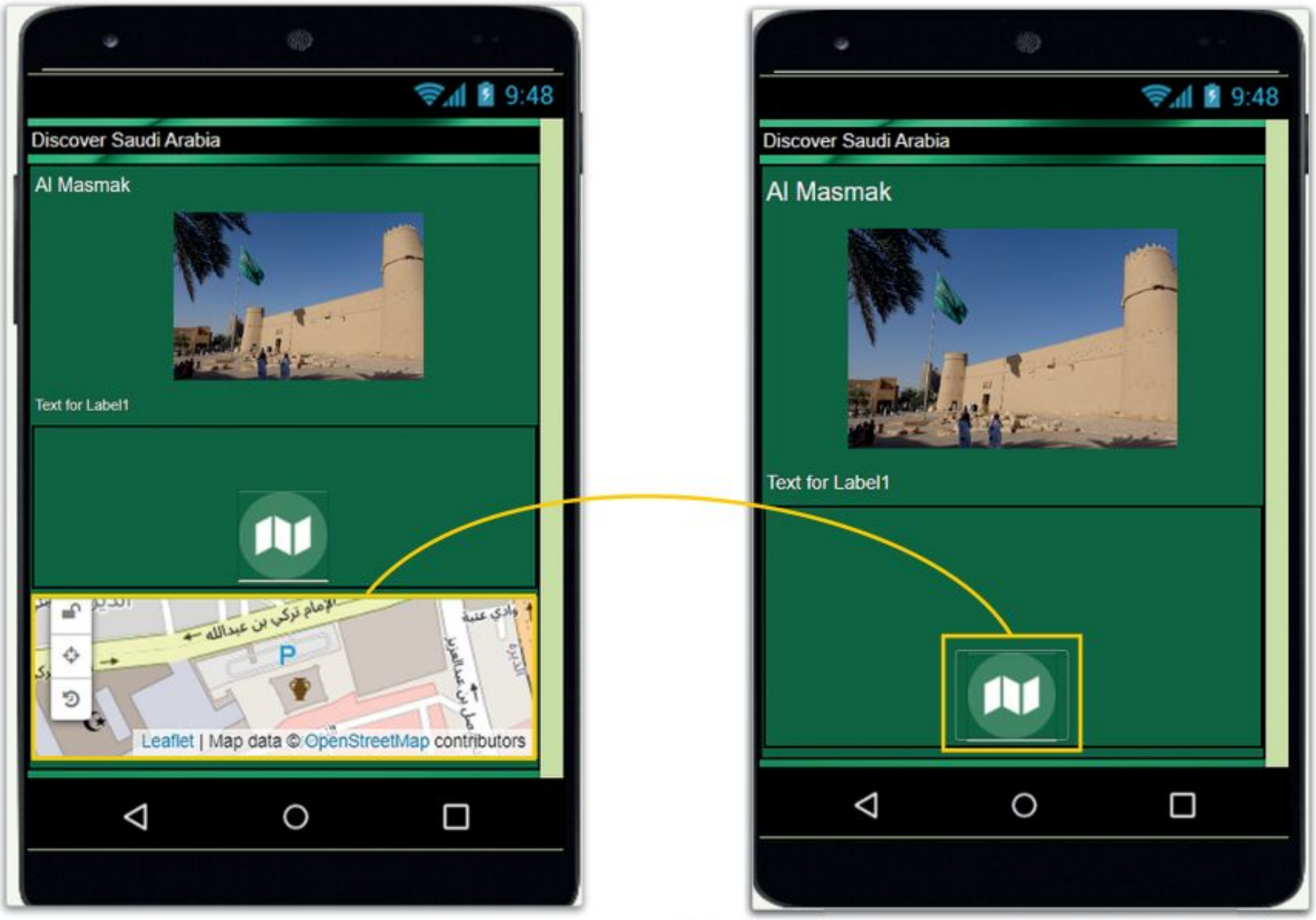
لإضافة مُكوِّن Map (الخريطة):

- من مجموعة Maps (الخرائط)، حدِّد المُكوِّن Map (خريطة)، 1 وضعه ضمن VerticalArrangment1 (الترتيب العمودي1). 2
- اضبط خاصية Height (الارتفاع) إلى Fill Parent (تعبئة المساحة)، 3 وخاصية Width (العرض) إلى Fill Parent (تعبئة المساحة). 4
- ألْغِ تحديد خاصية Visible (مرئي). 5
- اضبط خاصية ZoomLevel (مستوى التكبير) إلى الرقم 16. 6



شكل 3.32: إضافة مُكوِّن Map (الخريطة)

عند تشغيل التطبيق النهائي على هاتفك المحمول، يظهر مُكوِّن الخريطة في موقع المَعْلَم المحدد، وستضيف في الدرس التالي الإحداثيات برمجيًا بناءً على هذا الموقع.



شكل 3.33: تفعيل مُكوّن الخريطة بواسطة زرّ الخريطة

يمكن التفاعل مع مُكوّن الخريطة بالطريقة نفسها المعروفة في خرائط التطبيقات الأخرى.

تمرينات

1 صِفْ كيف تساعدك مُكوِّنات HorizontalArrangement (الترتيب الأفقي)، وVerticalArrangement (الترتيب العمودي) في تشكيل مُخطط شاشة الهاتف المحمول.

2 حدِّد الفرق بين المُكوِّن ListPicker (قائمة الخيارات) والمُكوِّن Button (زرّ).

3 صمّم تطبيقًا بترتيب عمودي وترتيبين أفقيين، ويجب أن يحتوي كل ترتيب أفقي على زرّين، وأن تكون جميع المُكوّنات في مُجمَّع خاص بهم. استخدم خصائص المحاذاة المناسبة للمُكوّنات.

4 صمّم شاشة أخرى للتطبيق أعلاه تحتوي على HorizontalArrangement (الترتيب الأفقي) كمُجمع خارجي، وترتيبين عموديين VerticalArrangement (ترتيب عمودي) مع أزرار بداخلهما، وستكون جميع المُكوّنات في منتصف المُجمَّع الخاص بها. استخدم خصائص المحاذاة المناسبة للمُكوّنات.

5 صمّم شاشة أخرى للتطبيق أعلاه تحتوي على VerticalArrangement (ترتيب عمودي)، وثلاثة صفوف من HorizontalArrangement (الترتيب الأفقي)، وسيحتوي كل صف HorizontalArrangement (ترتيب أفقي) على صورتين، حيث تشير كل صورة إلى رياضة مختلفة. تأكد من أن جميع المُكوّنات مرتبة في وسط مُجمَّعها وأن كافة الصور لها الأبعاد نفسها.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

برمجة تطبيق الهاتف المحمول

تطبيقات البرمجة في مخترع التطبيقات

Programming Applications in App Inventor

قبل البدء بتطوير التطبيقات باستخدام اللبنات البرمجية، ستتعرف على بعض المفاهيم والأوامر الأساسية في عملية التطوير مثل:التعامل مع البيانات المتغيرة، وتنفيذ منطق البرنامج وتدفقه.

المتغيرات في مخترع التطبيقات

يتيح لك **مخترع التطبيقات** (App Inventor) إنشاء المتغيرات والتحكم بها، حيث يمكن تكوين المتغيرات وتهيئتها بأنواع متعددة من البيانات، مثل الأرقام العشرية والسلاسل النصية. يوجد للمتغيرات في مخترع التطبيقات نطاقات مخصصة لعملها كما يلي:

- **عام** (Global): يمكن الوصول إلى المتغيرات من خلال جميع عمليات التحكم والإجراءات واللبنات البرمجية.
- **محلي** (Local): لا يمكن الوصول إلى المتغيرات إلا داخل الإجراء الذي يتضمنها.

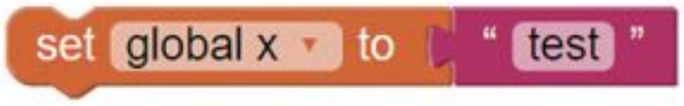
يتم استخدام المتغيرات المحلية لتوفير حجم ذاكرة الجهاز، حيث يتم إنشاؤها والوصول إليها فقط عند الحاجة إليها في الإجراء.

سيقتصر استخدامك في هذا المشروع على المتغيرات العامة، حيث يخلو هذا المشروع من الإجراءات المعقدة التي تحتاج إلى متغيرات محلية.

تهيئة متغير عام

الحصول على متغير عام

ضبط متغير عام

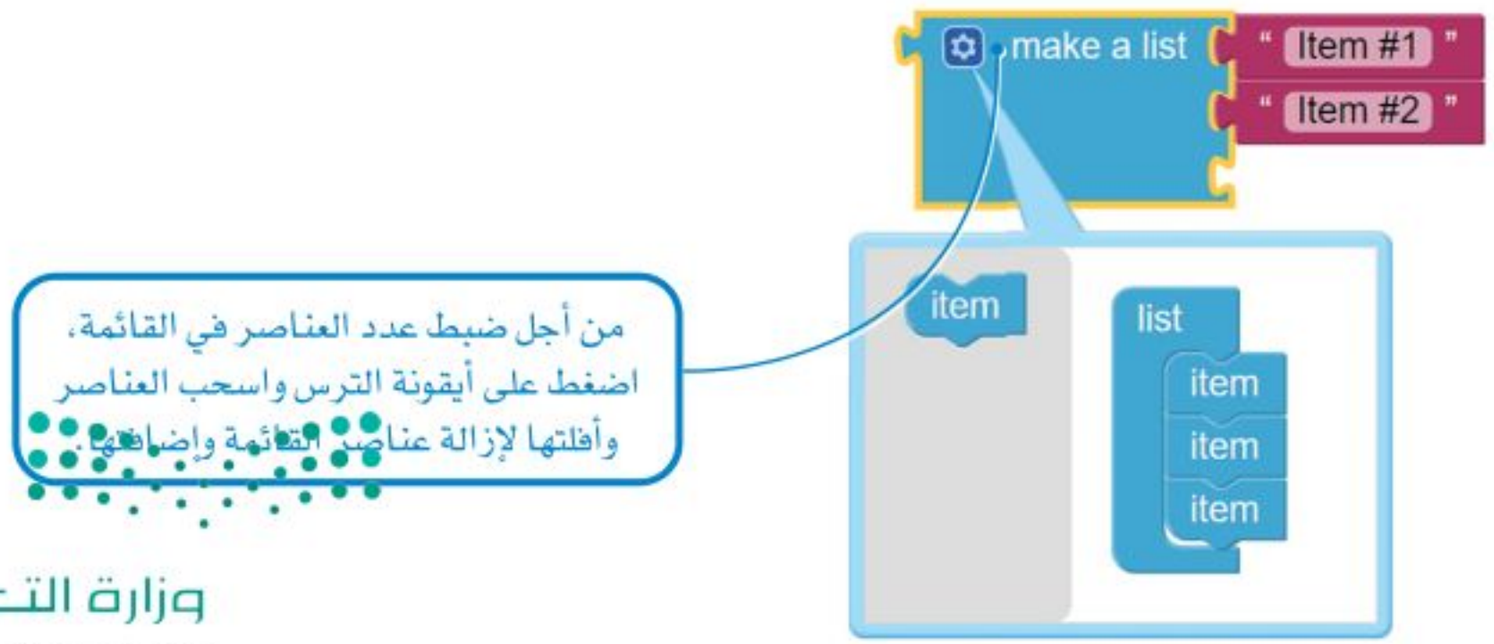


إنشاء قائمة فارغة

تهيئة القائمة بالبيانات

القوائم في مخترع التطبيقات

يمكن تعريف القوائم بأنها هيكل بيانات بسيط ومفيد يمكن استخدامه لتنفيذ **منطق التطبيق** (Application Logic)، ويقدم مخترع التطبيقات طُرقًا لمعالجة البيانات في القوائم، وللتفاعل معها.



مُكوِّن قائمة الخيارات The ListPicker Component

تُتيح لك ListPicker (قائمة الخيارات) إنشاء منطق خاص بالتطبيق اعتمادًا على اختياراتك، وعند الضغط على مُكوِّن ListPicker (قائمة الخيارات) يتغير مظهر الشاشة لعرض محتويات القائمة، وتسمى خاصية المُكوِّن التي تُخزن بيانات القائمة باسم Elements (العناصر).

الوصول إلى عناصر ListPicker (قائمة الخيارات)

ستأخذ ListPicker (قائمة الخيارات) متغيرًا يحتوي على قائمة كعناصر.

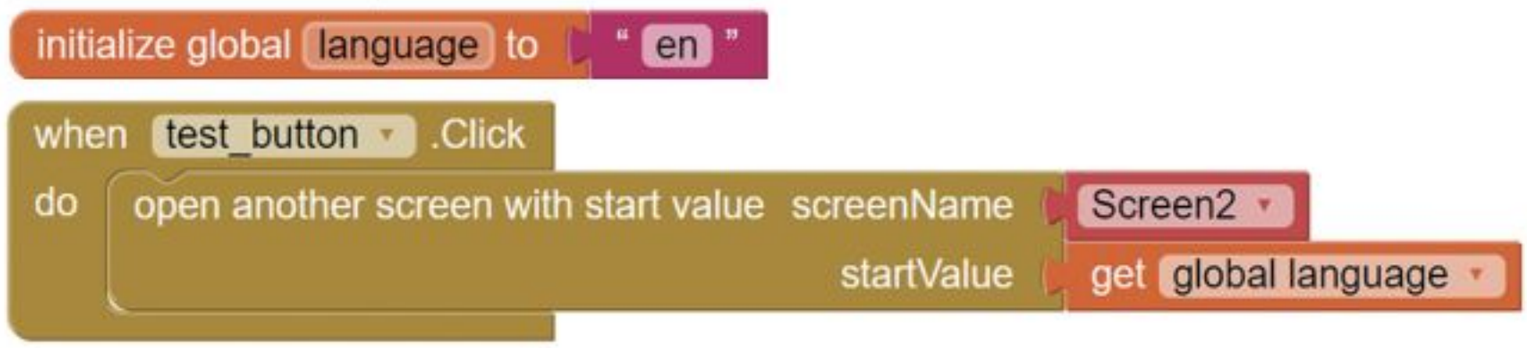
تهيئة عناصر ListPicker (قائمة الخيارات) باستخدام بيانات القائمة

التفاعل مع تحديد عنصر في ListPicker (قائمة الخيارات)

سيفتح التطبيق الشاشة ذات الاسم المُحدد في ListPicker (قائمة الخيارات).

إرسال المتغيرات إلى شاشة أخرى Sending Variables to Another Screen

عليك إرسال قيمة متغير اللغة إلى الشاشة التالية من أجل تهيئة النص في المتغير المناسب. يتيح مخترع التطبيقات للبرنامج إرسال قيمة تهيئة يمكن الوصول إليها بواسطة الشاشة التالية، وذلك عند استخدام أمر لفتح شاشة أخرى.

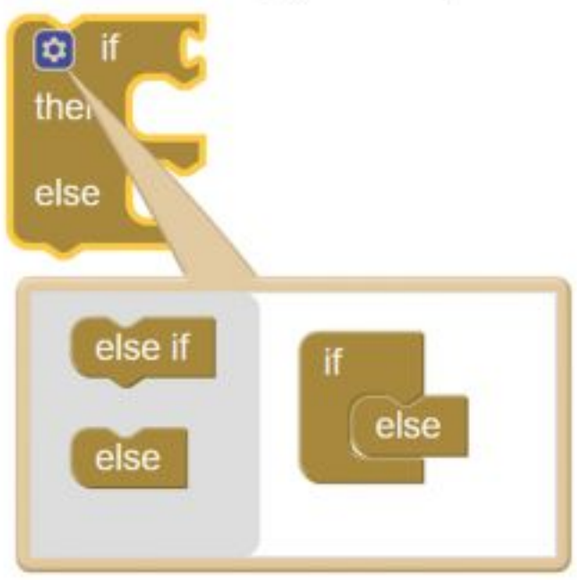


العبارات الشرطية If في مخترع التطبيقات Conditional If Statements in App Inventor

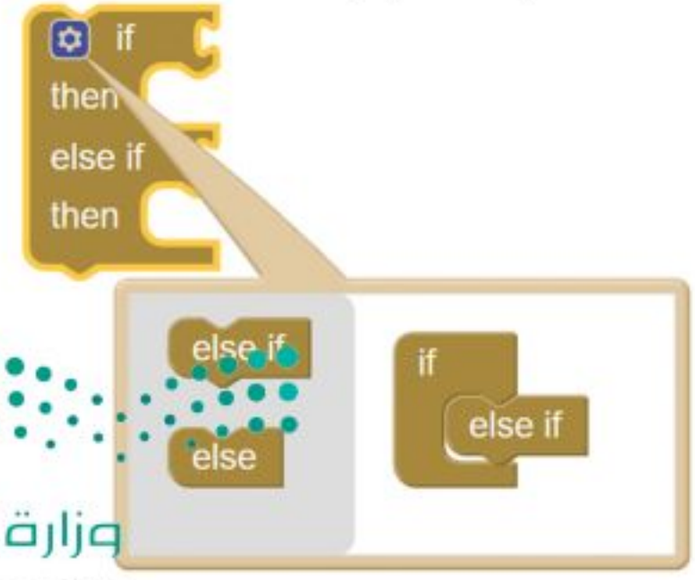
تم إنشاء اللبنات البرمجية If الشرطية في مخترع التطبيقات بشكل مشابه للقوائم، ويمكنك إضافة عبارتيّ if أو else if إلى اللبنات البرمجية باستخدام أيقونة الترس.

إضافة عبارة if البسيطة

إضافة عبارة else



إضافة عبارة else if



مُكوِّن ترتيب التمرير العمودي

The VerticalScrollArrangement Component

يتم استخدام مُكوِّن VerticalScrollArangement (ترتيب التمرير العمودي) لتنسيق ظهور المُكوِّنات الأخرى على المحور الرأسي، وإنشاء مُجمَّع لمحاذاتها. كما يدعم خاصية scrolling (التمرير) للمُكوِّنات التي لا تتناسب مع حجم الشاشة.

يمكن تغيير محاذاة المُكوِّنات الموجودة داخل المُجمَّع باستخدام اللبنتين البرمجيتين التاليتين:

مُكوِّن الترتيب الأفقي The HorizontalArrangement Component

يتم استخدام المُكوِّن HorizontalArrangement (الترتيب الأفقي) لتنسيق ظهور المُكوِّنات الأخرى أفقيًا، وإنشاء مُجمَّع لمحاذاتها. يمكن تغيير محاذاة المُكوِّنات الموجودة داخل المُجمَّع باستخدام اللبنتين البرمجيتين التاليتين:

برمجة الشاشة الرئيسة Programming the Home Screen

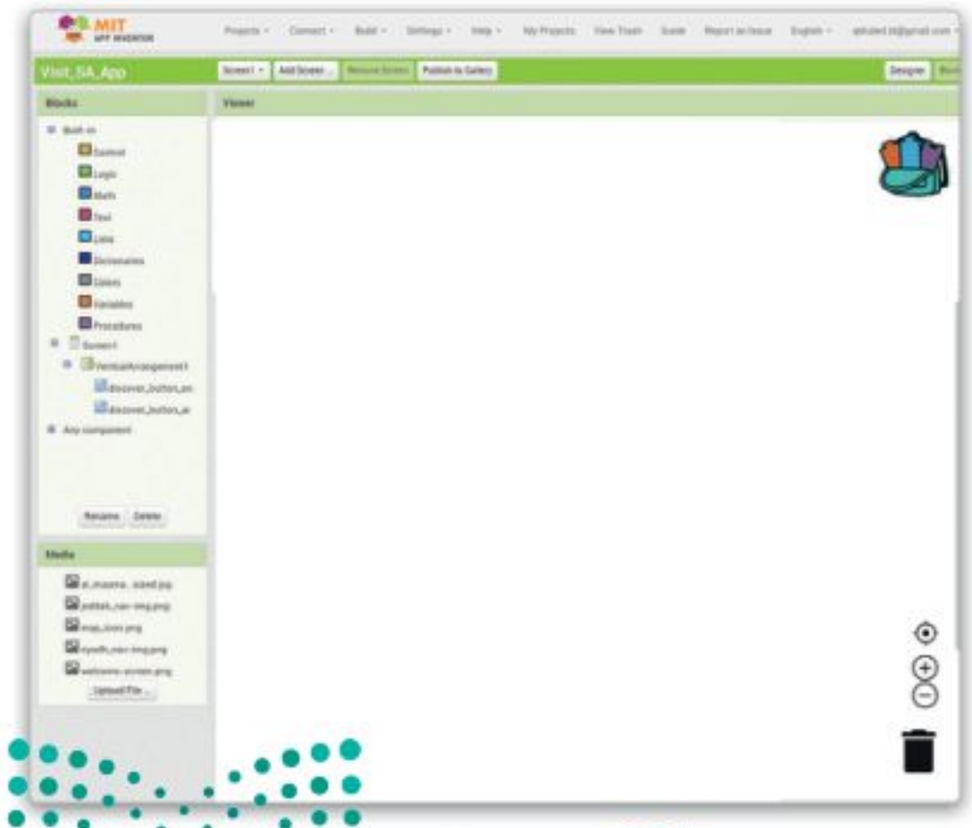
ستنقل الشاشة الرئيسة Screen1 (شاشة 1) المُستخدِم إلى شاشة Cities (المدن)، وستحدّد اللغة التي سيتم استخدامها على الشاشات التالية أيضًا.

برمجة أزرار دعم اللغة Programming Language Support Buttons

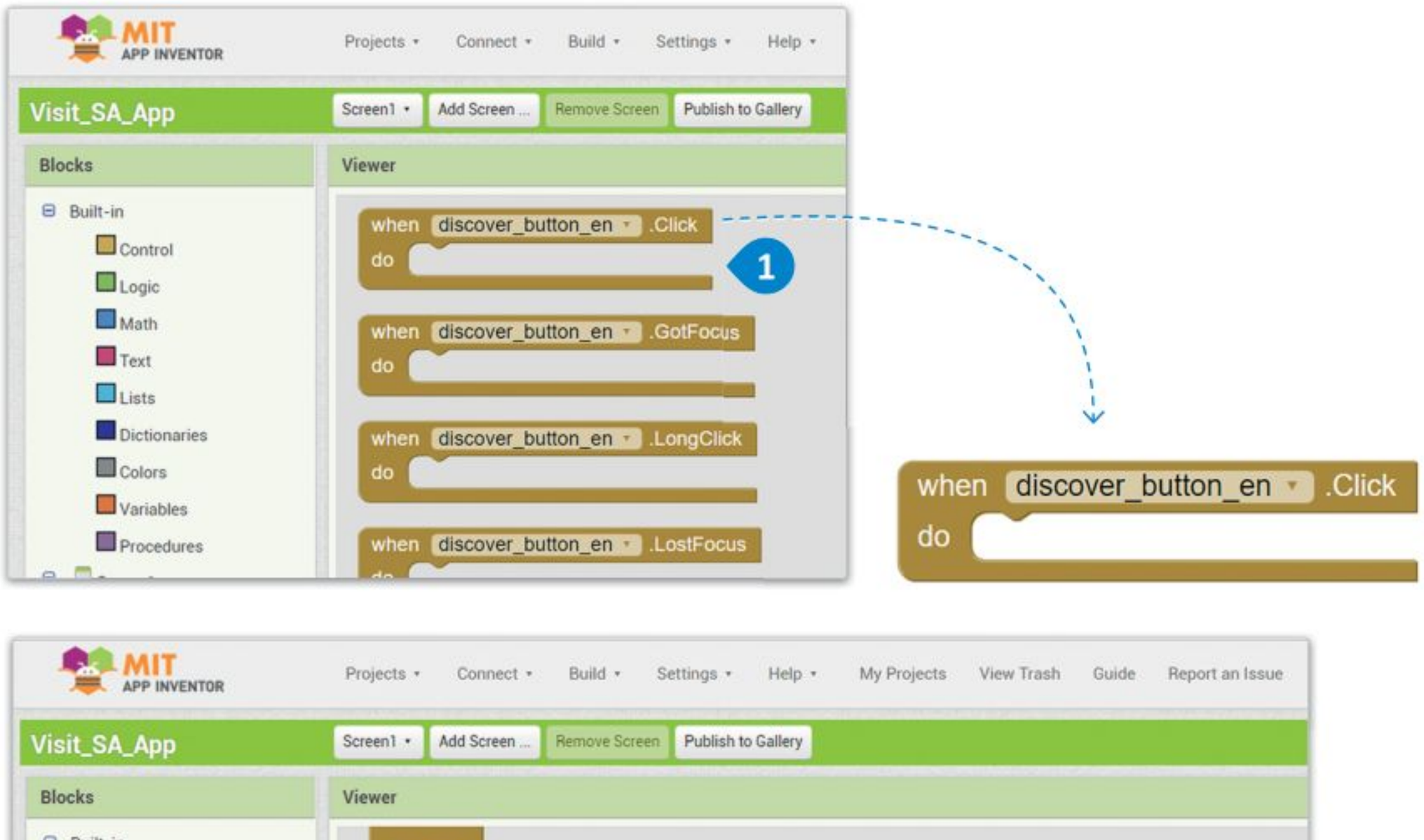
ستبرمج الآن أزرار اللغة لتغيير النص على الصفحة الرئيسة وتخزين متغير تحديد اللغة المناسبة للنص في الشاشة التالية. ستكون اللغة الإنجليزية هي اللغة الافتراضية لكل الصفحات.

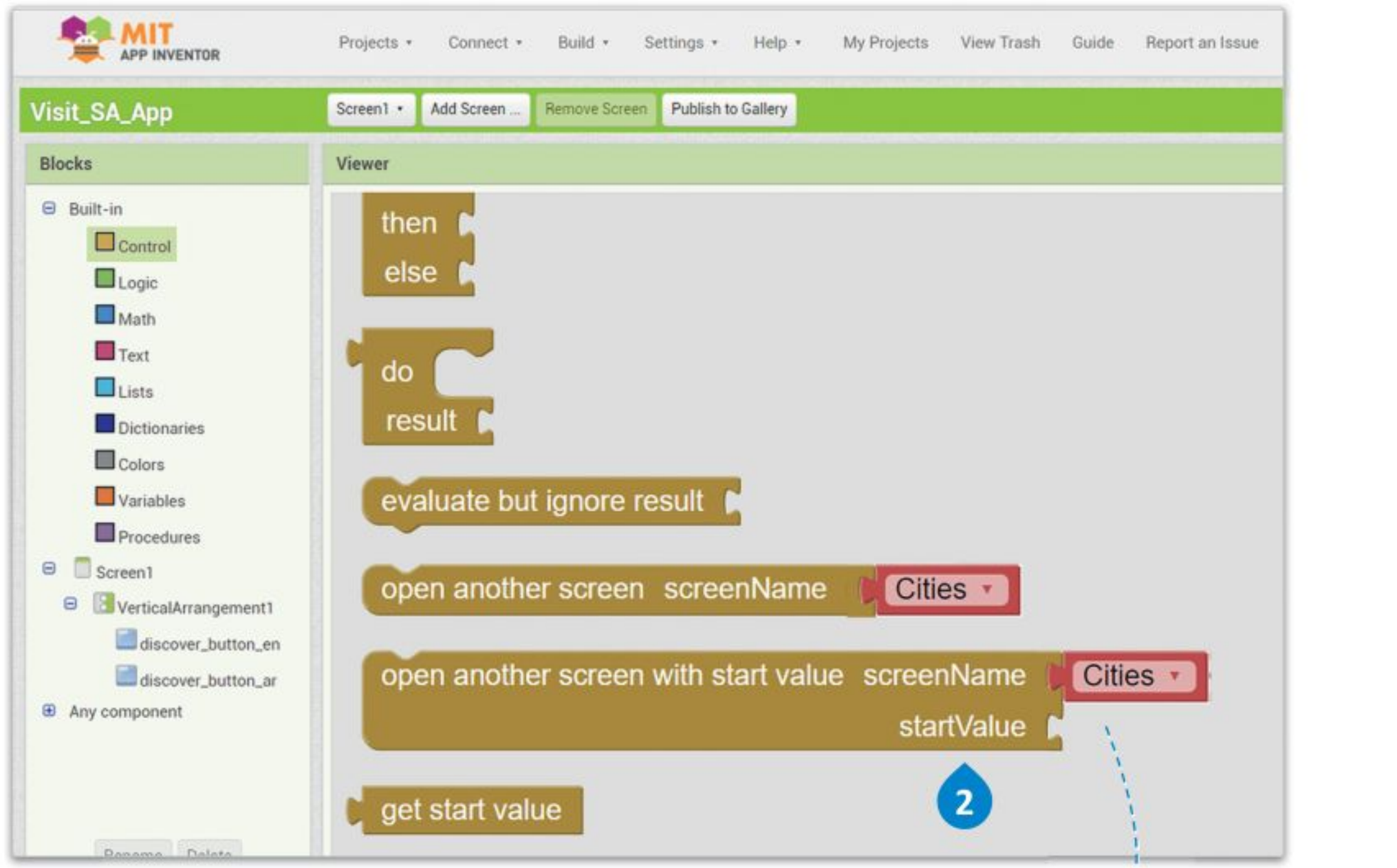
لبرمجة أزرار اللغة في الشاشة الرئيسة:

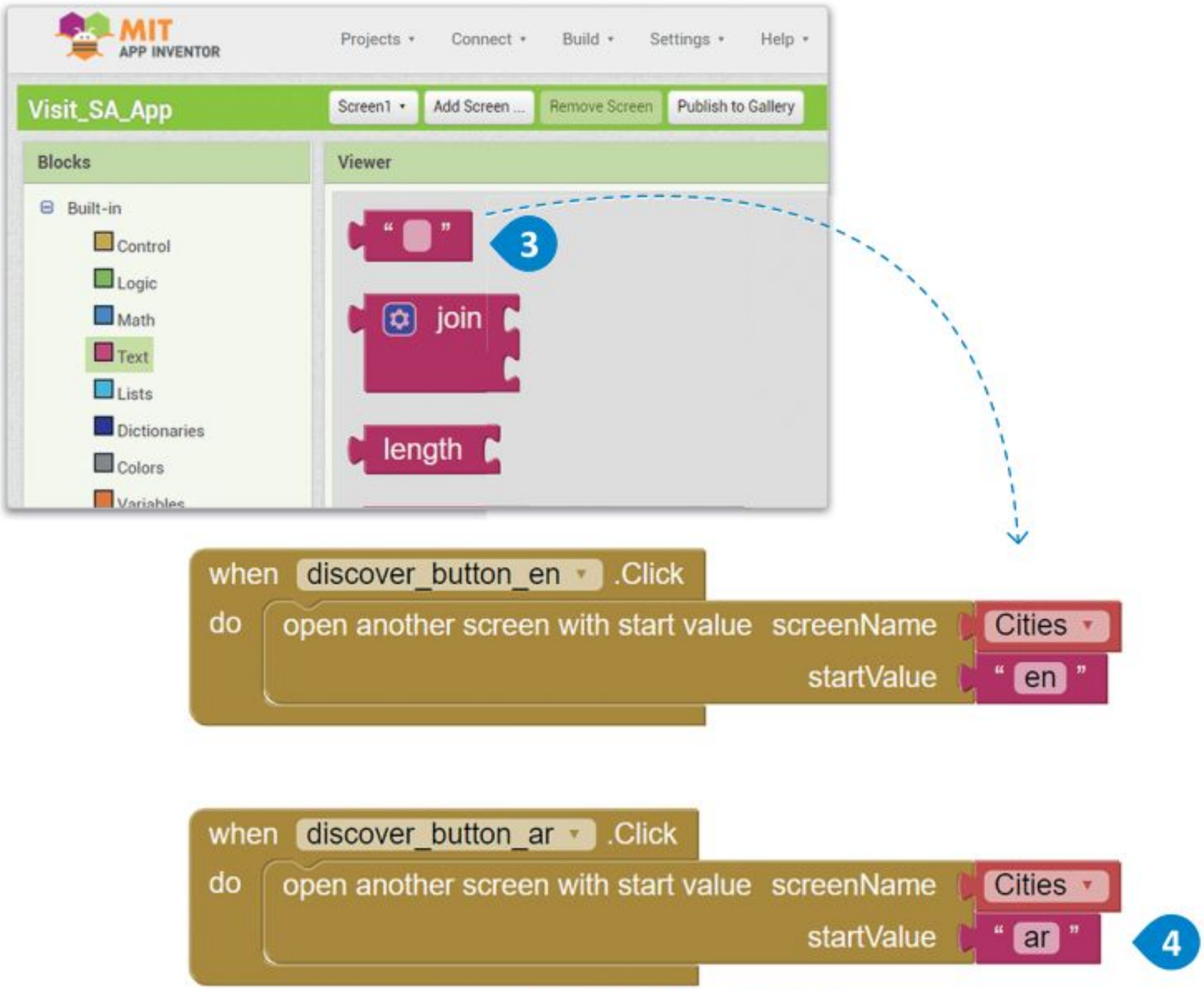
- حدِّد لبنة when.Click (عند.الضغط) لمُكوِّن discover_button_en (زرّ_اكتشف_ بالإنجليزية). 1
- حدِّد مجموعة التحكم وافتح شاشة Cities (المدن) باستخدام startValue (قيمة البداية). 2
- اضبط startValue (قيمة البداية) إلى en (الإنجليزية). 3
- كرِّر الخطوات أعلاه لمُكوِّن discover_button_ar (زرّ_اكتشف_ بالعربية) واضبط startValue (قيمة البداية) إلى ar (العربية). 4



شكل 3.34: صفحة اللبنات البرمجية الأولية


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
when discover_button_en .Click
do
when discover_button_en .GotFocus
do
when discover_button_en .LongClick
do
when discover_button_en .LostFocus
1
when discover_button_en .Click
do

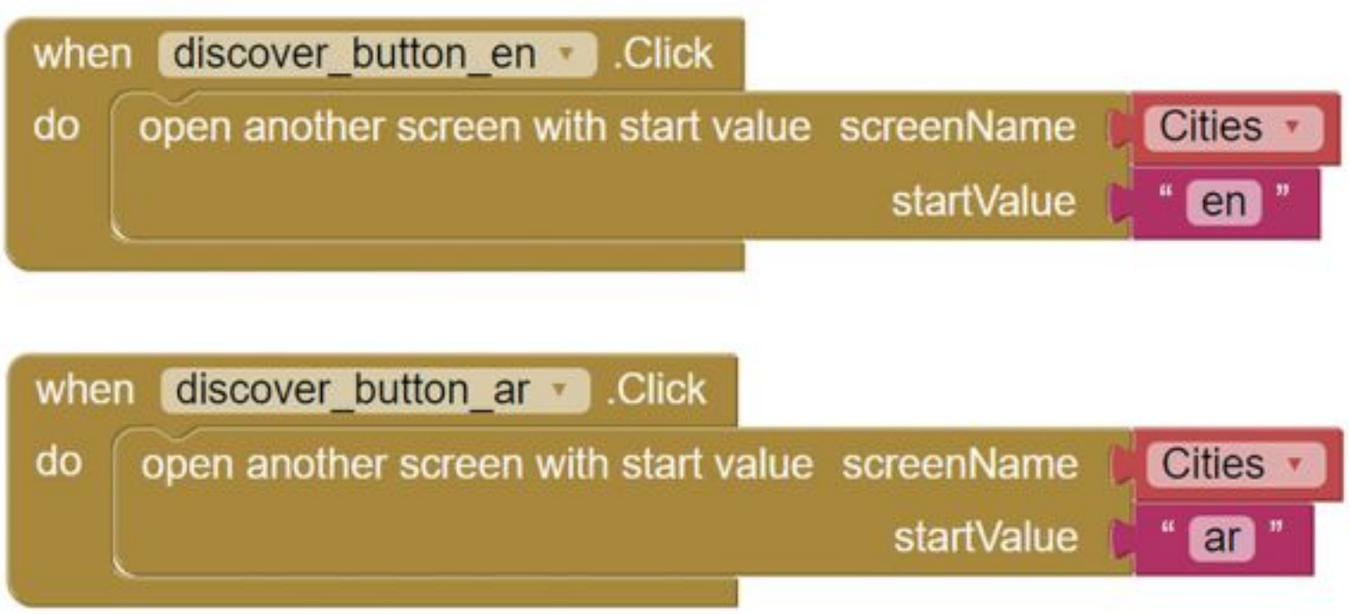

MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
Visit_SA_App
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
discover_button_en
discover_button_ar
Any component
then
else
do
result
evaluate but ignore result
open another screen screenName
Cities
open another screen with start value screenName
Cities
startValue
get start value
2
when discover_button_en .Click
do
open another screen with start value screenName
Cities
startValue



شكل 3.35: برمجة أزرار الشاشة الرئيسة

البرنامج بأكمله للشاشة الأولى (شاشة 1)

The Complete Code for the First Screen (Screen1)



شكل 3.36: البرنامج بأكمله للشاشة الأولى Screen1 (شاشة 1)

برمجة شاشة المدن Programming the Cities Screen

كما ذكرنا سابقًا، فإن الشاشة الرئيسة ستنقل المُستخدِم إلى الصفحة الخاصة بالمدن، كما ستحدد اللغة التي سيتم استخدامها على الشاشات التالية أيضًا.

إنشاء المحتوى الخاص بقائمة الخيارات Creating the Content for the ListPicker

يجب أن تملأ ListPicker Elements (عناصر قائمة الخيارات) بالنص المناسب حسب اللغة المُستخدَمة. الخطوة الأولى لذلك هي تحديد قوائم النقاط البارزة للوجهات المختلفة باللغتين الإنجليزية والعربية، والخطوة الثانية هي تهيئة مُكوِّنات ListPicker (قائمة الخيارات) باللغة المناسبة.

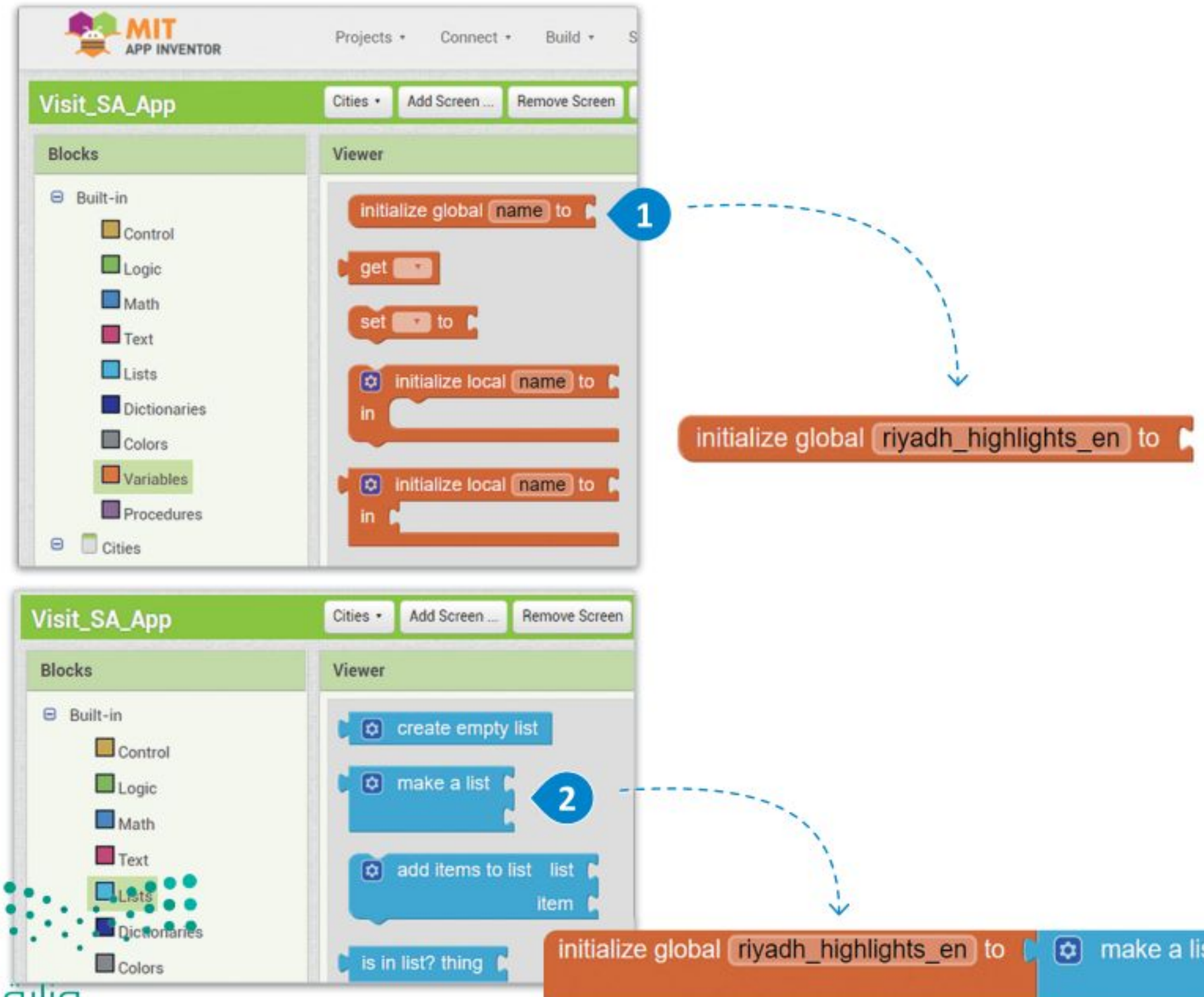
لإنشاء قوائم المحتوى:

> أنشئ متغيّرًا جديدًا يدعى riyadh_highlights_en (معالم_الرياض_بالإنجليزية). 1

> أنشئ اللبنة البرمجية الخاصة بالقائمة وضعها في المتغير riyadh_highlights_en (معالم_الرياض_بالإنجليزية). 2

> املأ القائمة بأسماء الأماكن باللغة الإنجليزية. 3

> كرِّر العملية للمتغير الخاص باللغة العربية riyadh_highlights_ar (معالم_الرياض_بالعربية). 4



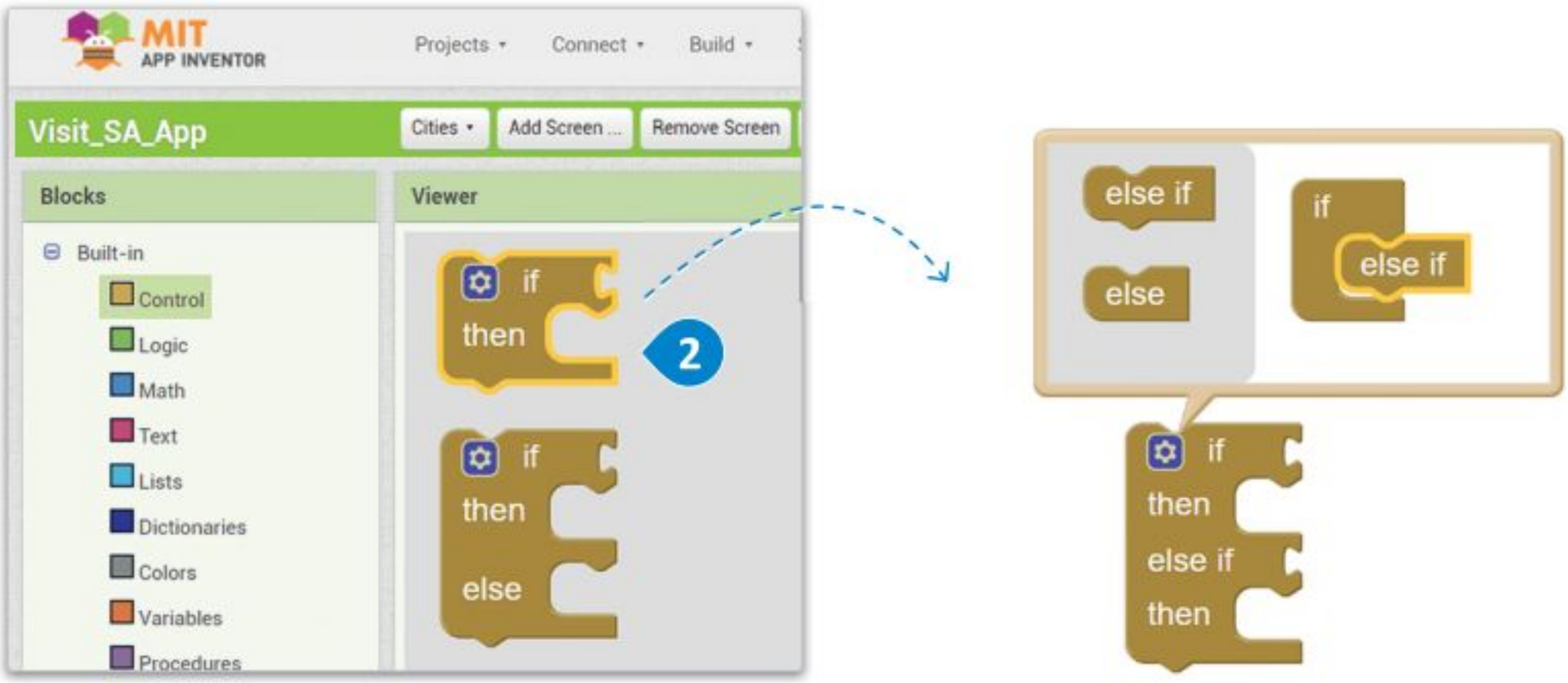
شكل 3.37: إنشاء قوائم المحتوى

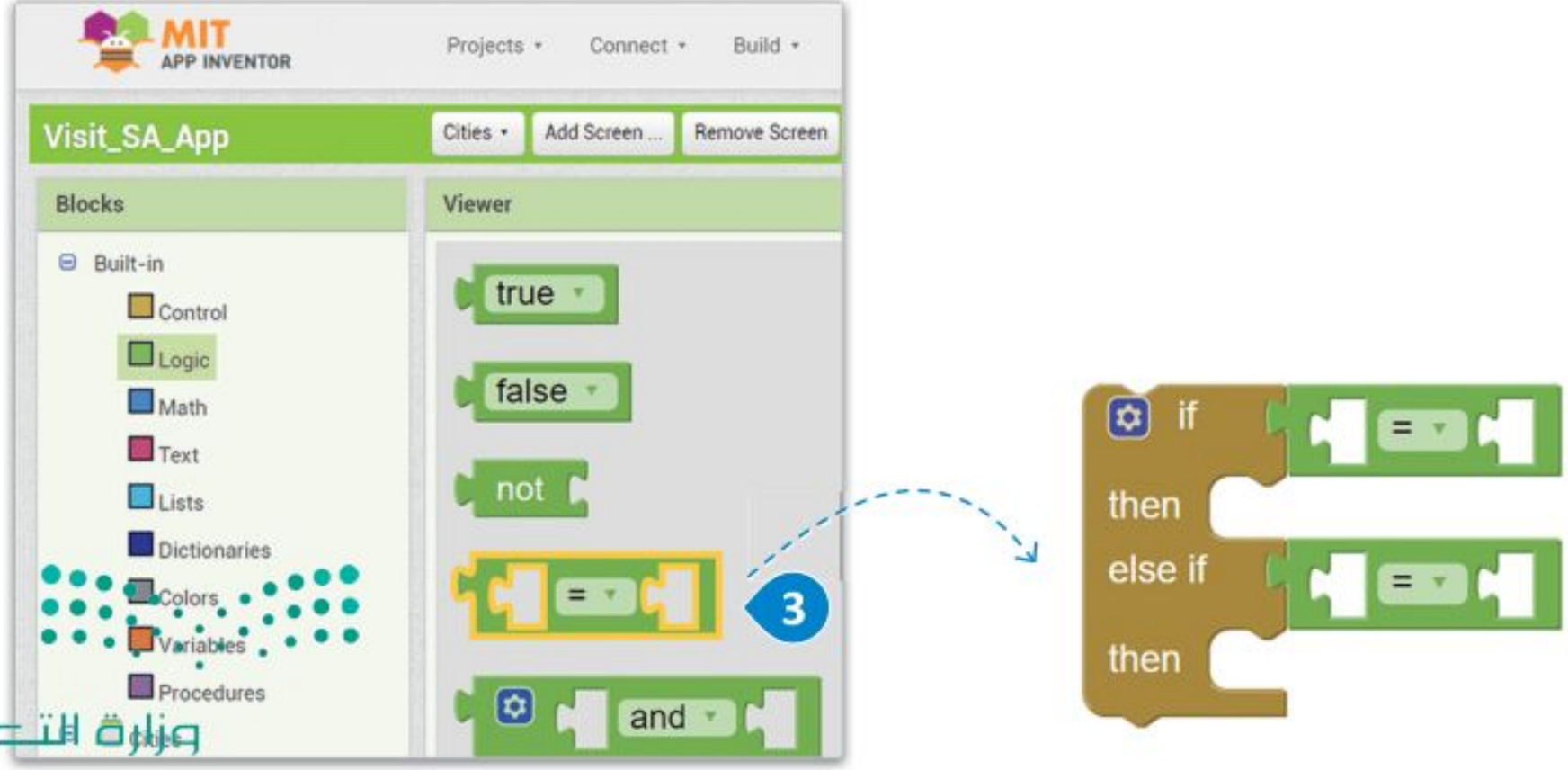
يتم استخدام عبارة if-then لبرمجة عنصر القائمة بحيث يفتح الشاشة المرتبطة به عند الضغط عليه.
إذا ضغط المُستخدِم على Al Masmak (المصمك)، فستفتح الصفحة المتعلقة به.

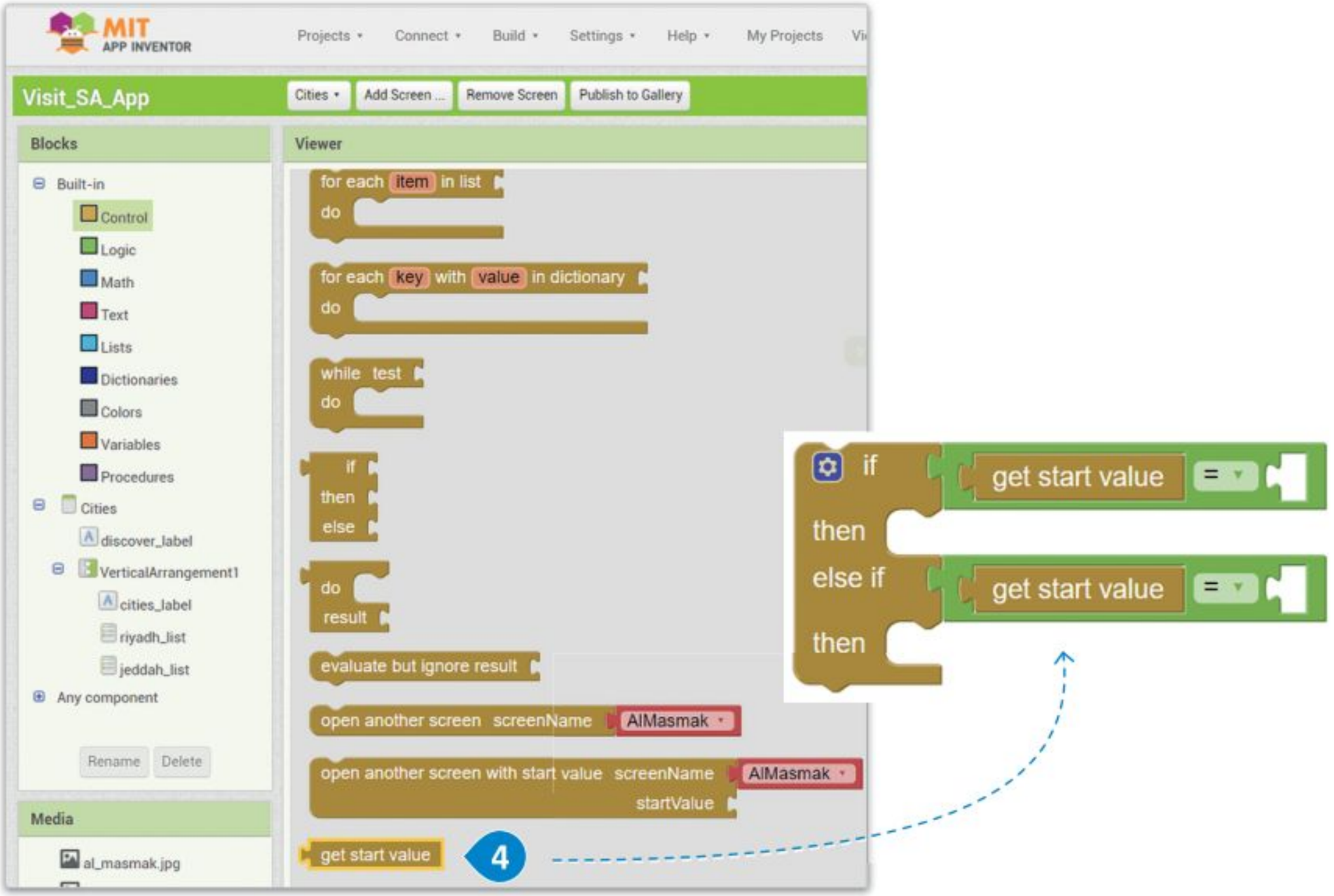
لتحديد محتوى القائمة:

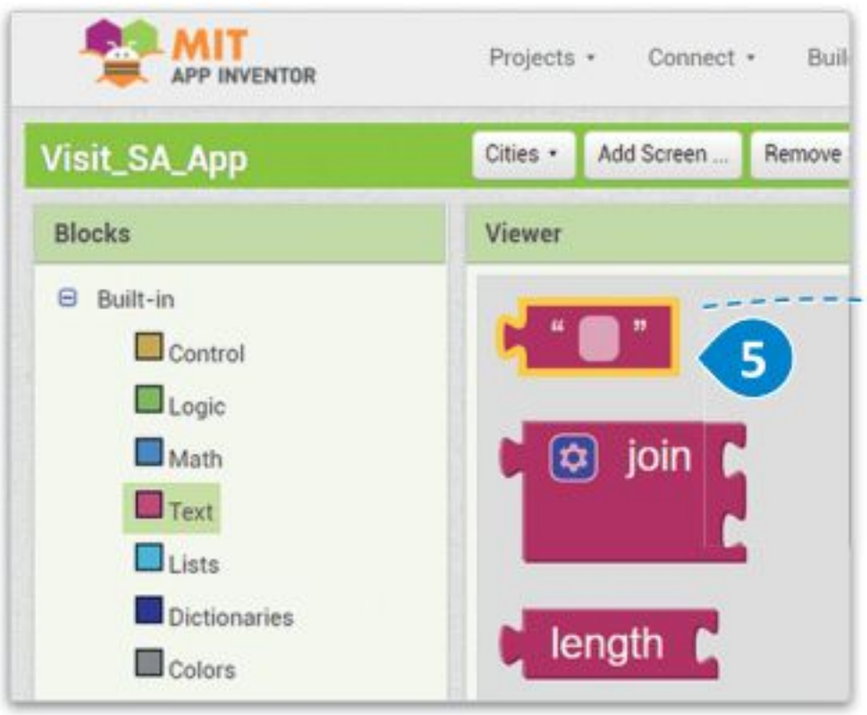
> حدِّد اللبنة BeforePicking (قَبْل الاختيار) لمُكوِّن riyadh_list (قائمة_الرياض). 1
> أنشئ عبارة if else if الشرطية. 2
> أضف عبارة equals (يساوي) لعبارة if وعبارة أخرى لعبارة else if. 3
> أضف متغير get start value (الحصول على قيمة البداية) إلى الجانب الأيسر من عبارة equals (يساوي). 4
> أضف en (الإنجليزية) و ar (العربية) إلى الجانب الأيمن من عبارة equals (يساوي). 5
> اختر أمر Set Elements (ضبط العناصر) من قائمة Riyadh_list (قائمة_الرياض). 6
> أضف متغيرات القائمة المناسبة للأمر أعلاه. 7
> أضف لبنة التعليمات البرمجية if else if إلى حدث BeforePicking (قَبْل الاختيار). 8

MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
when riyadh_list .AfterPicking
do
when riyadh_list .BeforePicking
do
when riyadh_list .GotFocus
do
when riyadh_list .LostFocus
do
1
when riyadh_list .BeforePicking
do

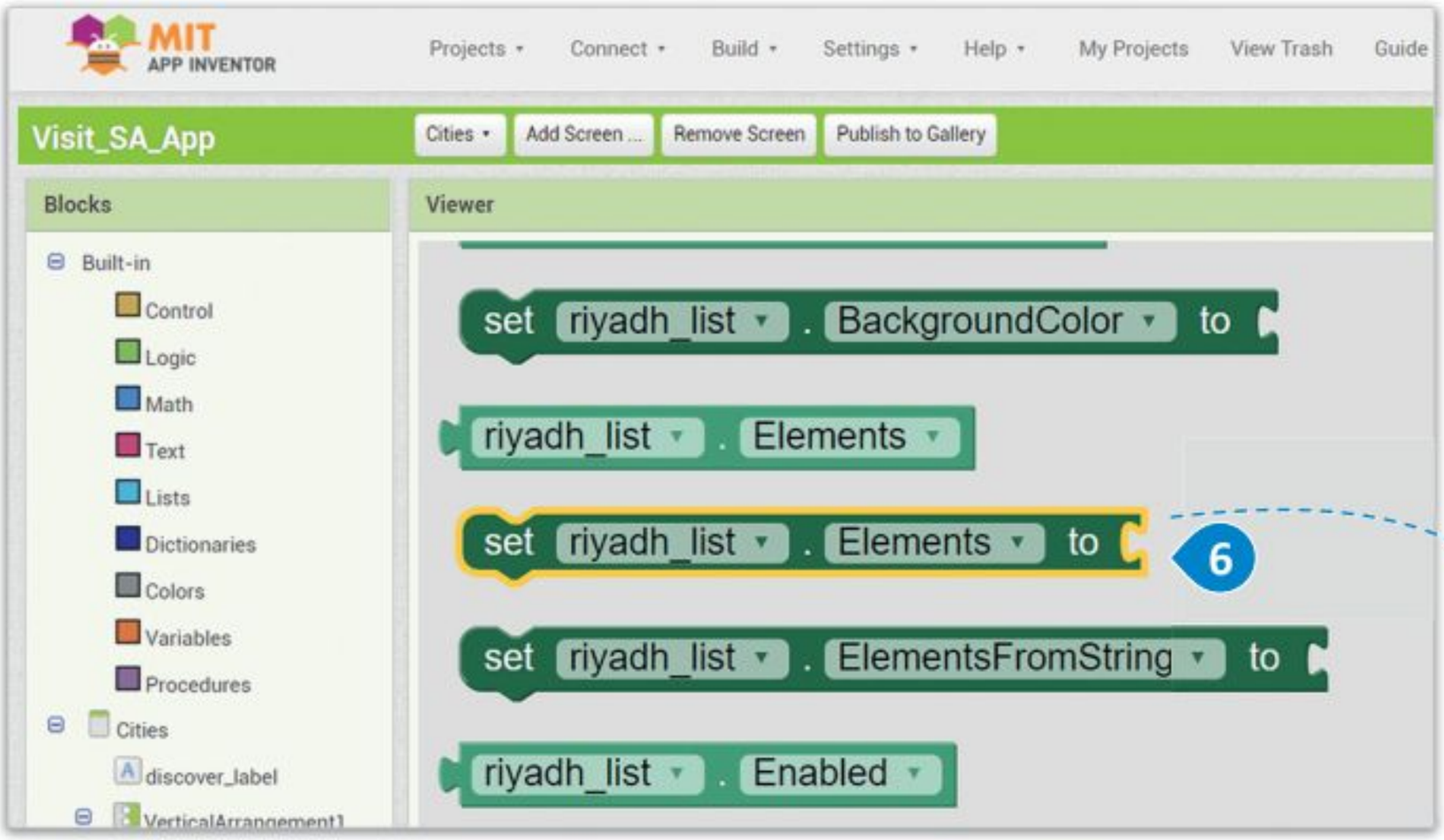

MIT APP INVENTOR
Projects
Connect
Build
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
if
then
if
then
else
2
else if
else
if
else if
if
then
else if
then

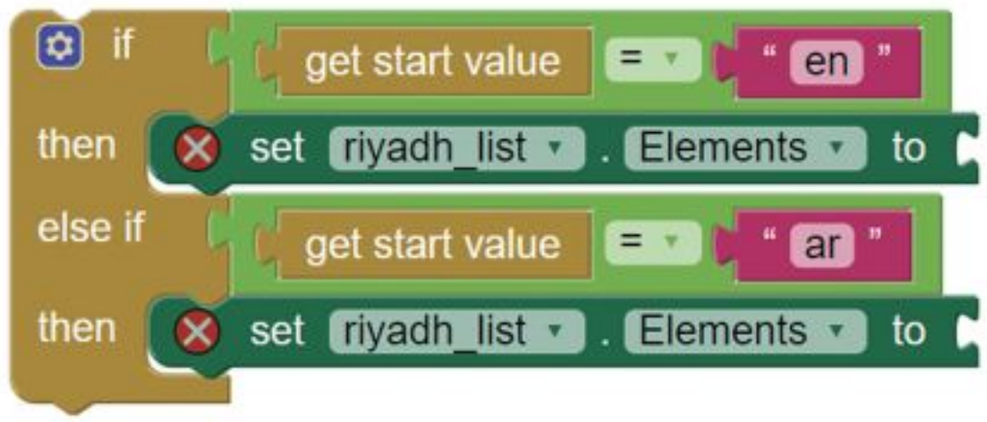

MIT APP INVENTOR
Projects
Connect
Build
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
true
false
not
=
3
and
if
then
else if
then
=
=

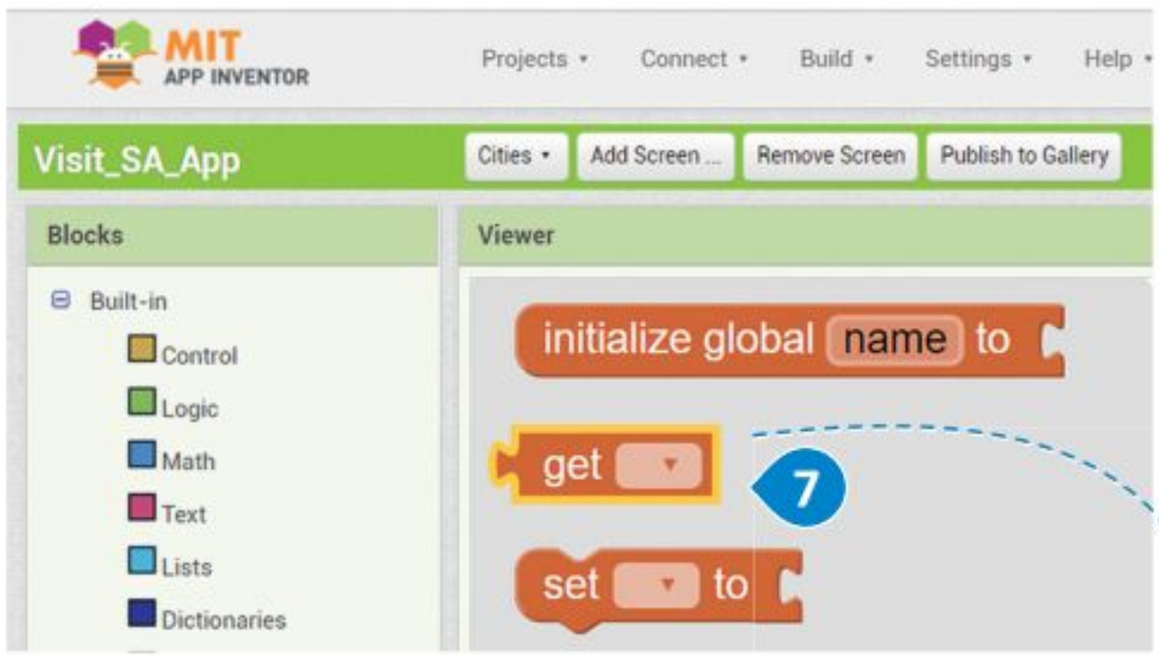

MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
discover_label
VerticalArrangement1
cities_label
riyadh_list
jeddah_list
Any component
Rename
Delete
Media
al_masmak.jpg
for each item in list
do
for each key with value in dictionary
do
while test
do
if
then
else
do
result
evaluate but ignore result
open another screen screenName
AlMasmak
open another screen with start value screenName
AlMasmak
startValue
get start value
4
if
get start value
=
then
else if
get start value
=
then

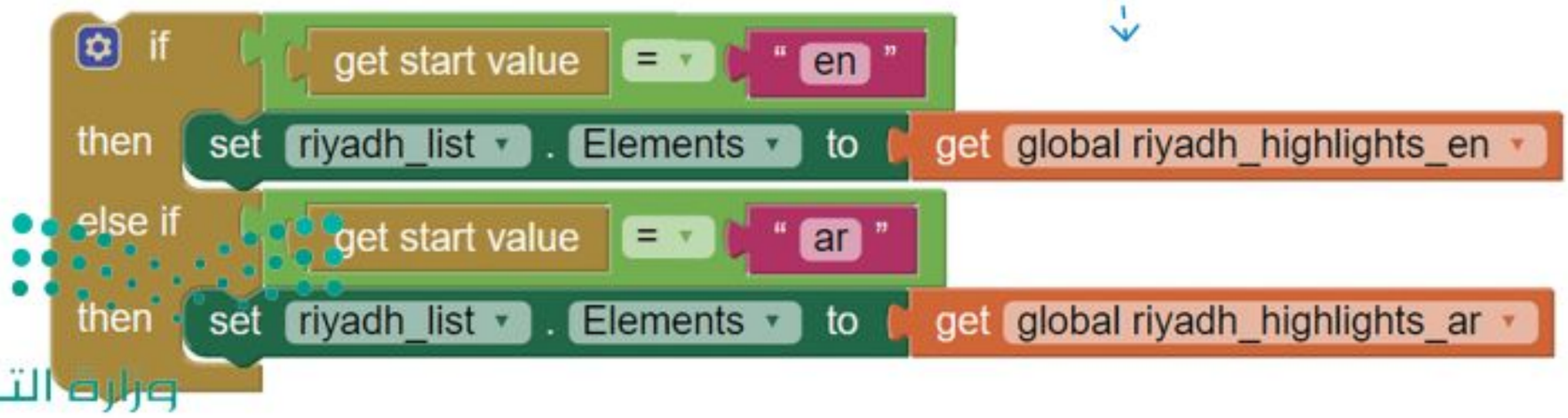

MIT APP INVENTOR
Projects
Connect
Visit_SA_App
Cities
Add Screen ...
Remove
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
" "
5
join
length

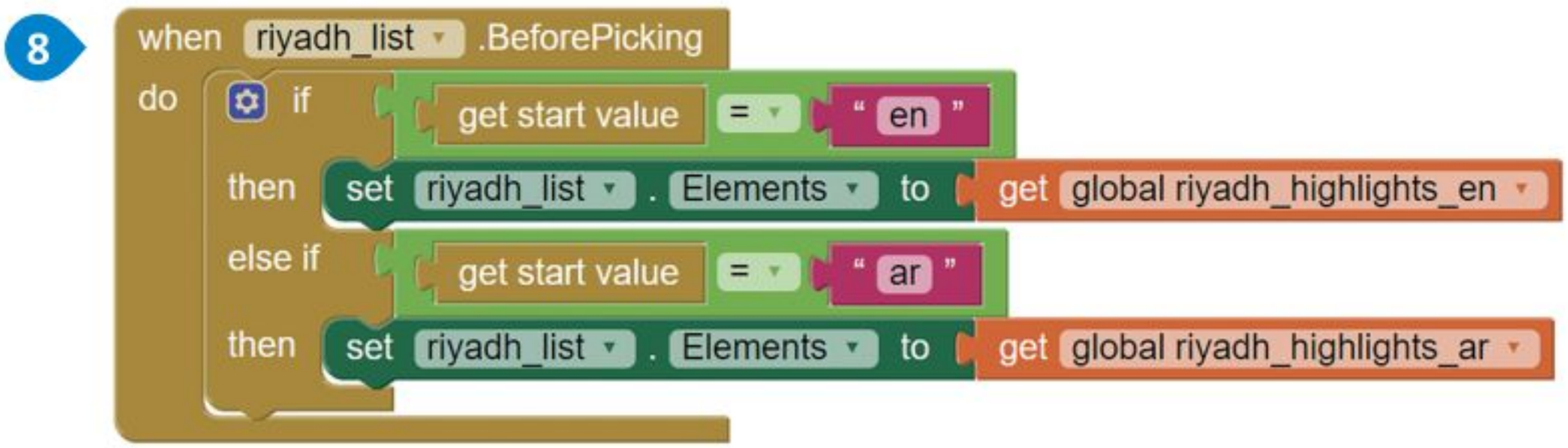
if
get start value
=
" en "
then
else if
get start value
=
" ar "
then


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
discover_label
VerticalArrangement1
set riyadh_list . BackgroundColor to
riyadh_list . Elements
set riyadh_list . Elements to
6
set riyadh_list . ElementsFromString to
riyadh_list . Enabled


if
get start value
=
" en "
then
set riyadh_list . Elements to
else if
get start value
=
" ar "
then
set riyadh_list . Elements to


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
initialize global name to
get
7
set to


if
get start value
=
" en "
then
set riyadh_list . Elements to
get global riyadh_highlights_en
else if
get start value
=
" ar "
then
set riyadh_list . Elements to
get global riyadh_highlights_ar



شكل 3.38: تهيئة محتوى ListPicker (قائمة الخيارات)

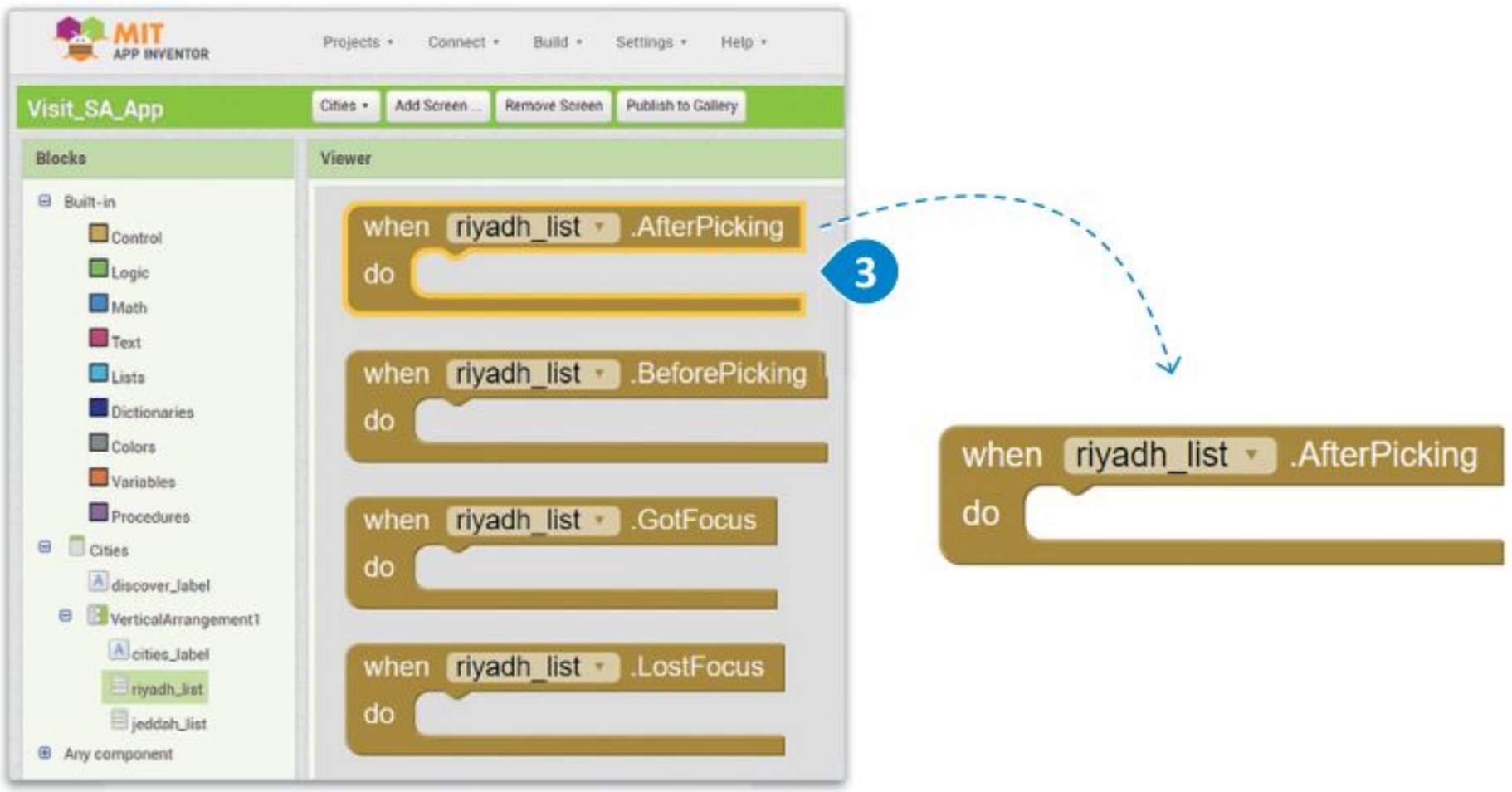
فتح صفحة جديدة من قائمة الخيارات Opening a New Page from the ListPicker

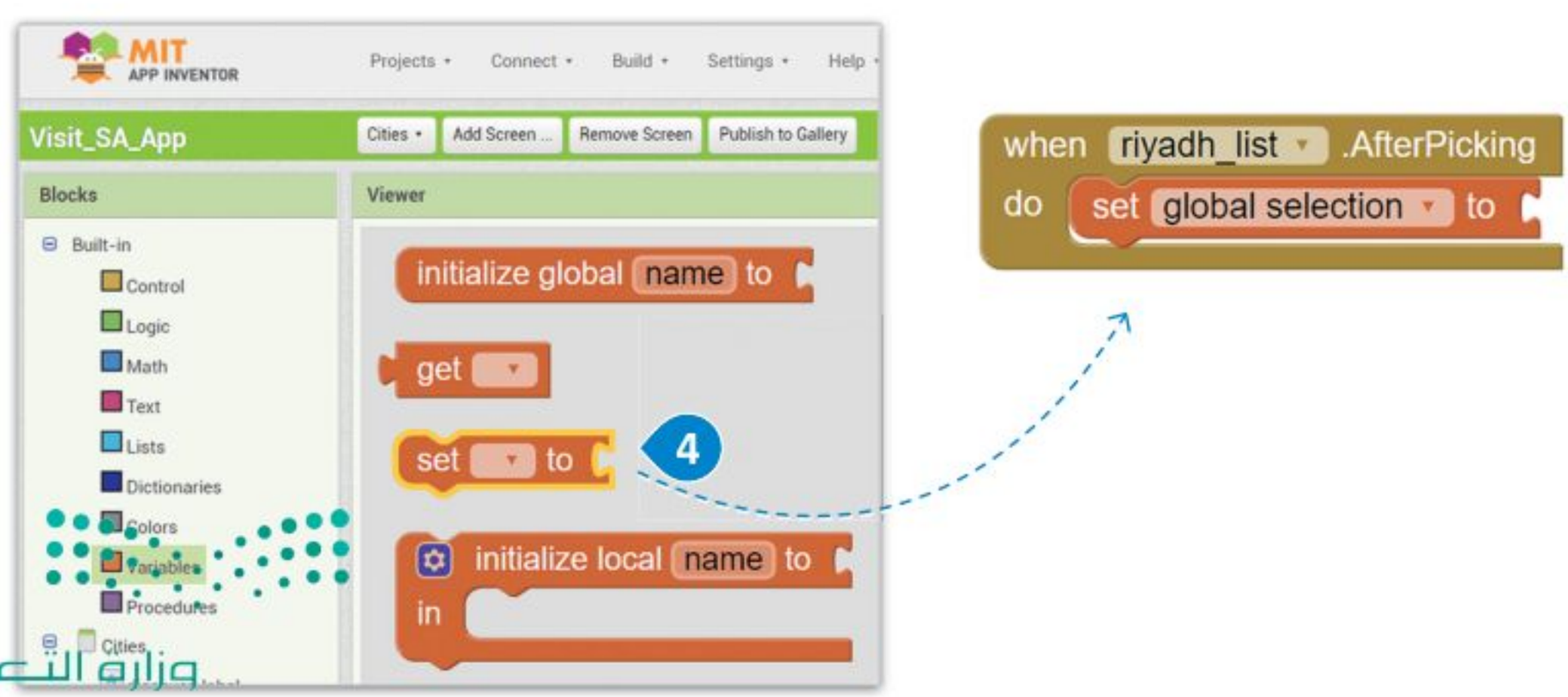
بعد أن أصبح لديك قائمة من المعالم البارزة للمدينة التي تختارها، سيتعيّن عليك أن تجعل لكلّ مَعْلَم صفحةً خاصّة تعرض فيها معلومات عنه، وسيحدد كل اختيار من ListPicker (قائمة الخيارات) المَعْلَم المطلوب لفتحه، بحيث تدعم الصفحة الخاصة بكل من المعالم اللغتين الإنجليزية والعربيّة.

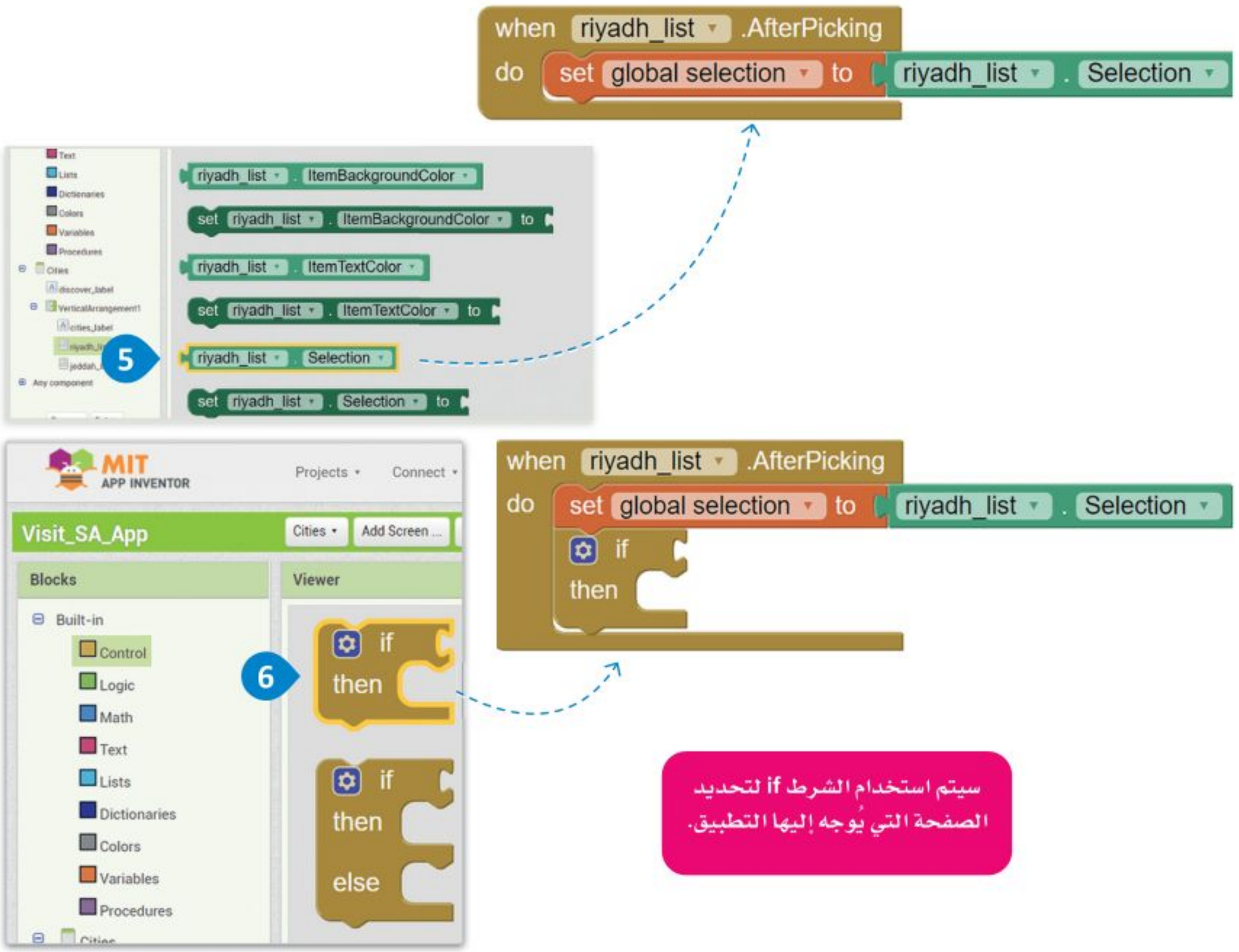
للحصول على اسم الصفحة الجديدة للمَعْلَم من التحديد ListPicker (قائمة الخيارات):

> أنشئ variable (متغير) جديد اسمه selection (اختيار). 1
> أضف empty text block (لبنة نص فارغة) إلى المتغير selection (اختيار). 2
> حدِّد الحدث AfterPicking (بَعْد الاختيار) لمُكوِّن riyadh_list (قائمة_الرياض). 3
> أضف المتغير selection (اختيار) داخل الحدث AfterPicking (بَعْد الاختيار). 4
> اضبط المتغير selection (اختيار) إلى خاصية Selection (اختيار) لـ riyadh_list (قائمة_الرياض). 5
> أضف لبنة التعليمات البرمجية if داخل الحدث AfterPicking (بَعْد الاختيار). 6

MIT APP INVENTOR
Projects
Connect
Visit_SA_App
Cities
Add Screen ...
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
discover_label
VerticalArrangement1
cities_label
riyadh_list
jeddah_list
Any component
2
join
length
initialize global selection to
when riyadh_list .AfterPicking
do


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
discover_label
VerticalArrangement1
cities_label
riyadh_list
jeddah_list
Any component
when riyadh_list .AfterPicking
do
3
when riyadh_list .BeforePicking
do
when riyadh_list .GotFocus
do
when riyadh_list .LostFocus
do
when riyadh_list .AfterPicking
do


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
initialize global name to
get
set to
4
initialize local name to
in
when riyadh_list .AfterPicking
do
set global selection to



شكل 3.39: تحديد الصفحة من ListPicker (قائمة الخيارات)

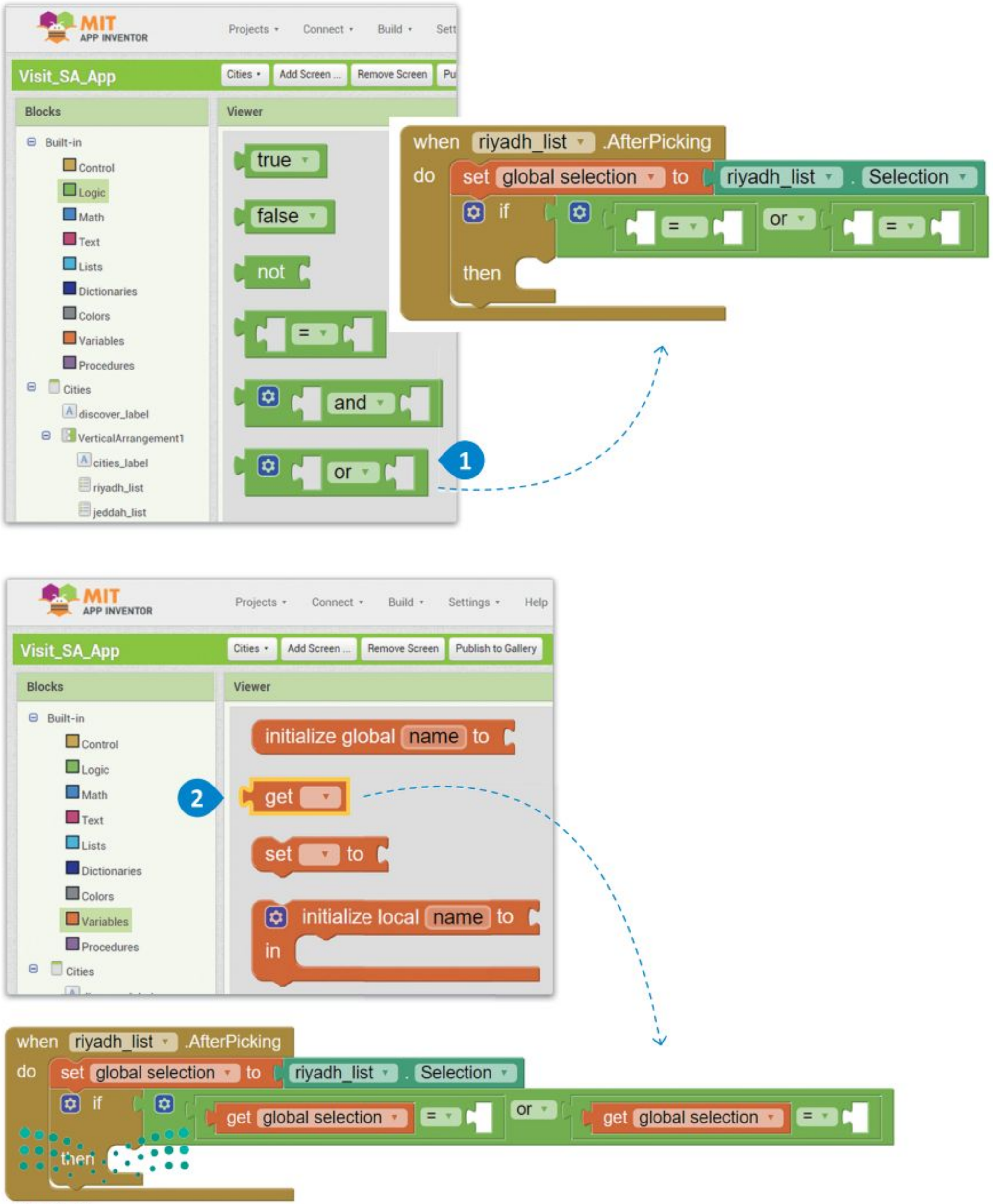
فتح الصفحة المناسبة لاختيار قائمة الخيارات

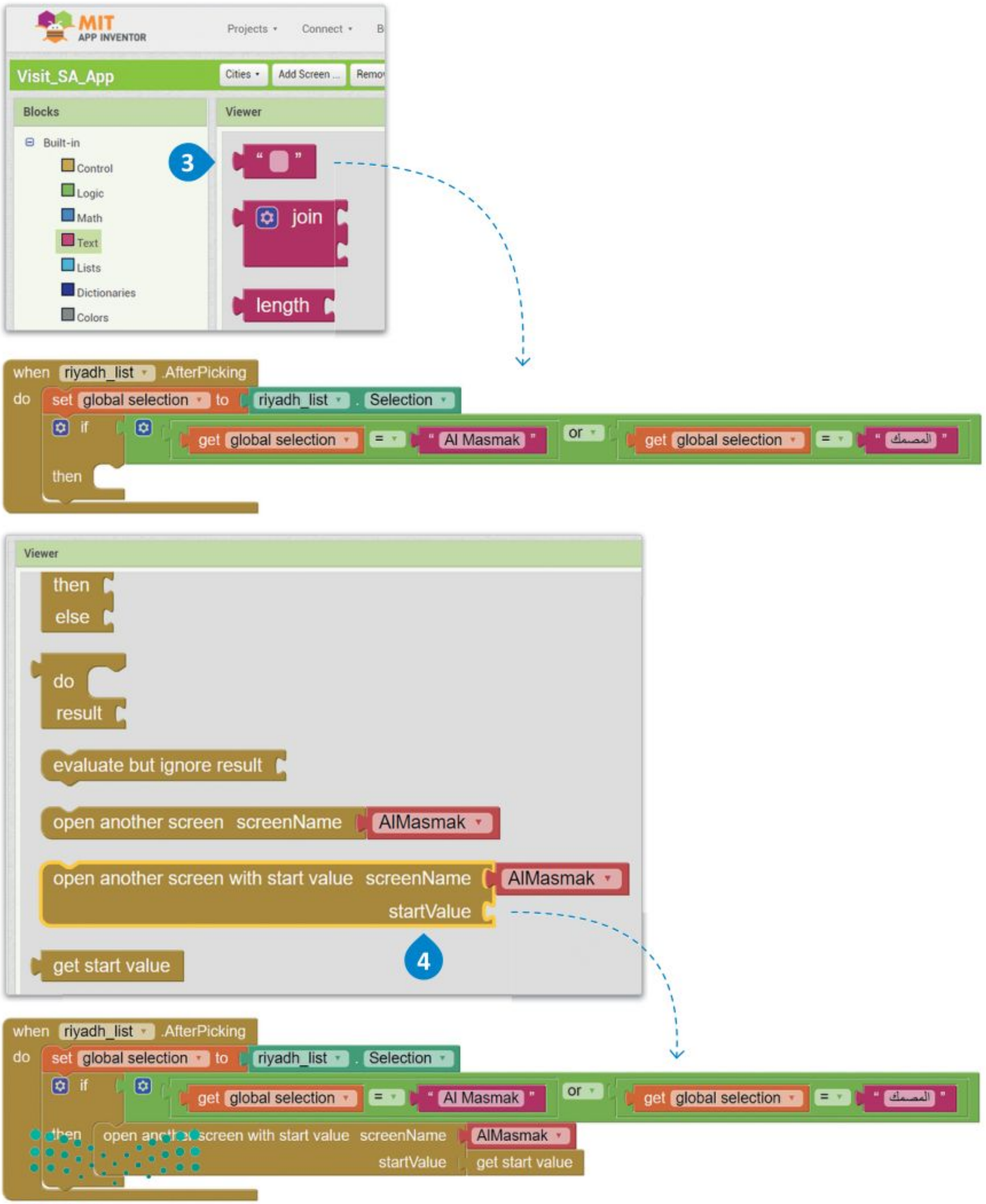
Opening the Appropriate Page for the ListPicker Selection

سيتمّ عرض الصفحة الخاصة بالمَعْلَم بناء على الاختيار من ListPicker (قائمة الخيارات)، حيث تتعرف لبنة التعليمات البرمجية على المَعْلَم المُراد عرض صفحته سواء كانت باللغة الإنجليزية أو باللغة العربية، وترسل خيار اللغة إلى الشاشة أيضًا.

لفتح الصفحة المناسبة من التحديد ListPicker (قائمة الخيارات):

- أضف شرط or الذي يحتوي على شرطَي equals (يساوي)، وضعهما في عبارة If داخل حدث AfterPicking (بَعْد الاختيار). 1
- أضف المتغير selection (اختيار) إلى الجانب الأيسر من شرطَي equals (يساوي). 2
- أضف النصوص المتعلقة بمَعْلَم Al Masmak (المصمك) باللغة الإنجليزية والعربية إلى الجانب الأيمن من شرطَي equals (يساوي). 3
- افتح شاشة Al Masmak (المصمك) مع تحديد StartValue (قيمة البداية) التي استخدمتها في الشاشة السابقة. 4

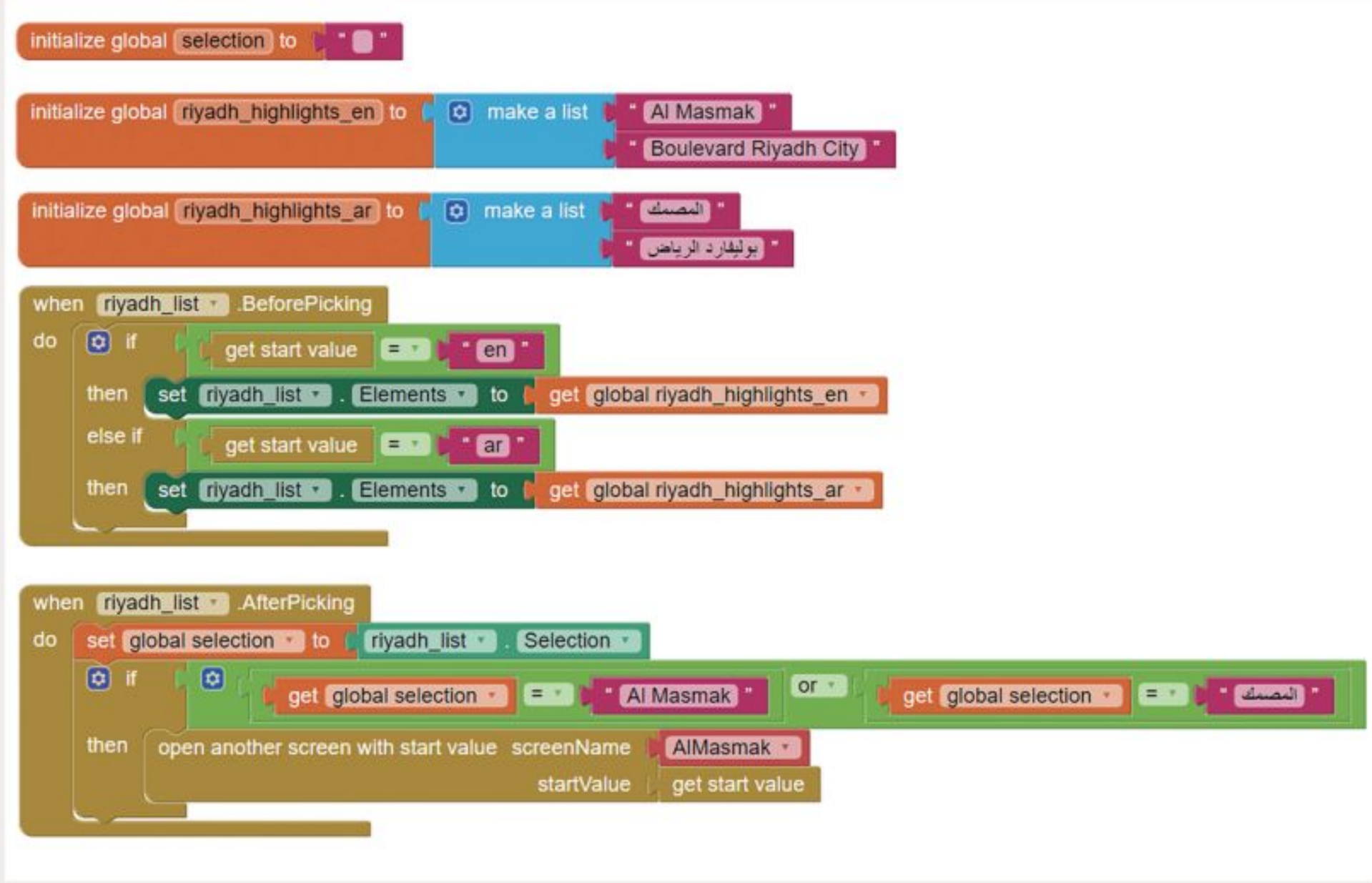

MIT APP INVENTOR
Projects
Connect
Build
Sett
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
discover_label
VerticalArrangement1
cities_label
riyadh_list
jeddah_list
true
false
not
=
and
or
1
when riyadh_list .AfterPicking
do set global selection to riyadh_list . Selection
if
or
=
then
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
Cities
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Cities
2
initialize global name to
get
set to
initialize local name to
in
when riyadh_list .AfterPicking
do set global selection to riyadh_list . Selection
if
get global selection
=
or
get global selection
=
then



شكل 3.40: فتح الصفحة المناسبة من التحديد ListPicker (قائمة الخيارات)

البرنامج بأكمله للشاشة الثانية (المدن)

The Complete Code for the Second Screen (Cities)



شكل 3.41: البرنامج بأكمله للشاشة الثانية

برمجة شاشة مَعلَمْ (المصمك)

Programming the Highlight Screen (Al Masmak)

ستتغير لغة الشاشة الخاصة بالمَعْلَم وكذلك تنسيق النص اعتمادًا على اختيار اللغة من الصفحة الرئيسة، وستقدم هذ الشاشة أيضًا خيار عرض خريطة تفاعلية لموقع المَعْلَم.

تغيير اللغة ديناميكيًا لصفحة المَعْلَم

Dynamically Changing the Language for the Highlight Page

سيتم توجيه المُستخدِم إلى الصفحة الخاصة بالمَعْلَم، والتي تحتوي على نص حول المَعْلَم وكذلك على صورة للمَعْلَم المحدد، وستتغير لغة النص وفقًا للغة المحددة. ستتم محاذاة النص بشكل صحيح بناءً على اللغة التي تم اختيارها في الصفحة الرئيسة، حيث يتم محاذاة النص إلى اليسار إذا كان اختيار اللغة هو الإنجليزية، ومحاذاة النص إلى اليمين إذا كان اختيار اللغة هو العربية.

لتغيير اللغة ديناميكيًا:

> أنشئ المتغيرات المناسبة لتسمية النصوص. 1

> أضف نصًا لتسميتي title (العنوان)، و description (الوصف) الخاصين بمَعْلَم Al Masmak (المصمك) بكل من اللغتين الإنجليزية والعربية. 2

> حدِّد الحدث initialize (التهيئة) لشاشة Al Masmak. 3

> أضف لبنة التعليمات البرمجية if else if داخل حدث initialize (التهيئة). 4

> أضف شرط equals (يساوي) داخل كل عبارة if. 5

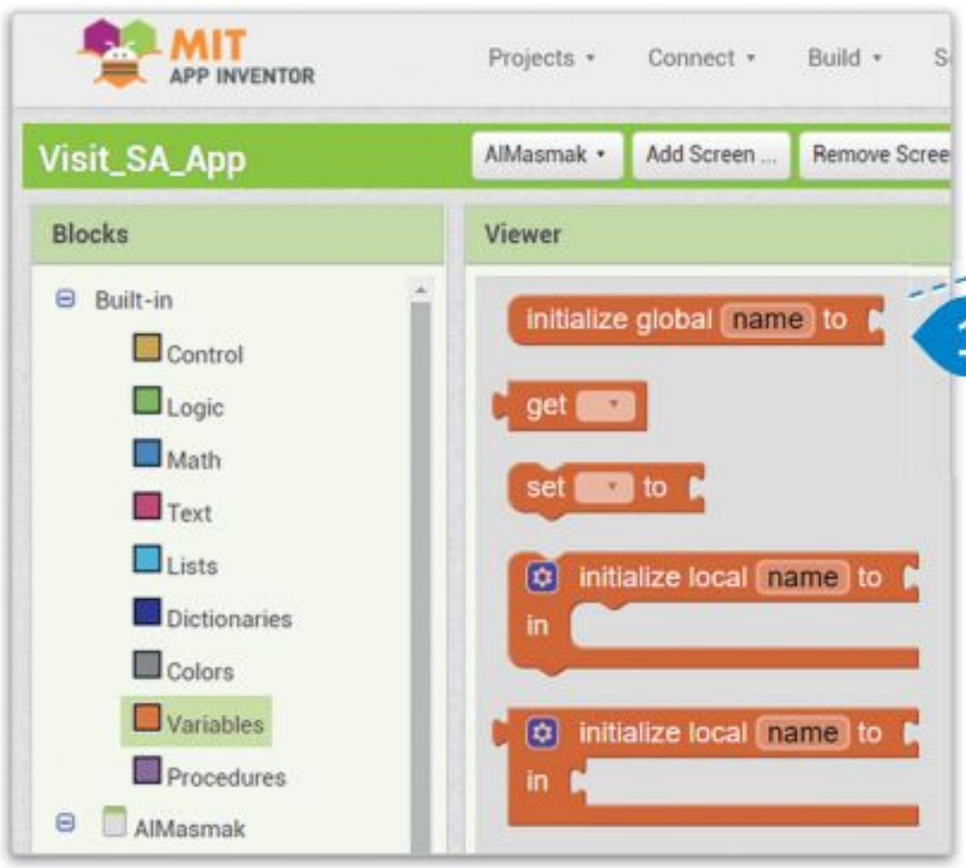
> أضف get start value (الحصول على قيمة البداية) إلى الجانب الأيسر من كل شرط equals (يساوي). 6

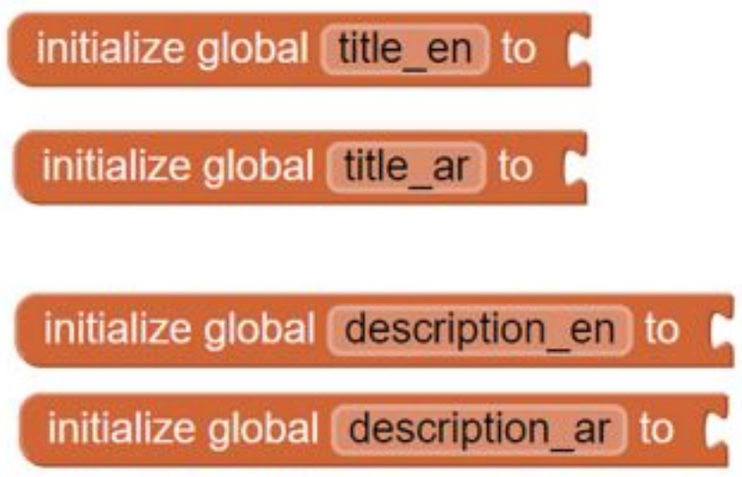
> أضف en(الإنجليزية) و ar(العربية) إلى الجانب الأيمن من شرط equals (يساوي). 7

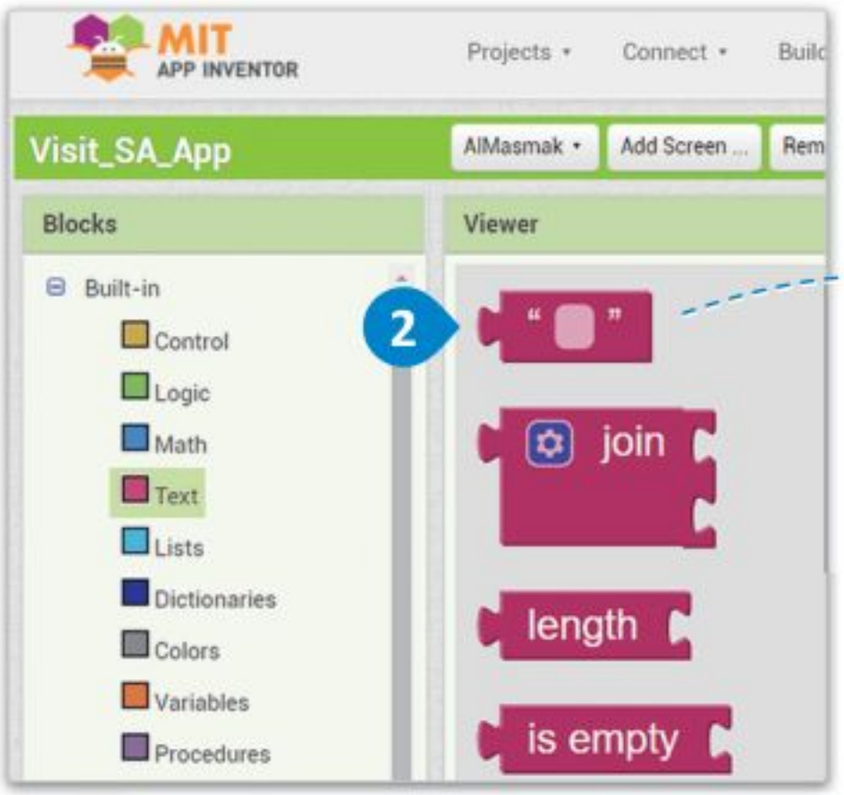
> اضبط خاصية Text (النص) الخاصة بـ title_label (عنوان_التسمية) إلى title variable (عنوان المتغير) الخاص باللغة المناسبة. 8

> اضبط خاصية Text (النص) الخاصة بـ description_Label (وصف_التسمية) إلى description variable (وصف المتغير) للغة المناسبة. 9

> اضبط خاصية Alignhorizontal (المحاذاة الأفقية) لأداة verticalRangement1 (الترتيب العمودي1) إلى اتجاه المحاذاة المناسب للغة. 10







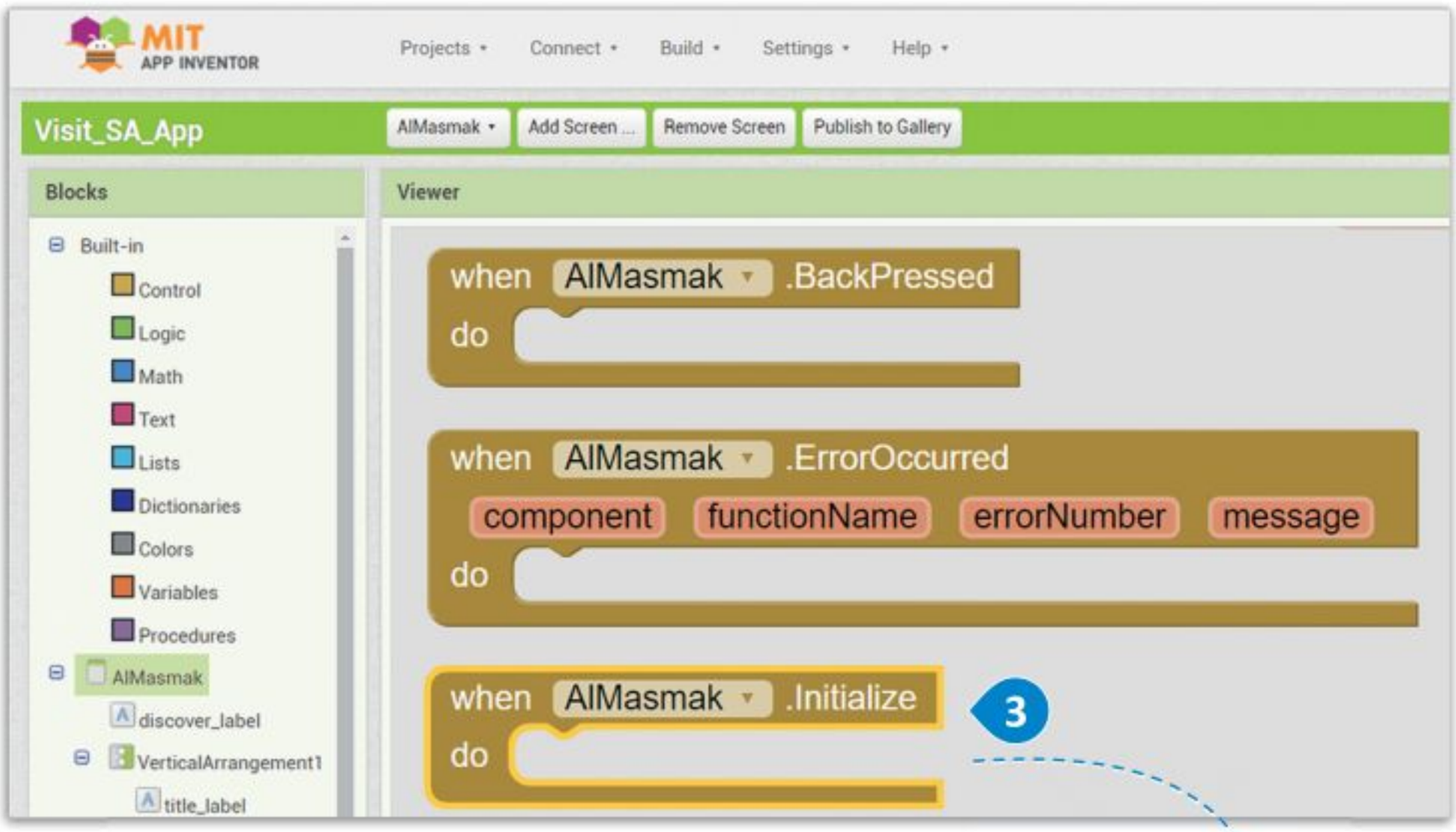
ابحث في الإنترنت عن وصف لقصر المصمك (Al Masmak) باللغتين الإنجليزية والعربية.

initialize global title_en to " Al Masmak "

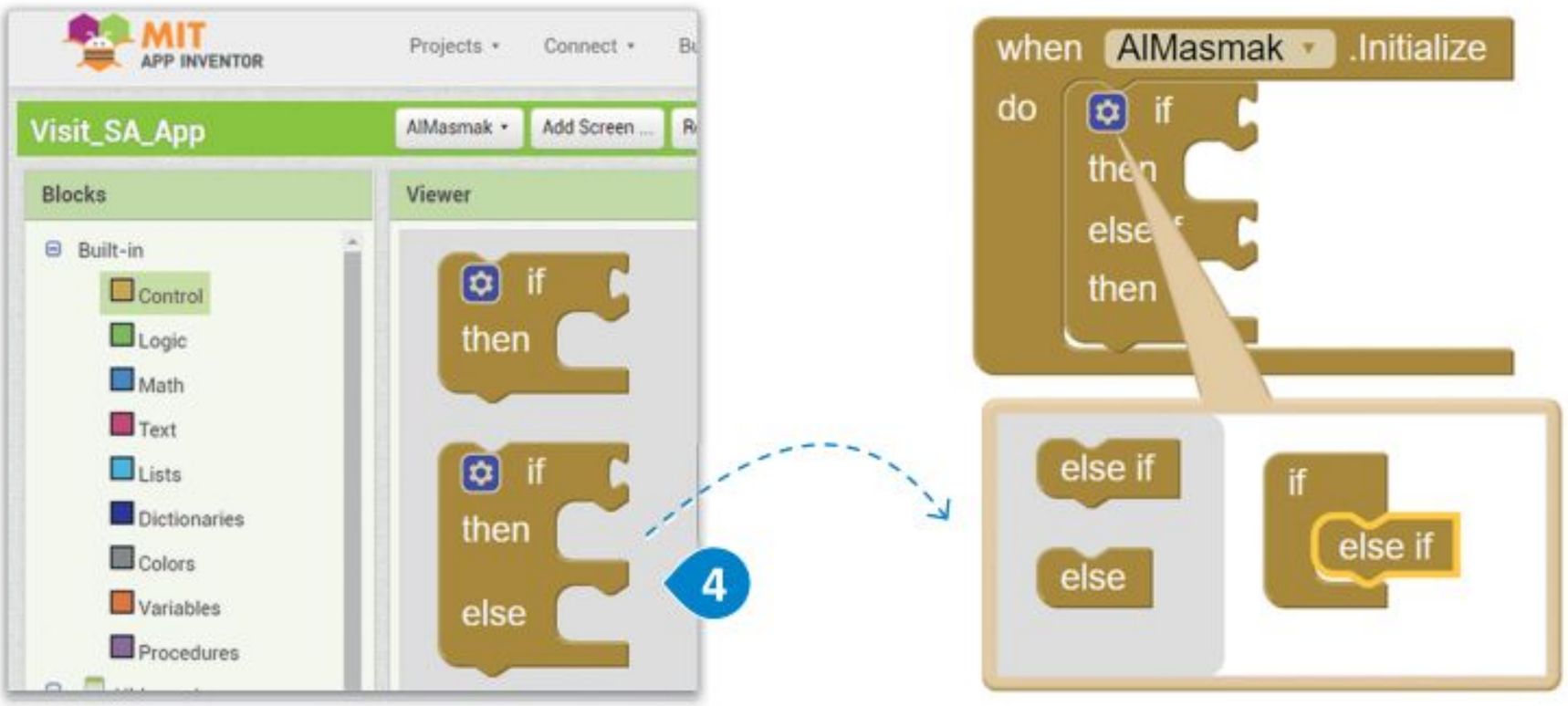
initialize global title_ar to " المصمك "

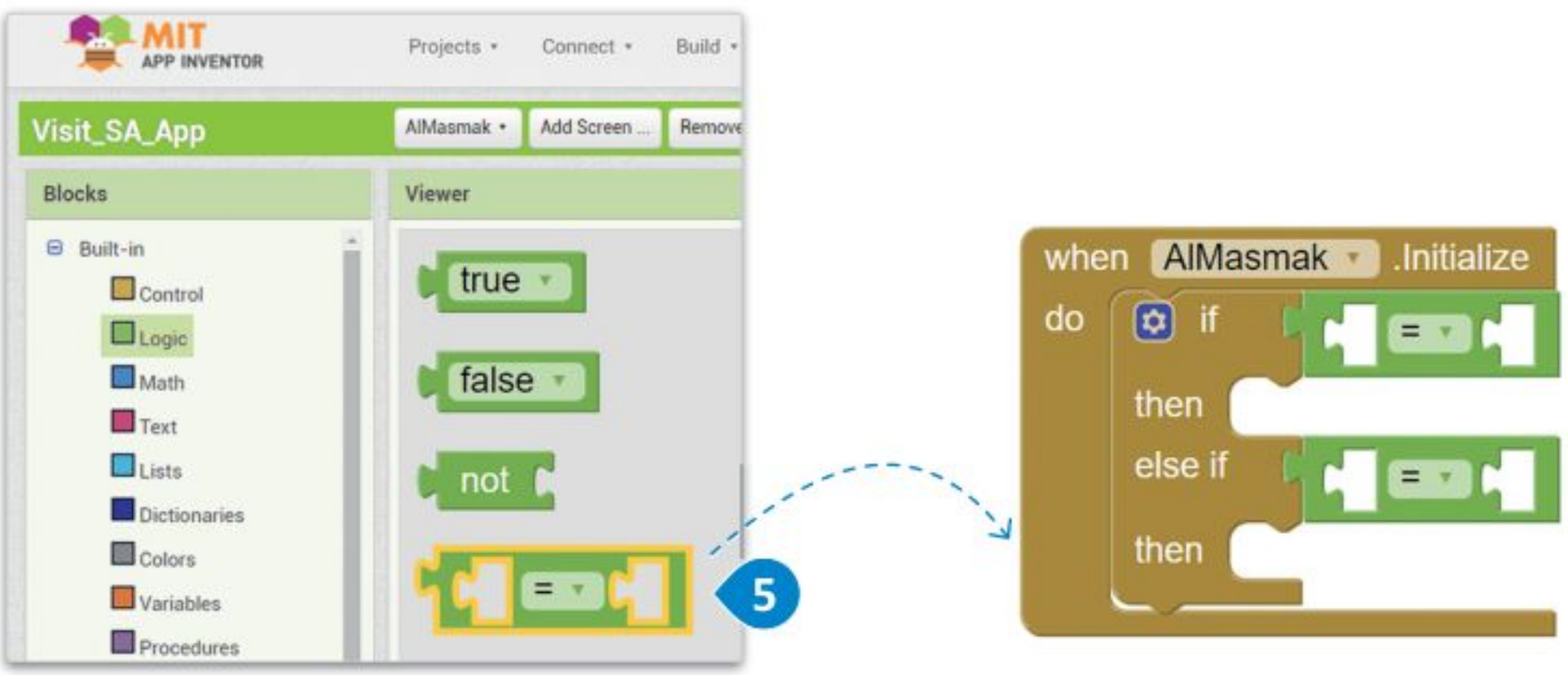
initialize global description_en to " Located in the heart of the old quarter in Riyad... "

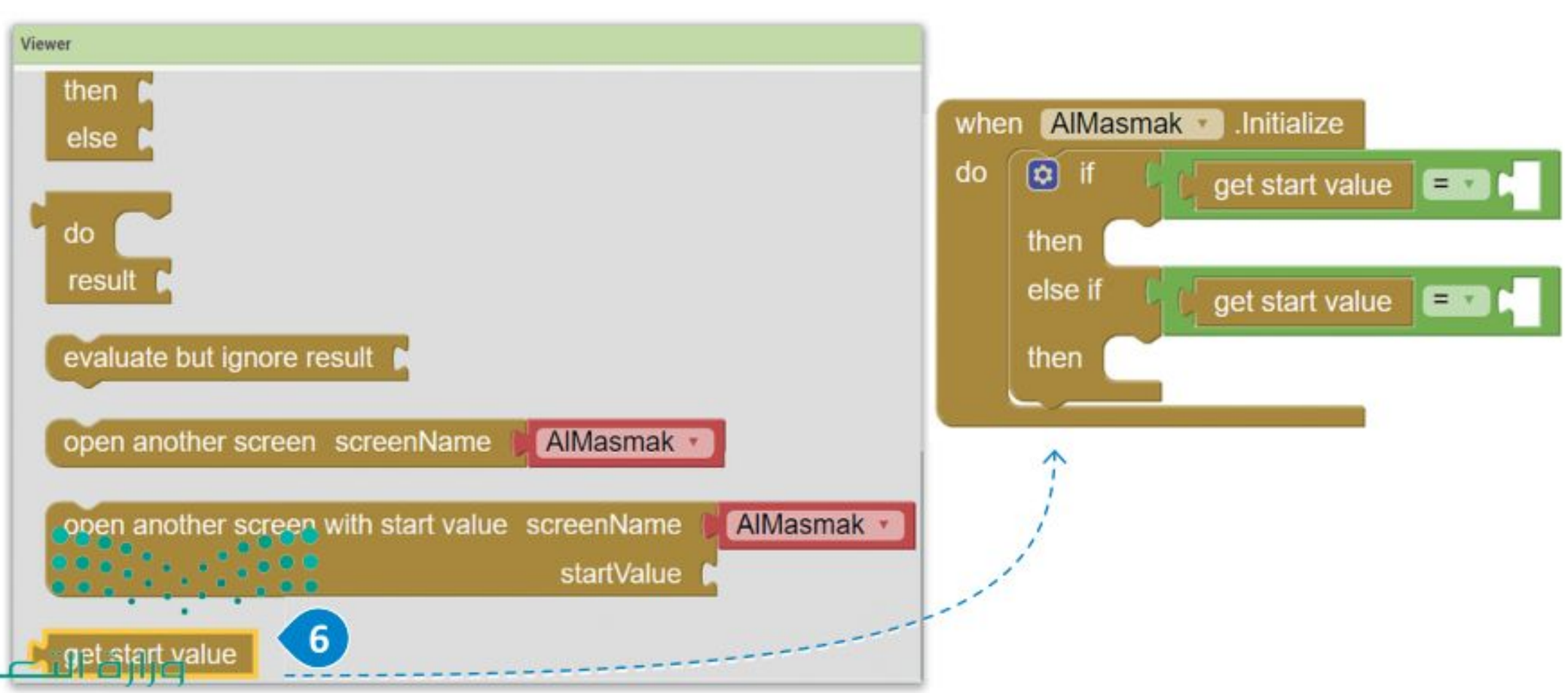
initialize global description_ar to " يقع قصر المصمك في وسط مدينة الرياض، وهو عبارة عن... "



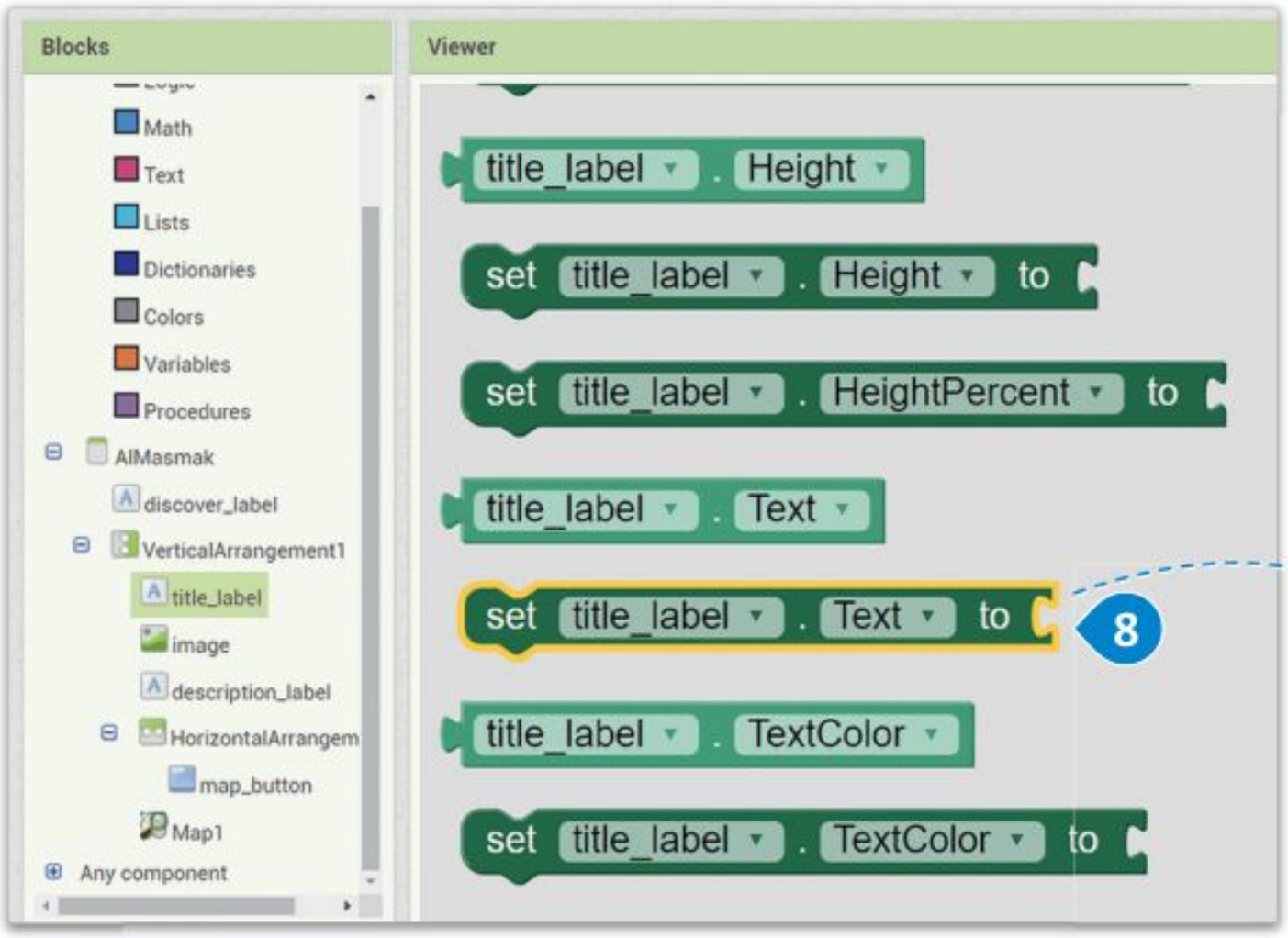
when AlMasmak .Initialize
do

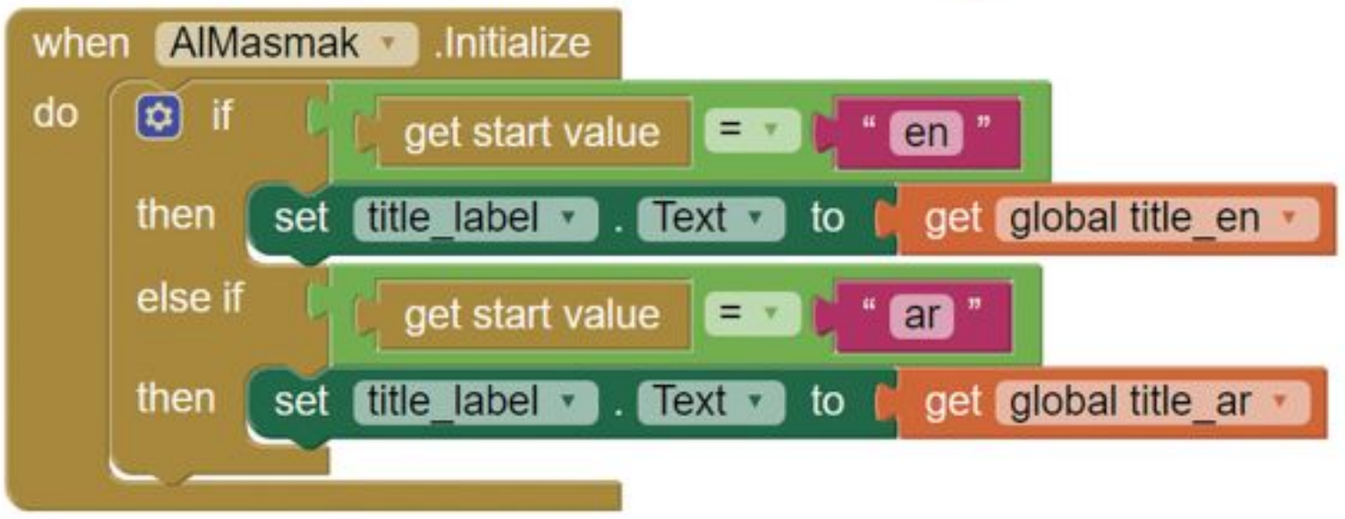

MIT APP INVENTOR
Projects
Connect
Visit_SA_App
AIMasmak
Add Screen ...
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
if
then
else
4
when AIMasmak .Initialize
do
if
then
else if
else
if
else if

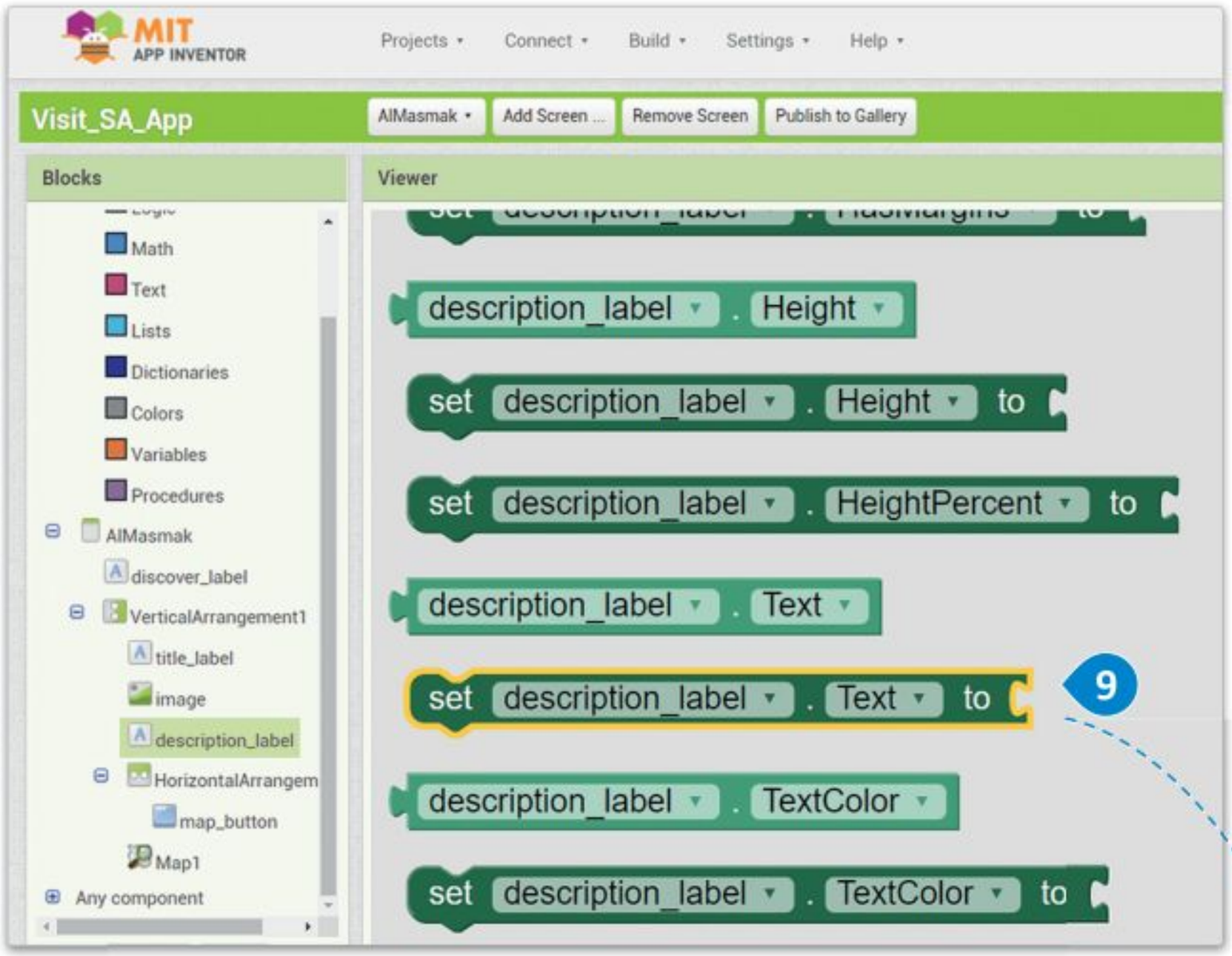

MIT APP INVENTOR
Projects
Connect
Build
Visit_SA_App
AIMasmak
Add Screen ...
Remove
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
true
false
not
=
5
when AIMasmak .Initialize
do
if
then
else if
then

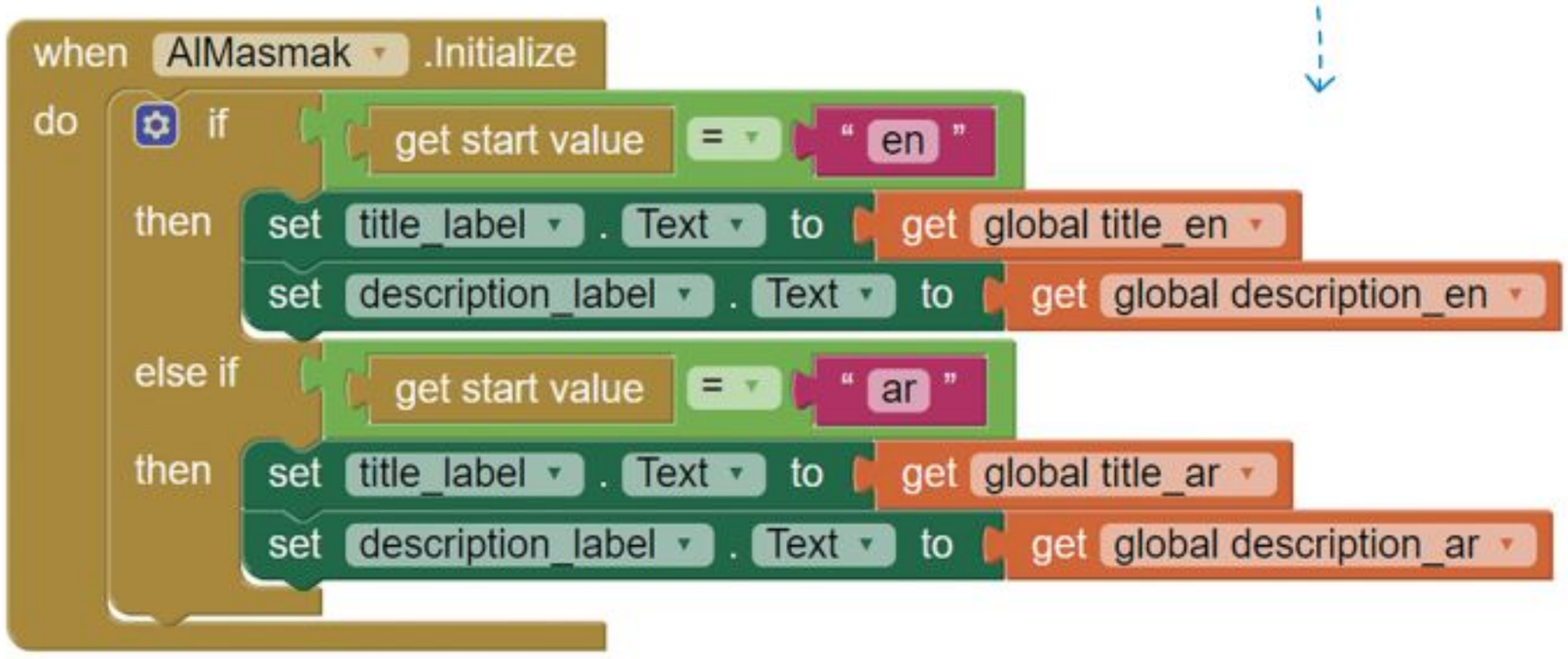

Viewer
then
else
do
result
evaluate but ignore result
open another screen screenName
AIMasmak
open another screen with start value screenName
AIMasmak
startValue
get start value
6
when AIMasmak .Initialize
do
if
get start value
=
then
else if
get start value
=
then

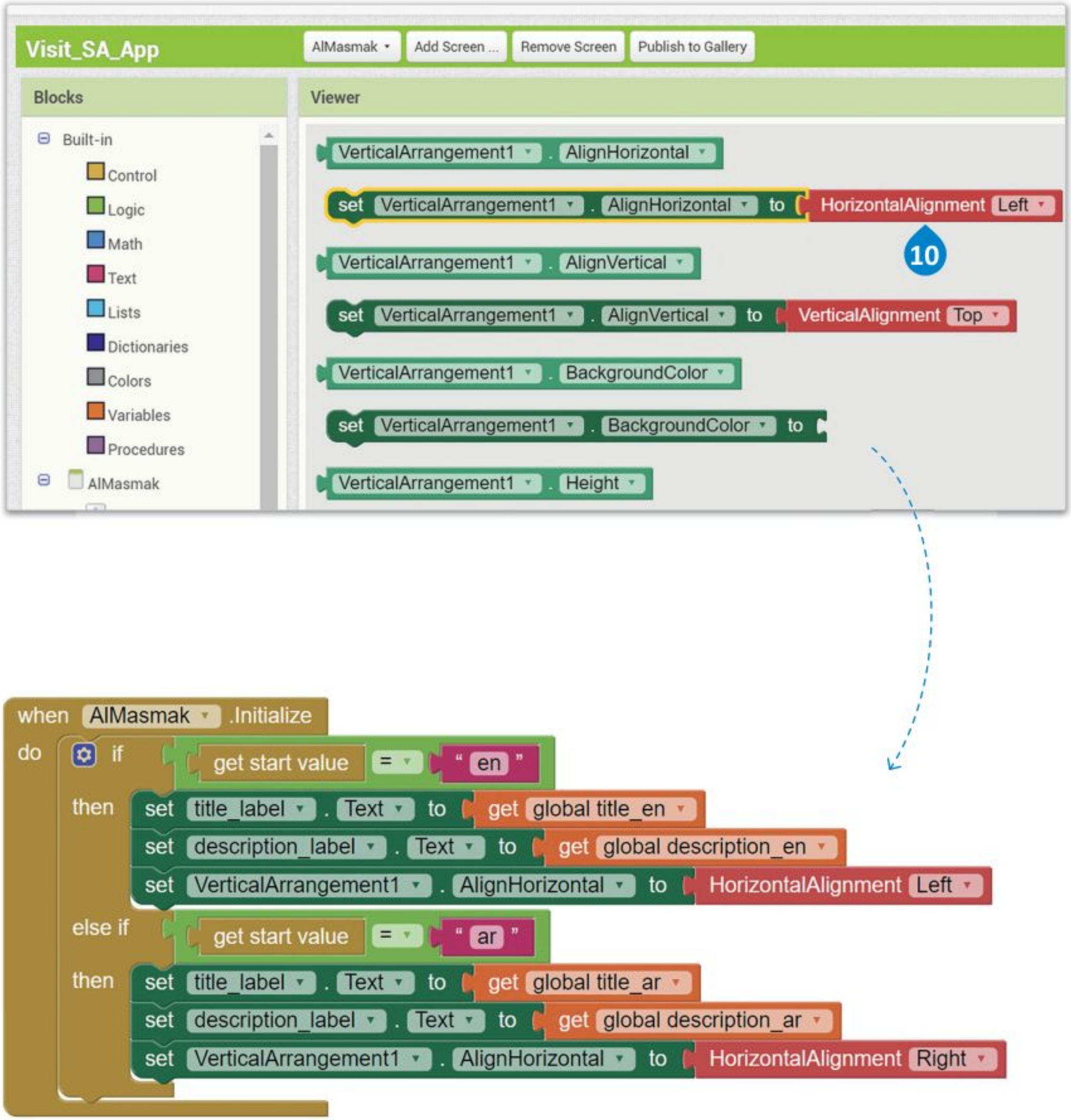
MIT APP INVENTOR
Projects
Connect
Build
Visit_SA_App
AlMasmak
Add Screen ...
Remove Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
join
length
is empty
7
when AlMasmak .Initialize
do
if
get start value
=
" en "
then
else if
get start value
=
" ar "
then


Blocks
Viewer
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
AlMasmak
discover_label
VerticalArrangement1
title_label
image
description_label
HorizontalArrangem
map_button
Map1
Any component
title_label . Height
set title_label . Height to
set title_label . HeightPercent to
title_label . Text
set title_label . Text to
title_label . TextColor
set title_label . TextColor to
8


when AlMasmak .Initialize
do
if
get start value
=
" en "
then
set title_label . Text to
get global title_en
else if
get start value
=
" ar "
then
set title_label . Text to
get global title_ar


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
AlMasmak
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
AlMasmak
discover_label
VerticalArrangement1
title_label
image
description_label
HorizontalArrangem
map_button
Map1
Any component
description_label . Height
set description_label . Height to
set description_label . HeightPercent to
description_label . Text
set description_label . Text to
9
description_label . TextColor
set description_label . TextColor to


when AlMasmak .Initialize
do if get start value = " en "
then set title_label . Text to get global title_en
set description_label . Text to get global description_en
else if get start value = " ar "
then set title_label . Text to get global title_ar
set description_label . Text to get global description_ar



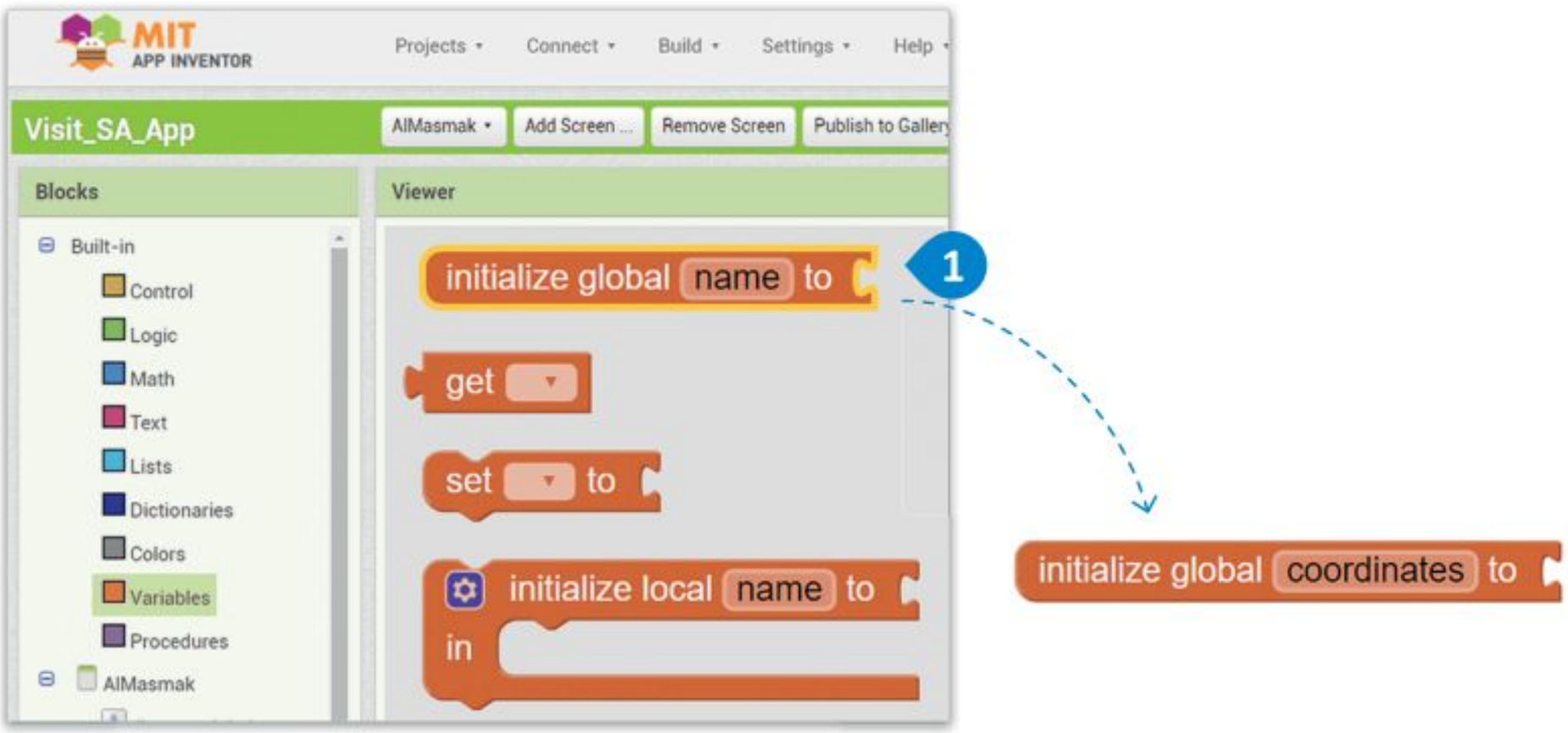
شكل 3.42: تغيير اللغة ديناميكيًا

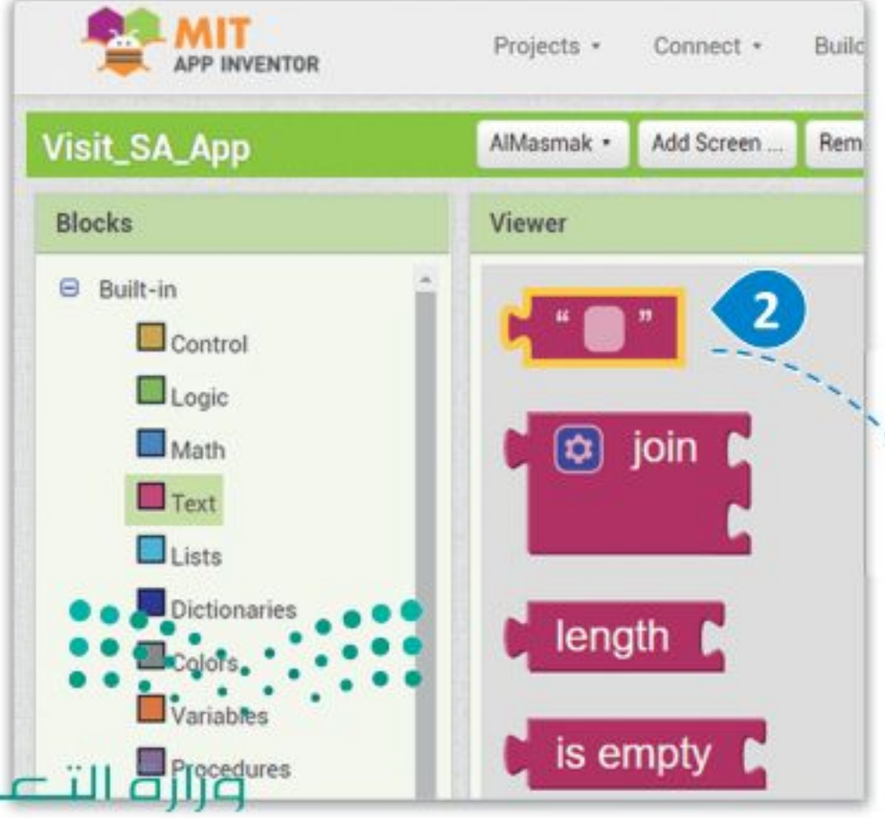
برمجة الخريطة التفاعلية Programming the Interactive Map

سيفتح التطبيق خريطة تفاعلية للمَعْلَم عند ضغط المُستخدِم على زرّ الخريطة، وسيتم تهيئتها حسب إحداثيات المَعْلَم.

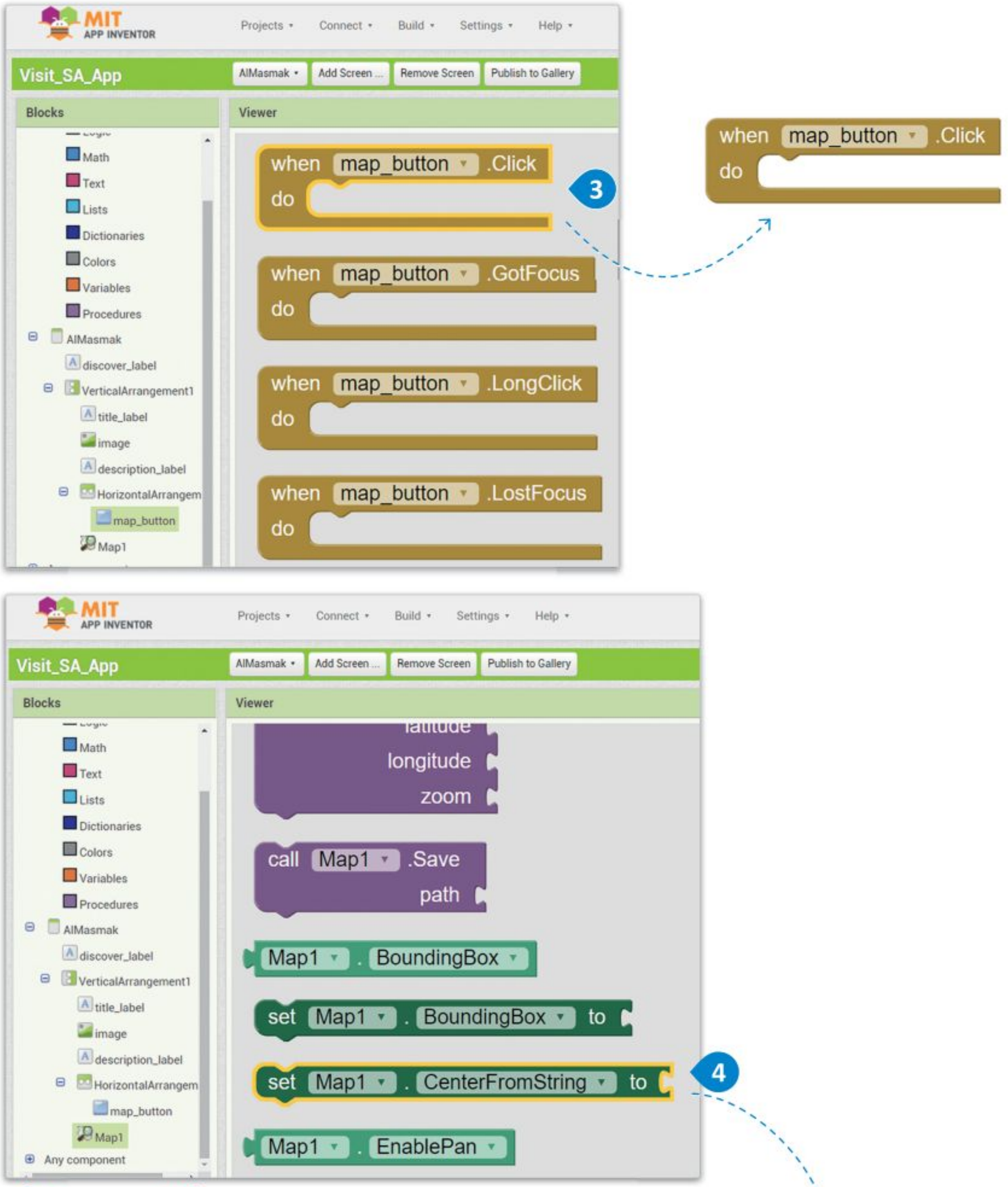
لبرمجة الخريطة التفاعلية:

> أضف متغيرًا جديدًا باسم coordinates (الإحداثيات). 1
> أضف النص التالي إلى متغير coordinates (الإحداثيات): 46.7134، 24.6312 . 2
> حدِّد حدث Click (الضغط) لمُكوِّن map_button (زرّ_الخريطة). 3
> اضبط خاصية CenterFromString (توسيط من السلسلة النصية) لمُكوِّن map (الخريطة) إلى المتغير coordinates (الإحداثيات). 4
> اضبط خاصية Visible (مرئي) لمُكوِّن map (الخريطة) للبنة التعليمات البرمجية لتكون true (صحيحة). 5





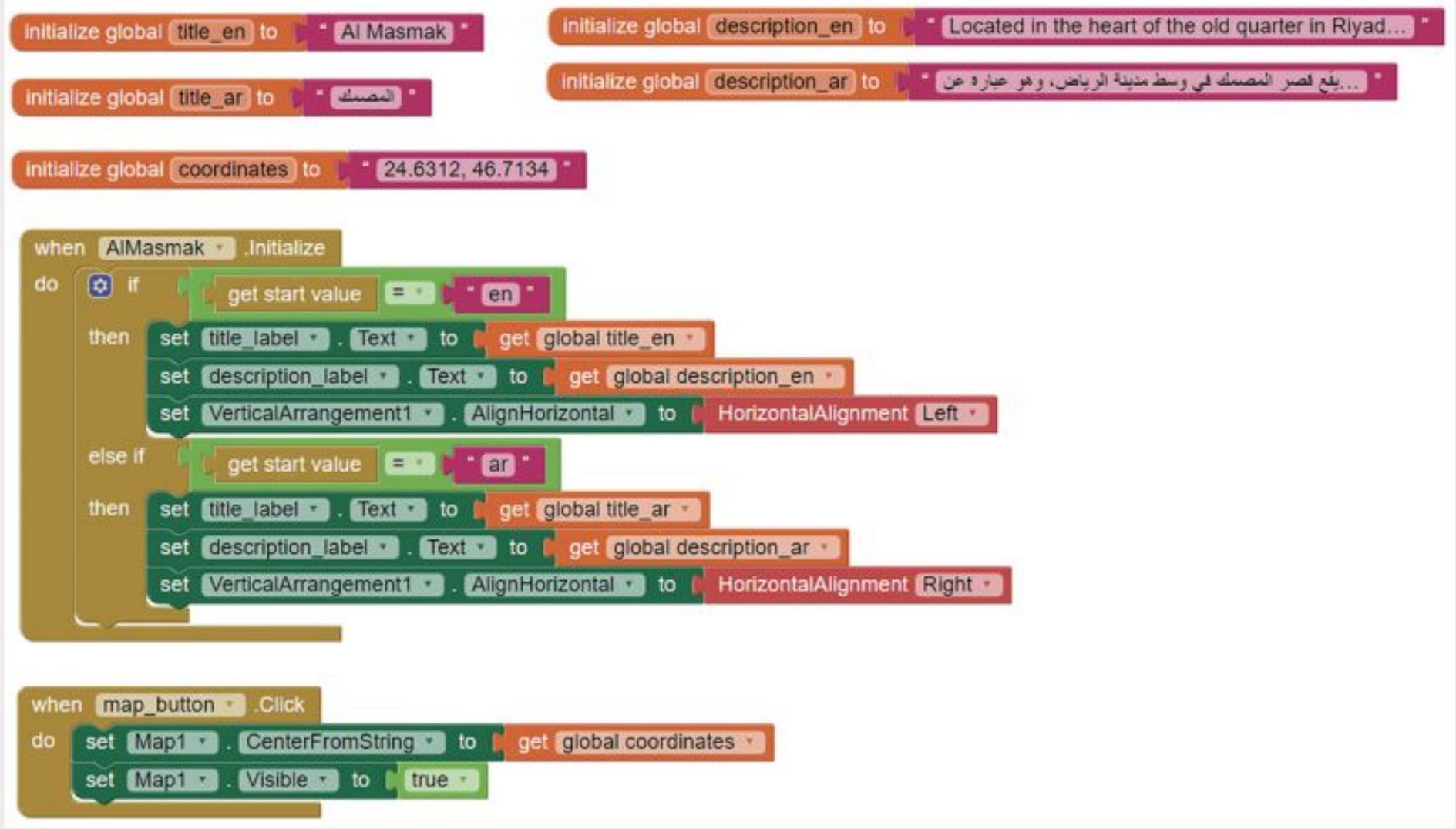
وزارة التعليم
Ministry of Education
2023 - 1445


MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
AlMasmak
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
AlMasmak
discover_label
VerticalArrangement1
title_label
image
description_label
HorizontalArrangem
map_button
Map1
when map_button .Click
do
when map_button .GotFocus
do
when map_button .LongClick
do
when map_button .LostFocus
do
3
when map_button .Click
do
latitude
longitude
zoom
call Map1 .Save
path
Map1 . BoundingBox
set Map1 . BoundingBox to
set Map1 . CenterFromString to
Map1 . EnablePan
4
Any component
when map_button .Click
do set Map1 . CenterFromString to get global coordinates

شكل 3.43: برمجة الخريطة التفاعلية

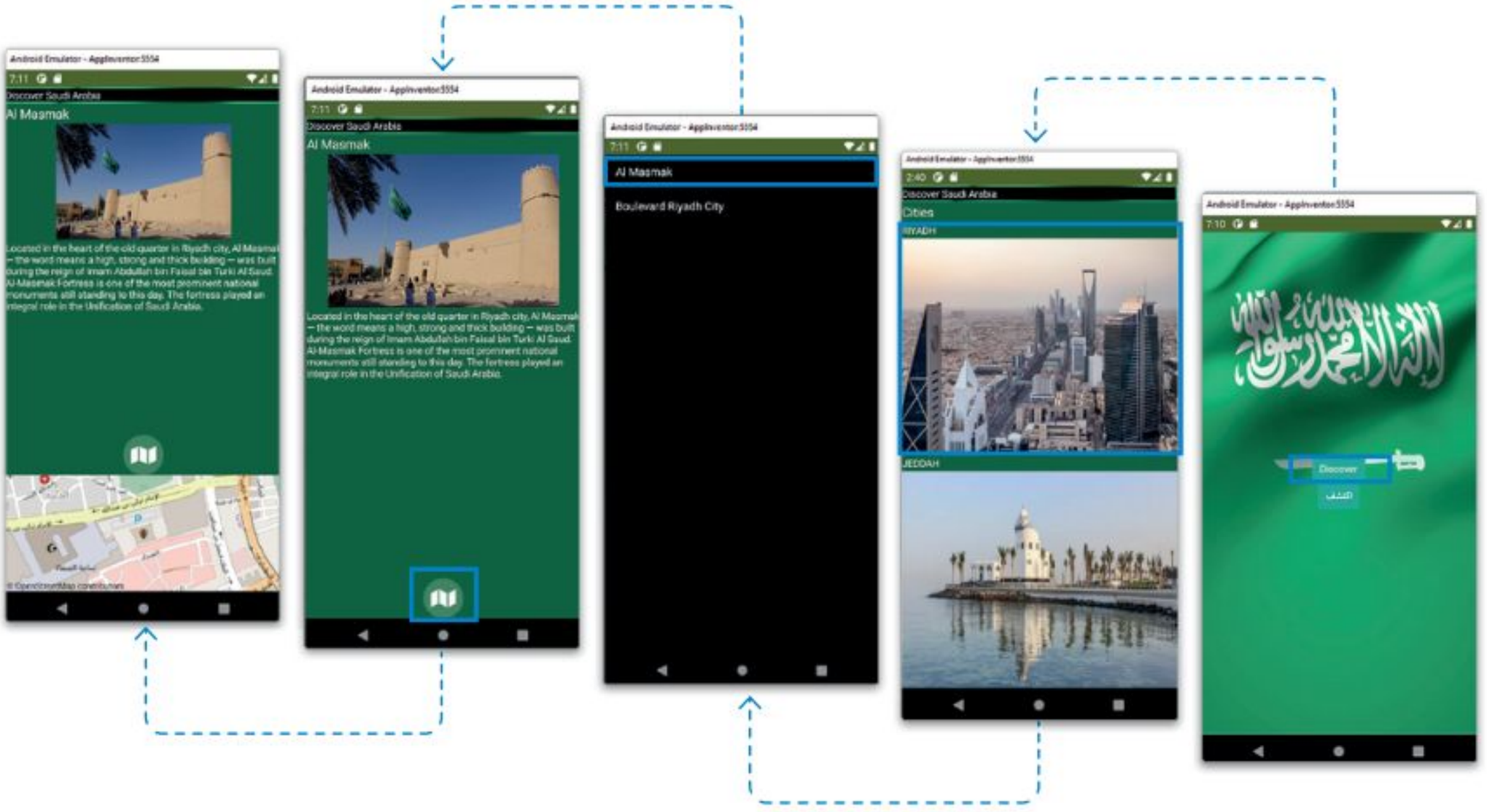
البرنامج بشكله النهائي للشاشة الثالثة (المصمك)

The Complete Code for the Third Screen (Al Masmak)

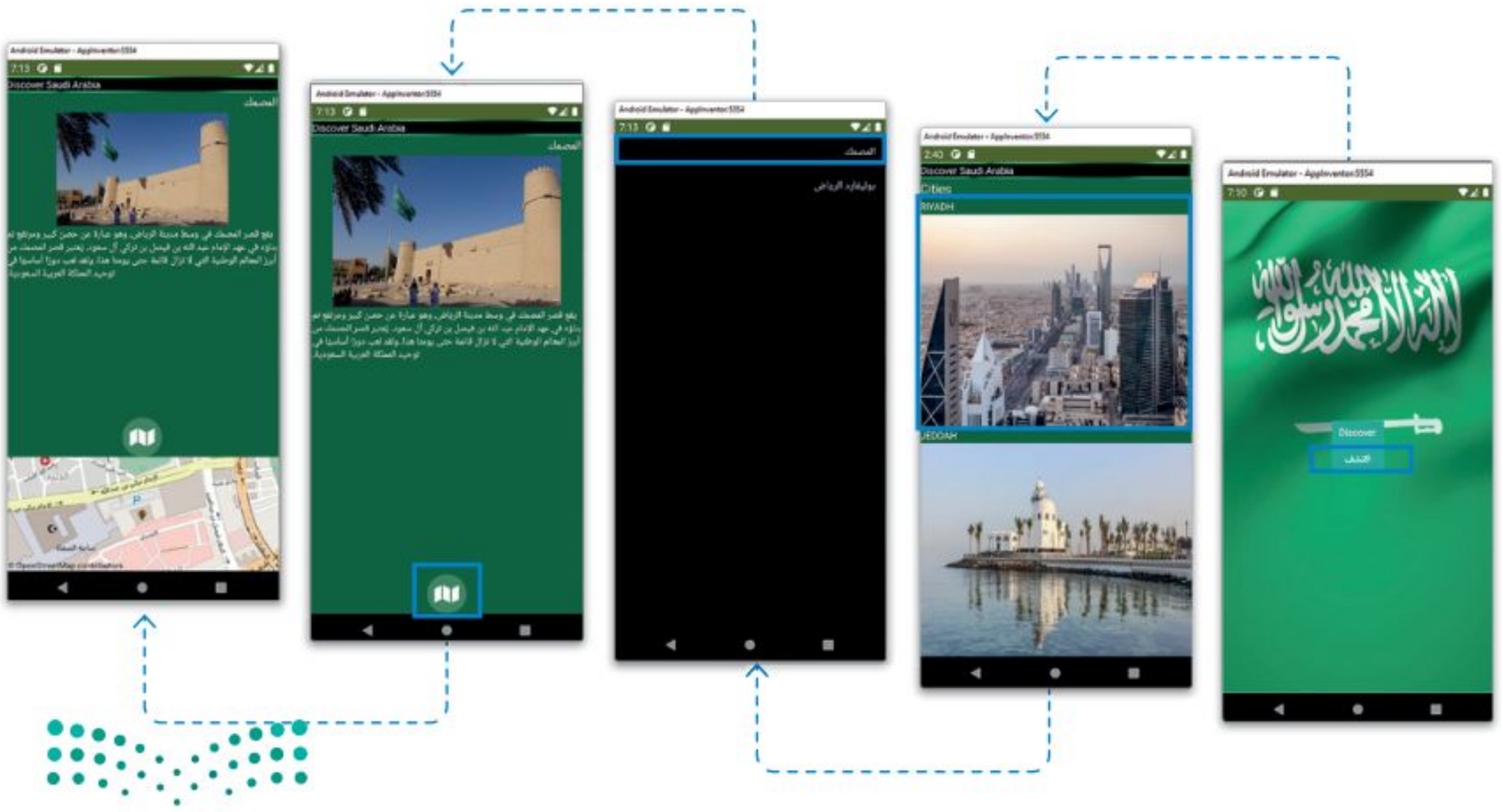


شكل 3.44: البرنامج بشكله النهائي للشاشة الثالثة

لقد أصبح التطبيق جاهزًا الآن وعليك اختباره. يمكنك استخدام **محاكي الأندرويد** (Android Emulator)، أو تنزيل التطبيق وتثبيته وتشغيله على جهاز أندرويد الخاص بك، ويمكنك أيضًا مسح رمز **الاستجابة السريعة** (QR) باستخدام جهاز الأندرويد لمعاينته. يمكنك مشاهدة شاشات تطبيقك عند تشغيل البرنامج باستخدام المحاكي في الصور التوضيحية التالية:



فيما يلي الشاشات التي ستظهر عند الضغط على الزرّ باللغة العربية:



شكل 3.45: اختبار التطبيق في المحاكي

تمرينات

1 وضّح كيف يتيحُ المُكوّنان HorizontalArrangement (الترتيب الأفقي) وVerticalArrangement (الترتيب العمودي) إمكانية تشكيل تخطيط الصفحة.

2 أضف معالم أخرى بارزة لكل مدينة في تطبيقك وأنشئ المزيد من الصفحات لكل مَعْلَم. ابحث في الإنترنت عن المعلومات والصور حول كل مَعْلَم جديد.

3 أضف زرّي انتقال جديدَين في كل صفحة، وبرمجهما بحيث ينقل أحدهما المُستخدِم إلى الشاشة الرئيسة، وينقلُه الآخر إلى الشاشة السابقة.

4 أضف صَفًا إضافيًا يحتوي على أزرار لتبديل اللغة من الإنجليزية إلى العربية في كل صفحة. عدّل برمجة كل صفحة لتنفيذ هذه العملية.

5 في الصفحة الخاصة بالمَعْلَم، أضف label (تسمية) جديدة تعرض الإحداثيات الخاصة بمُكوّن الخريطة التفاعلية. يمكنك العثور على خصائص هذه الإحداثيات بالضغط على مُكوّن map (الخريطة) من صفحة اللبنات البرمجية.

6 في الصفحة الخاصة بالمَعْلَم، أضف زرّين جديدين يمكّنان المستخدم من اختيار نوع الخريطة التفاعلية الذي يفضله من بين النوعين: (Aerial) العرض الجوي و (Road views) عَرض الشوارع. يمكنك العثور على خصائص نوع الخريطة من خلال الضغط على مُكوّن الخريطة في صفحة اللبنات البرمجية.

المشروع

1. في الوحدات السابقة بدأت مشروعك الذي يتمحور حول رؤية عام 2030، وأنشأت نموذجًا أوليًّا باستخدام برنامج **بنسل** (Pencil). ستنشئ الآن تطبيقك.

2. أنشئ تطبيقًا في **مخترع التطبيقات** (App Inventor) يهدف إلى عرض المدن العملاقة في رؤية 2030، وهي أمالا ونيوم والقدية.

3. سيتمكن المُستخدِم من اختيار المدينة والتعرف على بعض الصور الخاصة بها والاطلاع على وصف لمشروعاتها.

4. طوِّر التطبيق مع عناصر التحكم في التصميم وعناصر التنقّل المناسبة.

ماذا تعلّمت

- تصميم واجهة مُستخدِم للتطبيق مع نموذج مبدئي شبكي.
- تطوير تطبيق تجربة مُستخدِم وظيفي وتفاعلي للمُستخدِمين.
- استخدام التغذية الراجعة لتحسين التطبيق بصورة مستمرة.
- برمجة منطق الأعمال المُعقد للتطبيق.
- تجميع محتويات التطبيق وعرضها بشكل سليم.

المصطلحات الرئيسة

ترتيب أفقي	HorizontalArrangement
قائمة الخيارات	ListPicker
متغيرات	Variables
ترتيب عمودي	VerticalArrangement

اللبنات البرمجية	Blocks
الإحداثيات	Coordinates
مُحاكي	Emulator
حدث	Event
معالج الأحداث	Event handler

4. قابلية الوصول إلى البرمجيات والتضمين الرقمي

سيتعرّف الطالب في هذه الوحدة على عملية اختبار التطبيقات، وسيختبر التّطبيق الّذي أنشأه في الوحدة السّابقة وينشره، وسيتعرّف أيضًا على المفاهيم المتعلقة بالفجوة الرقميّة ومبادئ التضمين الرّقميّ وتحدّياته، وكيفية التغلب عليها. وفي الختام سيُطوِّر ميزات قابلية الوصول لتطبيق الهاتف المحمول الذي أنشأه لجعله أكثر شمولًا وقابليّة لوصول المُستخدِمين ذوي الاحتياجات الخاصة.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يُميِّز المراحل والمنهجيات المختلفة لاختبار البرمجيات.
- يَختبِر تطبيقات الهاتف المحمول في مخترع التطبيقات MIT وينشرها.
- يُحدِّد المشاكل الناتجة عن الفجوة الرقميّة.
- يتعرَّف على الحلول المتاحة لتعزيز التضمين الرقمي.
- يُحلِّل ميزات قابلية الوصول المختلفة للأجهزة وكذلك للبرمجيات.
- يُعزِّز تطبيقات الهاتف المحمول بإضافة ميزات قابلية الوصول.

الأدوات

- مخترع التطبيقات MIT (MIT App Inventor)
- بنسل بروجكت (Pencil Project)

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول

اختبار التطبيقات ونشرها

أهمية اختبار التطبيق The Importance of Application Testing

من الطبيعي أن يشعر المبرمِج أو المطوِّر عند الانتهاء من تطوير أحد التطبيقات بالثقة التامة بأن تطبيقه يعمل بشكلٍ صحيح. في الواقع، قد يحدث كثير من الأخطاء للعديد من الأسباب، وقد لا تُحقِّق النسخة الأولى من التطبيق النتائج المرجوة منه. ولذلك يجب عليك التحقّق من عمله بشكل صحيح، والعثور على الأخطاء إن وجدت، واكتشاف الأخطاء الأخرى غير المكتشفة سابقًا، والتي تتعلق باستخدام التطبيق بشكل غير الصحيح.

ما جودة البرمجيات؟ What is Software Quality

جودة البرمجيات هي دراسة التطبيق أو المُنتَج البرمجي بشكلٍ عام للتحقق مما إذا كان يفي بمواصفات المُستخدِم، وما إذا كان يعمل بشكلٍ صحيح من كافة النواحي. تهتم جودة البرمجيات بالتحقق من الأخطاء مهما كبُرت أو كَان عددها قليلًا، وتولي اهتمامًا خاصًا بإمكانيات الصيانة والتحسين للمُنتَج.

يوضِّح الجدول التالي سمات الجودة الرئيسة للمُنتَج البرمجي:

جدول 4.1: السمات الرئيسة لجودة البرمجيات

السمة	الوصف
آداء الوظائف (Functionality)	يعمل المُنتَج البرمجي بشكلٍ يفي بمتطلبات المُستخدِم النهائي، ويمكنه إنجاز جميع المهام المحدَّدة.
الموثوقية (Reliability)	يمكن تصنيف المُنتَج البرمجي على أنه موثوق به ويعتمد عليه عندما يخلو من الأخطاء ويؤدي المهام المُتوقَّعة بموارد محدودة.
قابلية الاستخدام (Usability)	يكون المُنتَج البرمجي أكثر قابلية للاستخدام إذا تمكَّن المُستخدِمون باختلاف قدراتهم من استخدامه والوصول بسهولة إلى إمكانياته.
الكفاءة (Efficiency)	يكون المُنتَج البرمجي ذو كفاءة عند عمله بشكلٍ سليم دون أن يهدر موارد الجهاز الذي يعمل عليه مثل طاقة المعالِج أو الذاكرة أو الشبكة.
قابلية الصيانة (Maintainability)	يكون المُنتَج البرمجي قابلًا للصيانة إذا كان بالإمكان إصلاح الأخطاء وتنفيذ الميزات الجديدة فيه بسهولة.
قابلية النقل (Portability)	يكون المُنتَج البرمجي قابلًا للنقل إذا كان بإمكانه العمل على أنظمة تشغيل أخرى على أجهزة مختلفة ومع برامج أخرى.

الفرق بين تصحيح أخطاء التطبيق والاختبار
Difference between Application Debugging and Testing

التصحيح (Debugging) هو عملية إزالة الأخطاء والأعطال من البرمجيات التي يتم اكتشافها عادةً بعد الاختبار، في حين أنّ **الاختبار** (Testing) هو عملية التحقق من صحة البرمجيات. على سبيل المثال، قد يحدث أثناء الاختبار ألّا يعرضَ مُكوِّن محدَّد في **واجهة المُستخدِم** (UI) لتطبيق المعلومات الصحيحة للمُستخدِم. ستجد خلال عملية التصحيح أن هناك عملية حسابية خاطئة في **منطق البرنامج** (Program Logic) هي المُتسببة في الخطأ السابق أثناء عرضه، وبذلك سيتمّ تصحيح تلك العملية فقط وليس مُكوِّن واجهة المُستخدِم. يجب إجراء الاختبار المناسب للوصول إلى مرحلة التصحيح.

لا يضمن الاختبار صحة البرنامج بشكل كامل، أو بنسبة 100%، ولكنه يكشف عن الأخطاء المحتملة.

مَن ينفِّذ الاختبارات؟ Who Performs Tests

غالبًا ما يقوم المبرمِج أو المطوِّر باختبار البرنامج الذي ينشئه، ولكن اختبار المبرمِج لبرنامجه لا يُعدُّ الخيار الأفضل من حيث ضمان الجودة؛ نظرًا لأنه يصعب عليه أحيانًا ملاحظة أخطائه؛ ولذلك سيكون في حاجة إلى تكليف شخص آخر بإجراء الاختبار، والذي يُطلق عليه لقب المُختبِر. يقوم المُختبِر بإجراء الاختبارات على البرنامج مع التركيز على وظائفه، حيث يتم اختبار النتائج باستخدام مجموعات بيانات الإدخال المختلفة. وهكذا فإنَّ المبرمِجين يقومون بإجراء الاختبارات الأولية، ولكن المُختبِر هو المُخوَّل بالحكم على جودة البرمجيات وعملها كما هو مُتوقَّع. تتضمن بعض شركات تطوير البرمجيات قسمًا خاصًا للاختبار مهمته التحقق من صحة عمل البرامج، ويُطلق عليه عادةً اسم **قسم ضمان الجودة** (Quality Assurance Department).

اختيار بيانات الاختبار Choosing Test Data

إن أفضل طريقة لاختبار البرنامج هي حساب مُخرَجاته المُتوقَّعة يدويًا قبل تشغيله، ثم معرفة ما إذا كانت نتائج تنفيذه تتطابق مع ما تم حسابه أم لا، وبمعنى آخر، عليك كتابة المُخرَجات المُتوقَّعة من البرنامج قبل تشغيله ومقارنتها مع المُخرَجات الحقيقية له، ويجب أن يعمل البرنامج بشكل صحيح بغض النظر عن البيانات التي يُدخِلها المُستخدِم. ولكن في حالة إدخال بيانات غير صالحة، يجب تنبيه المُستخدِم إلى أن البيانات غير مقبولة ويُطلَب منه إعادة الإدخال، ويحتاج المطوِّر أن يأخذ بالاعتبار جميع القيم الممكنة للبيانات المُدخَلة، ولاختبار البرنامج بشكل صحيح، تحتاج إلى اختيار بيانات الاختبار التي تُمثِّل جميع احتمالات مُدخَلات المُستخدِم. تنقسم بيانات الاختبار إلى الفئات التالية:

- **بيانات عادية** (Normal Data): تُستخدَم البيانات العادية عند تعامل المُستخدِم مع البرنامج بشكلٍ طبيعي، وتتضمن مجموعات من القيم من نفس نوع البيانات المُتوقَّعة. على سبيل المثال، إذا كان عليك إدخال قيمة الشهر كعدد صحيح من 1 إلى 12، فإن البيانات العادية هي عدد صحيح من 1 إلى 12.
- **بيانات حدودية** (Boundary Data): هي البيانات الموجودة على **القيم المتطرفة** (Outliers) لنطاق القيم المُتوقَّعة. على سبيل المثال، إذا كنت تتوقَّع إدخال قيمة العام بين 1900 و2020، فإن القيم المتطرفة هي 1900 و2020، لذا فأنت تختبر البرنامج عند إدخال 1900 أو 2020 كأرقام في البرنامج لمعرفة ما إذا كانت هناك أية أخطاء.
- **بيانات خاطئة** (Erroneous Data): هي البيانات الواقعة خارج نطاق القيم المُتوقَّعة، بالإضافة إلى أي نوع آخر غير صحيح من البيانات. في المثال السابق، إذا أدخل المُستخدِم قيمة الشهر بالرقم 0 أو 13، أو أدخل كلمة يناير بدلًا من العدد الصحيح 1، فسيكون هناك خطأ.

الاختبار المؤتمت Automated Testing

تصبح كثير من البرمجيات أكثر تعقيدًا بعد إصدارها التجريبيّ، وبشكلٍ خاص عند خضوعها للتحسينات والتعديلات التي قد تُغيِّر وظائفها أو تضيف عليها مزيدًا من الوظائف. في هذه الحالة، يجب تنفيذ الاختبارات السّابقة نفسها، وعمل بعض

وزارة التعليم
Ministry of Education
2023 - 1445

الاختبارات الجديدة للتحقق من عمل البرمجيات بشكل صحيح. يحتاج العدد الكبير من الاختبارات إلى المزيد من الوقت والجهد عند تنفيذها بشكلٍ يدوي، لذلك يلجأ المُختبِرون إلى إنشاء الاختبارات المؤتمتة التي يتم تحديثها كل مرة يتم فيها تغيير البرنامج. يَكتب المُختبِر المقطع البرمجي الخاص بالاختبار باستخدام عدة أدوات لأتمتة هذه العملية، ومن أمثلة أُطُر الاختبارات الأكثر شيوعًا لنظام **الأندرويد** (Android): **أبيوم** (Appium)، **واسبريسو** (Espresso)، **ويو آي اوتوميتور** (UI Automator).

استراتيجيات الاختبارات Testing Strategies

تنقسم الاختبارات إلى عدة فئات اعتمادًا على درجة تعقيد البرنامج أو التطبيق قيد الاختبار. يَستخدِم المطوِّرون والمسؤولون عن الاختبارات العديد من استراتيجيات الاختبار، ويُعدُّ اختبارُ الوظائف البرمجية، واختبارُ البرنامج الرئيس هما الأكثر استهلاكًا للوقت والجهد، والجدول التالي يوضّح استراتيجيات الاختبارات الأكثر شيوعًا:

جدول 4.2: استراتيجيات الاختبارات الشائعة

الاسم	الوصف
اختبارات التشغيل التجريبي (Dry Run Testing)	يُمكنك من خلال هذه الطريقة تتبُّع تسلسل منطق البرنامج أثناء تنفيذ الحاسب لكل عبارة برمجية في المقطع البرمجي، مع تسجيل قيمة كل متغير في جدول التتبُّع.
اختبارات قابلية الاستخدام (Usability Testing)	يتم إجراء اختباري قابلية الاستخدام أو تجربة المُستخدِم (UX) للتأكد من أن البرمجيات سهلة الاستخدام وواضحة للمُستخدِم النهائي.
اختبارات الصندوق الأسود (Black-Box Testing)	يتم خلالها التعامل مع أجزاء البرنامج التي يتم اختبارها كصندوق مغلق، وذلك بتجاهل طبيعة المقطع البرمجي والتعامل فقط مع بيانات الإدخال والإخراج لمعرفة ما إذا كان المُختبِر يحصل على النتائج المُتوقَّعة عند إدخال البيانات أم لا.
اختبارات الصندوق الأبيض (White-Box Testing)	يُمكن للمُختبِر في اختبار الصندوق الأبيض أن يطّلع على المقطع البرمجي، ولذلك تركِّز عملية الاختبار على التحقق من صحة تنفيذ البرنامج. يتضمن ذلك اختبار منطق البرمجة، وهياكل البيانات، والخوارزميات، ومعالجة الأخطاء، والشروط الحدودية.
اختبارات الوحدة (Unit Testing)	هي اختبار وظيفة كل برنامج بمفرده للتأكد من أداء كل عملية على حِدة، قبل التحقق من عمل البرنامج بأكمله.
الاختبارات التكاملية (Integration Testing)	تتحقق الاختبارات التكاملية من سلوك أجزاء البرنامج المختلفة عندما تعمل معًا كنظام متكامل.
اختبارات الأداء (Performance Testing)	تتحقق اختبارات الأداء من أداء البرنامج أو النظام عند حدوث زيادة كبيرة في عدد المُستخدِمين أو البيانات التي تتم معالجتها. تسلط اختبارات الأداء الأضواء على المشاكل التي تحتاج إلى إصلاح لضمان قابلية التوسّع للبرنامج أو النّظام.
اختبارات القَبول (Acceptance Testing)	تتعلق اختبارات القبول بالتحقق من تلبية البرنامج أو النظام لجميع متطلبات المُستخدِمين باحتياجاتهم المختلفة، وعادة ما يتم عمل هذا النوع من الاختبارات عند إنشاء البرمجيات الكبيرة متعددة المُستخدِمين.
اختبارات الاختراق (Penetration Testing)	تركِّز اختبارات الاختراق على أمان البرنامج أو النظام، وتتحقق من كيفية حماية البرنامج من الهجمات والاختراق.
اختبارات الضغط (Stress Testing)	هو أسلوب اختباري يركز على تقييم أداء النظام البرمجي في ظل ظروف قاسية. يهدف اختبار الضغط إلى تحديد نقطة انهيار النظام البرمجي وتحديد كيفية تصرفه عندما يكون تحت أقصى ضغط.

تخطيط الاختبار Test Planning

إن **خطة الاختبار** (Test Plan) أو **جدولة الاختبار** (Test Schedule) هي قائمة تحتوي على الاختبارات المُخطَّط إجراؤها للتحقق من دقة عمل البرنامج، وتسجيل نتائج كل اختبار.

- يتضمن الجدول بيانات الاختبار، والغرض منه، والنتائج المُتوقَّعة والفعلية لتشغيل البرنامج. يُطلق على كل صف في جدول الاختبار **حالة اختبار** (Test Case).
- يتحقق سيناريو الاختبار من صحة جزء محدد من وظائف البرنامج، وقد يحتوي على واحدة أو مجموعة من حالات الاختبار.
- يتم وضع معايير قبول محددة بدقة في كل سيناريو اختبار.

يجب أن تختلف **حالة الاختبار** (Test Case) عن **حالة الاستخدام** (Use Case)، فكما رأيت سابقًا تحدِّد حالة الاستخدام كيفية استخدام البرنامج أو النظام لأداء مهمة محدَّدة، وعادةً ما تكون على شكل مُخطَّط يوضِّح تسلسل الإجراءات التي سيتبعها المُستخدِم عند تفاعله مع البرنامج. على النقيض، يهدف الاختبار إلى إنشاء حالات خاطئة بشكلٍ متعمد وذلك باستخدام بيانات صالحة وغير صالحة. غالبًا ما يتم التخطيط لسيناريوهات الاختبار وحالاته قبل إتمام البرمجة الفعلية.

توثيق الاختبار Test Documentation

تحتاج عملية الاختبار إلى توثيقٍ دقيق لتحقيق الاستفادة المرجوة منها في اختبارات الإصدارات التالية للبرنامج، وتتضمن عملية توثيق الاختبار ما يلي:

- **سياسة الاختبار** (Testing Policy): تحتوي على وصف مبادئ الاختبار وأساليبه وأهدافه.
- **خطة الاختبار** (Test Plan): تحتوي على وصف البرمجيات ووظائفها والأجزاء المطلوب اختبارها ونطاق الاختبارات.
- **مواصفات الاختبار** (Test Specifications): تحتوي على تفاصيل كل سيناريو من سيناريوهات الاختبار ومعايير التقييم الخاصة به.
- **وصف الاختبار** (Test Description): يحتوي على بيانات الاختبار والإجراءات لكل حالة اختبار.
- **تقرير تحليل الاختبار** (Test Analysis Report): يحتوي على نتائج كل سيناريو اختبار.
- **تقرير الخطأ** (Bug Report): يحتوي على تقرير عن أي خلل أو خطأ أو مشكلة في البرمجيات.
- **تقرير ملخص الاختبار** (Test Summary Report): يحتوي على التقرير النهائي الذي يلخص عملية الاختبار بكاملها.

اختبار التطبيق السياحي لزيارة السعودية Testing Visit Saudi Tourism Application

بعد إنشائك التطبيق الخاص بك، عليك توزيعه ليتم اختباره. من المهم أن يمتلك المُختبِرون خلفياتٍ متعددة ومختلفة، حيث يمنح ذلك المُطوِّر مزيدًا من المعلومات حول ما يجب تصحيحه في التطبيق، والميزات التي يجب إضافتها. على سبيل المثال، سيواجه مُستخدِمو تطبيقك الذين يعانون من صعوبة الرؤية أو ضعف السمع، صعوبات في الحصول على المعلومات الصحيحة. في درس لاحق، ستقوم بتنفيذ ميزات إمكانية الوصول لهذا التطبيق.

حَزْم التّطبيق وتوزيعه Packaging and Distributing an Application

بعد الانتهاء من تطوير تطبيقِك واختباره، ستحتاج إلى **حَزمه** (Package) في تنسيق ملف قابل للتنزيل والتثبيت على الأجهزة المحمولة بنظام الأندرويد. يمكنك توزيع التطبيق بإحدى هاتين الطريقتين:

- **النشر في متجر التطبيقات** (Publishing to a store application): تحميل الحِزمة إلى متجر التطبيقات حتى يتمكن المُستخدِمون من العثور عليها من أي مكان.
- **تنزيل الحِزمة على هاتفك** (Downloading a package on your phone): تنزيل الحِزمة من حاسبك أو من رابط موقع ويب، وتثبيت التطبيق مباشرة على هاتفك المحمول الفعلي.

وزارة التعليم
Ministry of Education
2023 - 1445

تعيين الإصدار لتطبيقك Versioning your Application

بِغَضِّ النظر عن الطريقة التي تعتمدها في توزيع تطبيقك، فإن الخطوة الأولى في حَزم تطبيقك هي **تعيين الإصدار** (Versioning). تحتاج التطبيقات إلى تنفيذ الإصلاحات والتحديثات بشكلٍ مستمر، ولا يمكن لأي تطبيق أن يستمر بالعمل دون التطوير المستمر. من المعتاد أن يحتفظ التطبيق بالاسم نفسه، ولكن قَد يتم خلال عملية التحديث إجراء تغييرات في واجهة المُستخدِم أو في بعض وظائف ذلك التطبيق، لذلك يجب أن تكون هناك طريقة للتمييز بين هذه التطبيقات. **تعيين الإصدار** (Versioning) هو استخدام **رمز المُعَرِّف** (Identifier Code) ليشير إلى إصدار التطبيق الذي يقوم المُستخدِم بتثبيته حاليًا، ويعتبر بمثابة الطريقة الأفضل للتمييز بين الإصدارات المختلفة له. في التطبيقات المخصصة لنظام الأندرويد، يتم تحديد تعيين الإصدار بالخصائص التالية:

- **رمز الإصدار** (VersionCode): رقم مُعرِّف افتراضي يبدأ بـ 1، وتتم زيادته في كل مرة يتم تحميل إصدار جديد إلى متجر التطبيقات.
- **اسم الإصدار** (VersionName): سلسلة يمكن تعيينها على أي قيمة، ويتم ضبطها افتراضيًا على الرقم "1.0"، ووفقًا للمعايير المعتمدة دوليًا، يجب زيادة الرقم الأول في كل مرة تقوم فيها بإجراء تحديث رئيس للتطبيق، وزيادة الرقم الثاني في كل مرة تقوم فيها بإجراء تحديث بسيط للتطبيق. على سبيل المثال، التطبيق الذي يبدأ **باسم الإصدار** "1.0 " ويخضع لتحديث طفيف يصبح **اسم الإصدار** "1.1"، وعند تنفيذ تحديث رئيس على هذا التطبيق يصبح **اسم الإصدار** "2.0".

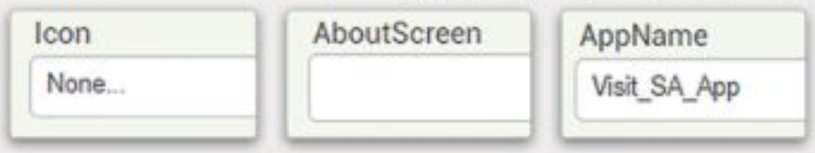
عند العمل باستخدام **مخترع التطبيقات** MIT (MIT App Inventor)، يتاح للمُطوِّر تعديل العديد من خصائص التطبيق العامة بما في ذلك إصدارات التطبيق من قسم properties (الخصائص) من Screen1 (شاشة 1)، وهذا هو سبب عدم إمكانية حذف Screen1 (شاشة 1) من التطبيق.

مظهر التطبيق Application Appearance

يجب تقديم بعض المعلومات للمُستخدِم الذي يرغب بتنزيل تطبيقك من المتجر، وهي:

- **اسم التطبيق** (Application Name): وهو الذي يظهر في المتجر وعلى هاتفك في قائمة تطبيقاتك.
- **وصف التطبيق** (Application Description): النصّ الذي يعطي لمحة موجزة عن التطبيق.
- **شعار التطبيق** (Application logo): أيقونة الشعار التي تظهر في المتجر وعلى هاتفك.

يتم تحديد هذه المعلومات في قسم properties (الخصائص) من Screen1 (شاشة 1)، ويجب تحديد الخصائص التالية:



حَزْم تطبيقك Packaging your Application

لتثبيت تطبيقك على هاتف محمول، عليك تحويل البرنامج الذي أنشأته في **مخترع التطبيقات** (App Inventor) إلى ملف يمكن تثبيته بواسطة أجهزة الأندرويد. يوجد نوعان من الحِزم لتطبيقات الأجهزة المحمولة بنظام الأندرويد:

.apk — تنسيق **حزمة الأندرويد القياسية** (Standard Android package)، وهو التنسيق الذي تم استخدامه منذ إنشاء نظام تشغيل أندرويد، ويُعدُّ هذا التنسيق أبسط طريقة لتوزيع تطبيقات أندرويد. يمكن للمُستخدِم تنزيل ملفات **أي بي كي** (APK) مباشرة من رابط موقع الويب أو تحميلها من متجر **قوقل بلاي** (Google Play).

.aab — **حزم تطبيقات الأندرويد** (Android App Bundles – AABs)، وهي نوع جديد من تنسيق الملفات يَتم استخدامها لحَزم تطبيقات الأندرويد. تحتوي الحزمة على ملف **أي بي كي** (APK)، إضافة إلى بيانات وصفية تسمح للتطبيق بالعمل بسلاسة على مجموعة واسعة من الأجهزة. لا يمكن توزيع هذه الحزم أو تنزيلها إلا من متجر قوقل بلاي.

وزارة التعليم
Ministry of Education
2023 - 1445

تمرينات

1 صِف الفرق بين اختبار تطبيق برمجيّ، وبين تصحيح أخطائه.

2 صنِّف أبرز أنواع استراتيجيات الاختبارات.

3 اذكر المكوّنات الرئيسة لعملية توثيق الاختبار.

4 اذكر طريقتين لتوزيع تطبيق لهواتف تعمل بنظام الأندرويد.

5 صِفْ الاختلافات بين النوعين الرئيسين من حِزم نظام الأندرويد.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثاني

التضمين الرقمي

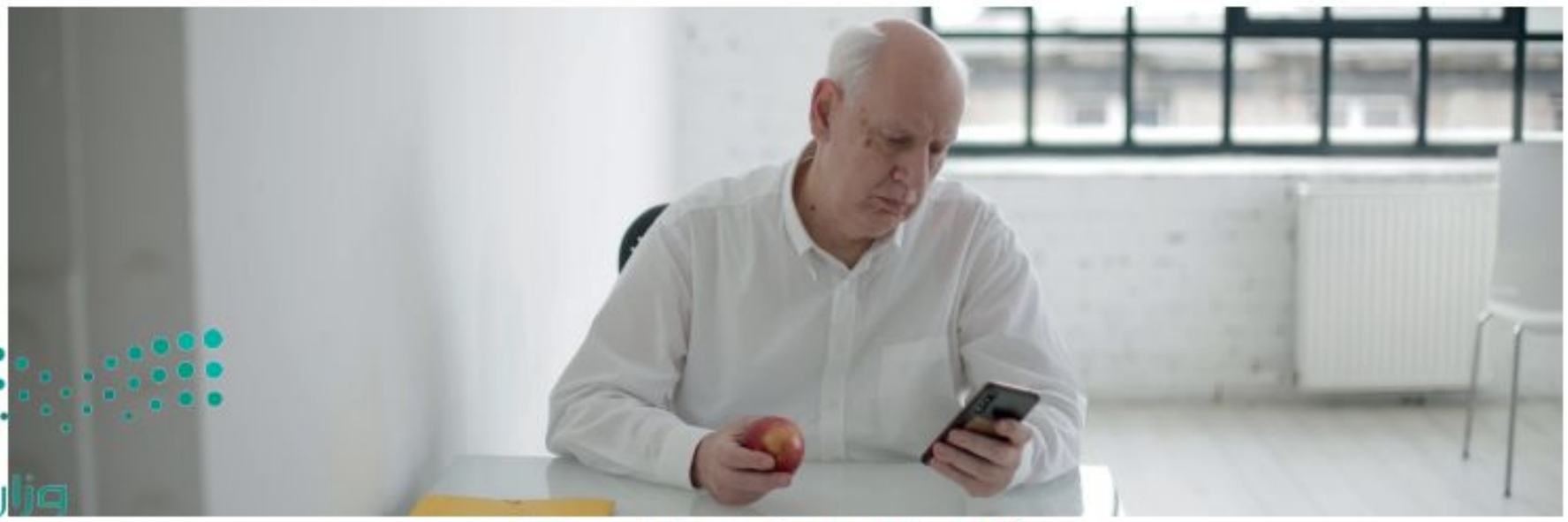
ما الفجوة الرقميّة؟ What is the Digital Divide

إن الفجوة الرقميّة أو التقَنيّة هي مشكلة اجتماعية تشير إلى التباين في كَمّ المعلومات والمهارات المتاحة لدى من تتوفر لديهم إمكانية استخدام أجهزة الحاسب والوصول إليها، وأولئك الذين لا يستطيعون الوصول إليها أو استخدامها. تُعدُّ مسألة توفير الوصول إلى الإنترنت بسرعة عالية وبتكلفة معقولة من أكثر القضايا المطروحة في المجتمعات المختلفة حول العالم في هذه الأيام. انتشر مصطلح الفجوة الرقميّة في أواخر تسعينيات القرن الماضي، وتمحورت التوقُّعات بتقلص هذه الفجوة مع مرور الوقت، ولكن ما حدث هو أن الأمور ازدادت سوءًا مع توسّع هذه الفجوة بشكلٍ أكبر. يجب مناقشة هذه المشكلة على مستوى عالٍ وفي إطارٍ عالمي، حيث بات من الواضح وجود فجوة كبيرة في قدرة الدول المتقدِّمة على الاستفادة من النمو المتزايد في تطور التقنيّة مقارنة بالدول النامية. يُعدُّ الاستخدام المناسب والوصول إلى التقنيّة والاتصالات أمرًا حيويًا لتحسين جودة الحياة. تختلف قدرة المجتمعات على الاستفادة من التقنيّة المتاحة، وأظهرت الأبحاث والدراسات أن تلك الاختلافات قد ترجع إلى وجود أجهزة حاسب منخفضة الأداء أو الجودة، أو بسبب رداءة شبكات الاتصال أو تكلفتها المرتفعة، ويتفاقم الأمر مع صعوبة الحصول على تدريب، أو صعوبة الوصول إلى محتوى عالي الجودة عبر الإنترنت، وعدم القدرة على الحصول على الدعم الفني.

تهدِف المملكة العربية السعودية إلى أن تصبح واحدة من الدول الرائدة في التحول الرقمي.

إن تقليص الفجوة الرقميّة من الخطوات المهمة للغاية في عملية التحول الرقمي، ولذلك قامت حكومة المملكة برسم التشريعات والسياسات التالية لمساعدة مواطني المملكة في هذه العملية، ومن ذلك:

- **التعلُّم الإلكتروني** (E-Learning): استثمرت الحكومة السعودية بشكلٍ واسع في مبادرات التعلُّم الإلكتروني لتوفير تعليم جيد لجميع المواطنين بِغضِّ النظر عن أماكن إقامتهم داخل المملكة. الهدف من استخدام التقنيّة هو الوصول إلى الطلبة في المناطق النائية والمحرومة وتوفير فرص متساوية للحصول على تعليم جيد.
- **الخطة الوطنية للاتصال واسع النطاق** (National Broadband Plan): تهدف الخطة الوطنية للاتصال واسع النطاق إلى زيادة الوصول إليه في جميع أنحاء البلاد وتقليص الفجوة الرقميّة. وتشمل الخطة أيضًا تقديم المساعدات لخدمات الاتصال واسع النطاق، وزيادة الاستثمار في البنية التحتية الرقميّة، وتحسين الوصول إليه في المناطق الريفية والنائية.
- **برامج التضمين الرقمي** (Digital Inclusion Programs): توجد العديد من برامج التضمين الرقمي التي تهدف إلى توفير التدريب على المهارات الرقميّة، والوصول إلى الأجهزة الرقميّة للفئات المجتمعية التي لم تحصل على التعليم العالي، أو المواطنين ذوي الدخل المنخفض في المملكة العربية السعودية. تهدف هذه البرامج إلى تحسين الوصول إلى الخدمات الرقميّة وتقليص الفجوة الرقميّة.



شكل 4.1: تؤثر الفجوة الرقميّة على كبار السن بشكلٍ كبير

العوامل المؤثرة على الفجوة الرقميّة Factors Affecting the Digital Divide

تساهم العديد من العوامل في زيادة الصعوبات الناتجة عن الفجوة الرقميّة، ولا تقتصر هذه الصعوبات على بلد معين، بل تمتد إلى قارات بأكملها، مما يجعل الفجوة الرقميّة قضية عالمية. فيما يلي قائمة بالعوامل التي تؤثر على الفجوة الرقميّة على مستوى العالم:

التعليم Education

تشير الدراسات إلى أن الحاصلين على شهادة جامعية يمكنهم الوصول إلى الإنترنت في أعمالهم أكثر بعشرة أضعاف من أولئك الحاصلين على تعليم ثانوي. إضافة إلى ذلك، يلعب التعليم ومحو الأمية دورًا رئيسًا في تسهيل التعامل مع الحاسب والوصول إلى الإنترنت، ويساعد الوصول إلى المزيد من الموارد والمعلومات إلى تسريع عملية تعلُّم الأفراد.

أعمار المُستخدِمين Users Age

يعاني معظم كبار السن من صعوبات في استخدام التقنيّة مقارنةً بالأجيال الشابة، كما أنهم يحتاجون إلى التدريب والتأهيل والدعم. يمكن تقديم الدعم لهم باستخدام الأدوات التي تسهل عليهم التعامل مع التقنيات الحديثة وبمساعدة أفراد الأسرة.

الموقع الجغرافي Geographical Location

يمتلك المقيمون في المناطق الحضرية والمدن ما يصل إلى عشرة أضعاف أجهزة الحاسب مقارنة بأولئك المقيمين في المناطق الريفية. تُفضِّل شركات الاتصالات إنشاء البُنى التحتية في المناطق الحضرية لخدمة العدد الأكبر من العملاء بتكلفة أقل من المناطق الريفية التي عادة ما تكون مترامية الأطراف وعدد سكانها محدود. يُمكن للشركات والحكومات التغلب على هذه المشكلة من خلال الحلول البديلة، كتوفير الاتصال بالإنترنت من خلال خطوط الطاقة، واتصالات الأقمار الصناعية.

المستوى الاقتصادي Economic Level

تتمتع المجتمعات الأكثر ثراءً بفرص أفضل من تلك الأكثر فقرًا لتبني التقنيات الجديدة. من الشائع أيضًا توفُّر خدمة الإنترنت المجانية في المرافق العامة في المناطق الرّاقية، وذلك على النقيض من المناطق الفقيرة حيث تكون الحاجة أكبر.

اختلافات اللغة Language Differences

تؤثر اللغة بشكل مباشر على تجربة المُستخدِم مع الإنترنت، حيث إنها تحكم كمية المعلومات التي يمكنه الوصول إليها، وجودتها والمجتمعات التي يمكنه التواصل معها. على سبيل المثال، قد يمنحك البحث في **قوقل** (Google) عشرة أضعاف المعلومات بلغة معينة مقارنة بلغة أخرى، فإذا لم تكن لغتك شائعة الاستخدام، فعلى الأغلب لن يتوفر محتوى كافٍ بهذه اللغة على الإنترنت.

الاحتياجات الخاصة Special Needs

قد تكون التقنيّة الحديثة متاحةً لعامة المُستخدِمين، ولكنها ليست كذلك لأولئك الذين يعانون من الإعاقات المختلفة كالإعاقات الحركية في اليد والأطراف، أو الإعاقات السمعية والبصرية. يواجه ذوو الإعاقات والاحتياجات الخاصة صعوبات وعقبات جمة عند استخدام التقنيّة، كعدم القدرة مثلًا على التفاعل مع الشاشات أو الهواتف الذكية. تشير الأبحاث والدراسات إلى أن حوالي 15% من سكان العالم يعانون من الإعاقات المختلفة، كما يواجه ما بين 2% إلى 4% من سكان العالم صعوبة في أداء أنشطتهم اليومية بشكل مستقل.

ما المقصود بالتضمين الرقمي؟ What is Digital Inclusion

يؤكد التضمين الرقمي على الوصول العادل إلى تقنيّة المعلومات والاتصالات واستخدامها لتمكين المشاركة الاجتماعية والاقتصادية، بما في ذلك التعليم، والخدمات الاجتماعية، والصحية، والأنشطة الاجتماعية، والمجتمعية. يشمل التضمين الرقمي الإنترنت واسع النطاق غير المُكلف، والأجهزة التي تدعم الإنترنت، والتدريب على محو الأمية الرقميّة، وتوفير التطبيقات خدمات الدعم الفني المناسبة، وتعزيز المحتوى الرقمي عبر الإنترنت الذي يشجّع الاكتفاء الذاتي والمشاركة والتعاون.

رغم وجود العديد من الحلول طويلة الأمد لمشكلة الفجوة الرقميّة، إلا أنه يجب العمل أولًا على تقليصها، وذلك من خلال معالجة قضايا ضعف البنية التحتية والتعامل مع تداعيات انخفاض مستويات التعليم والفقر. فيما يلي بعض الحلول في مختلف الجوانب التي يمكن أن تساعد في تقليل هذه الفجوة.

تكلفة التقنيّة The Cost of Technology

- خفض أسعار أجهزة المُستخدِمين وتكاليف الخدمات والاتصال بالإنترنت.
- توفير التمويل لمساعدة محدودي الدخل على تحمل أعباء التقنيّة الحديثة، وتخفيض الرسوم الجمركية على الأجهزة التقَنية لتشجيعهم على اقتناء الأدوات الرقميّة.
- تخفيض تكاليف الخدمات الإضافية اللازمة لاستخدام التقنيّة، مثل أجهزة الشحن بالطاقة الشمسية التي تُغني عن الحاجة لأجهزة الشحن الكهربائية.

مواءمة الاحتياجات Convenience

- تطوير المحتوى والتطبيقات باللغات المحلية لإتاحة المجال للمواطنين لفَهْم ذلك المحتوى بسهولة.
- معالجة قضايا الخصوصية وأمن البيانات التي قد تتسبب بتردد المُستخدِمين بشأن التقنيّة المتطورة.
- تعديل أماكن العمل لتناسب ذوي الاحتياجات الخاصة وتطوير البرمجيات المساعدة.
- توفير أجهزة الحاسب لجميع الأشخاص، بغض النظر عن قدراتهم البدنية أو التعليمية.

الكفاءة Efficiency

- توفير التدريب على وسائل تقنيّة المعلومات والاتصالات ومهاراتها.
- التركيز على التعليم وتحديث المناهج بشكل دوري.
- العمل على توفير المستوى المطلوب من المعرفة المعلوماتية لدى المواطنين لاستخدام تقنيّة الحاسب، حيث يواجه الكثيرون تحديات مختلفة بسبب الأمية الرقميّة، وتشمل هذه التحديات الازدياد الهائل في المعلومات وانحسار القدرة على العثور على المعلومات واستخدامها.

البنية التحتية Infrastructure

- توسيع وتحديث الشبكات وزيادة قدراتها الاستيعابية.
- تطوير الحلول ذات التكلفة المعقولة والقدرة الواسعة على التغطية للمناطق الريفية والنائية.
- تطوير بُنى تحتية مستقرة لدعم الاقتصاد الرقمي، ويشمل ذلك مشغلي شبكات الهاتف المحمول من الجيلين الرابع والخامس.
- توفير خدمات الاتصال بالإنترنت المجانية في الأماكن العامة مثل: المقاهي، والمكتبات، وكذلك في المرافق العامة للمدن.

مزايا تقليص الفجوة الرقميّة Advantages of Reducing the Digital Divide

- توسيع شرائح المُستخدِمين المجتمعية التي ستحصل على الخدمات الآلية التي تقدمها الدولة لمواطنيها، مثل الخدمات الإلكترونية الحكومية.
- إفساح المجال لمشاركة فئات المجتمع المختلفة في الاستبانات واستطلاعات الرأي المتعلقة بالخدمات التي تقدِّمها المؤسسات المختلفة.
- توسيع نطاق التعليم ليشمل عددًا أكبر من أفراد المجتمع من خلال استخدام الموارد التعليمية المختلفة المتاحة على الإنترنت.
- إتاحة المجال لأصحاب المشاريع والأعمال لتسويق مُنتَجاتهم، وتقديم مشاريعهم، وإنشاء أفكار لمشاريع جديدة تعتمد على استخدام العملاء للتقنيّة.

شكل 4.2: أهمية تقليص الفجوة الرقميّة

قابلية الوصول لذوي الاحتياجات الخاصة
Accessibility for People with Special Needs

قابلية الوصول هي عملية تصميم المُنتَجات والأجهزة والخدمات والمنشآت بطريقة تُمكِّن الجميع من الوصول إليها واستخدامها، ويُركِّز مفهوم قابلية الوصول على تمكين أو تسهيل وصول الأشخاص ذوي الإعاقة من خلال التقنيّة المساعدة.

عندما يتعلق الأمر بالتقنيّة، تُشير قابلية الوصول إلى الأجهزة والبرمجيات والتقنيات التي صُمِّمت لمساعدة الأشخاص في التغلب على إعاقاتهم، والأمر يشبه تلك المعدات والأدوات التي تساعد الأشخاص في التغلب على الإعاقات مثل: **الكراسي المتحركة** (Wheelchairs)، و**أجهزة السمع** (Hearing Aids)، و**المنحدرات** (Ramps) التي تسهل الدخول إلى المبنى، و**لغة برايل** (Braille).

يجب أن تكون المجالات التالية متاحة لذوي الاحتياجات الخاصة:

قابلية الوصول إلى الأجهزة Hardware Accessibility

- تتضمن أجهزة التقنيّة المسانِدة والمساعدة التي تدعم قابلية الوصول ما يلي: لوحات المفاتيح ذات الأحرف الكبيرة وأجهزة الفأرة الكبيرة، والمفاتيح التي يمكن تفعيلها بالضغط الخفيف، والعديد من الأجهزة الأخرى. توفِّر هذه الأجهزة للمُستخدِمين ذوي الاحتياجات الخاصة القدرة على استخدام أجهزة الحاسب بطرق بديلة. فيما يلي بعض الإرشادات لتصميم أجهزة التحكم والإدخال التي تتميز بقابلية الوصول:
- يجب أن تكون أدوات التحكم البديلة في متناول اليد، ويسهل الوصول إليها، ويمكن تمييزها عن طريق اللمس، ويمكن استخدامها بيد واحدة بطريقة سهلة.
- يتم تصميم المفاتيح وأدوات التحكم في هذه الأجهزة بحيث يتم لمسها والتعرف عليها، دون تنشيطها.
- يجب توفير طرق تحكم بديلة بالأجهزة التي تعمل باللمس مثل الأوامر الصوتية.
- يجب تصميم أجهزة تحكم بديلة للأشخاص ذوي الاحتياجات الخاصة بقدرات للتعرف على حالتهم (الحاجة للفتح أو الإغلاق)، والاستجابة للأوامر باستخدام حواس مختلفة بديلة لحاسة البصر مثل اللمس أو السمع.
- يجب أن تتصل أجهزة التحكم القابلة للوصول بأجهزة الحاسب والأجهزة الذكية باستخدام الأسلاك والمنافذ القياسية المتوفرة على الأجهزة الشائعة.

تُعدُّ المعرفة بقابلية الوصول وأدواتها ضرورية للمؤسسات والمطوِّرين القائمين على إنشاء مواقع الويب وأدوات الويب عالية الجودة بطريقة تضمن وصول مُنتَجاتهم وخدماتهم إلى الأشخاص ذوي الإعاقة.

فيما يلي قائمة بأجهزة الإدخال البديلة للأشخاص ذوي الإعاقات والاحتياجات الخاصة المختلفة:

- **لوحة مفاتيح الحاسب بلغة برايل** (Braille Computer Keyboard): لوحة مفاتيح الحاسب بلغة برايل هي لوحة مفاتيح مخصصة تساعد المُستخدِمين الذين يعانون من إعاقات بصرية على إدخال النصوص والتحكم في أجهزة الحاسب الخاصة بهم. تتكون هذه اللوحة من ست أو ثماني نقاط بارزة تمثل أحرف برايل، ويُمكن للمُستخدِم إدخال نص عن طريق الضغط على النقاط بأصابعه. تم تصميم لوحة المفاتيح لتُستخدَم مع برنامج قارئ الشاشة البرمجي الذي يحوّل إدخال برايل إلى نص مكتوب يتم عرضه على الشاشة.
- **فأرة التحكم عن طريق الرأس** (Head-Mouse Control): تسمى أيضًا باسم **فأرة تتبُّع الرأس** (Head-Tracking Mouse)، وهي جهاز يسمح بالتحكم في الفأرة عن طريق الرأس للأشخاص ذوي الإعاقات الحركية بتحريك فأرة الحاسب باستخدام حركات الرأس. تتكون هذه الفأرة من كاميرا صغيرة تتعقب حركات رأس المُستخدِم وتترجمها إلى حركات لمؤشر الفأرة على الشاشة. يمكن تركيب الكاميرا على عصابة رأس أو قبعة أو زوج من النظارات، ويتم توصيلها بالحاسب عبر منافذ **يو إس بي** (USB) أو **البلوتوث** (Bluetooth).
- **فأرة التحكم بالقدم** (Foot - Mouse Control): تسمى أيضًا باسم **فأرة العمل بالقدم** (foot-operated mouse)، وهي جهاز يسمح للمُستخدِمين ذوي الإعاقات الحركية بالتحكم في حركات مؤشر فأرة الحاسب على الشاشة باستخدام القدم. يتكون هذا الجهاز من منصة صغيرة بدواستين يقوم المُستخدِم بتشغيلها للتحكم في حركة مؤشر الفأرة على الشاشة. تتصل هذه الفأرة بالحاسب عبر منافذ **يو إس بي** (USB) أو **البلوتوث** (Bluetooth).
- **التحكم عن طريق إشارات الدماغ** (Brain EEG control): تسمى أيضًا باسم **واجهة الحاسب والدماغ** (Brain-Computer interface-BCI)، وهي تقنيّة تُمكِّن المُستخدِمين من التحكم في الأجهزة أو التطبيقات باستخدام أنشطة الدماغ. يقوم هذا الجهاز بقياس الإشارات الكهربائية التي يولدها الدماغ، والمعروفة باسم إشارات **مخطط كهربية الدماغ** (Electroencephalography-EEG)، ثم ترجمتها إلى أوامر يمكن للحاسب فهمها.
- **التحكم بتتبُّع حركة العين** (Eye Tracking Control): يعرف أيضًا باسم **التحكم بنظرة العين** (Eye Gaze Control)، وهي تقنيّة تسمح للمُستخدِمين بالتحكم في الأجهزة أو التطبيقات من خلال تتبُّع حركات أعينهم وتعمل عن طريق استخدام كاميرا أو مستشعر مُتخصص لتتبُّع حركات عيني المُستخدِم، ثم تتم ترجمة تلك الحركات إلى أوامر يستطيع الحاسب فهمها.

قابلية الوصول إلى البرمجيات Software Accessibility

توفر أنظمة التشغيل الحديثة مثل نظام التشغيل **ويندوز** (Windows) **وماك أو أس** (macOS) خيارات لضبط طرق العرض لتوفير قابلية الوصول. تتضمن هذه الخيارات أدوات تتيح تكبير محتويات الشاشة وعكس الألوان، كما تقدم أدوات أخرى لمن يعانون من مشاكل في الرؤية تقوم بقراءة النصوص المعروضة على الشاشة (**تحويل النصّ إلى كلام** (Text-to-Speech))، ووصف الأشياء والنصوص على الشاشة بشكل أكثر دقة، بالإضافة إلى إمكانية استخدام الأوامر الصوتية لأداء المهام الأساسية في الحاسب.

جدول 4.3: إعدادات قابلية الوصول في أنظمة التشغيل المختلفة

نظام التشغيل	الإعدادات
ويندوز (Windows)	Settings → Ease of Access الإعدادات ← سهولة الوصول
ماك أو أس (macOS)	System Preferences → Accessibility تفضيلات النظام ← قابلية الوصول
أي أو أس (iOS)	Settings → General → Accessibility الإعدادات ← عام ← قابلية الوصول
أندرويد (Android)	Settings → Accessibility الإعدادات ← قابلية الوصول

قابلية الوصول إلى الويب Web Accessibility

يجب أن يكون محتوى مواقع الويب متاحًا للجميع بغض النظر عن طبيعة الأجهزة أو البرمجيات التي يستخدمونها، أو لغتهم أو مواقعهم أو قدراتهم، ويجب أن يكون ذلك المحتوى في متناول الأشخاص ذوي الاحتياجات الخاصة والإعاقات والقدرات المختلفة سواء كانت سمعية، أو حركيّة، أو بصريّة، أو معرفيّة.

خاصية قابلية الوصول تزيل الحواجز التي تواجه المُستخدِمين ذوي الإعاقات في الوصول إلى المحتوى وتصفح الويب، ولهذا السبب يجب تصميم مواقع الويب وتطبيقاتها المختلفة بحيث تأخذ في الاعتبار جميع الأشخاص مع مراعاة الشمولية في التصميم، وذلك لتسمح لهم باستخدام الويب بشكل فعَّال.

ما قابلية الوصول إلى الويب؟ What is Web Accessibility

صُمِّمت المواقع والأدوات والتقنيات لتمكين الأشخاص ذوي الإعاقة من استخدام هذه المواقع، وعلى وجه التحديد ليتم تمكين هؤلاء الأشخاص من تصفح الويب والتنقل والتفاعل من خلال شبكة الإنترنت.

تشمل قابلية الوصول إلى الويب جميع الإعاقات التي يمكنها أن تؤثر على قدرة الشخص على الوصول إلى الويب، بما في ذلك:

- صعوبات النطق (Speech Difficulties).
- الإعاقة الحركية (Physical Disability).
- الإعاقة السمعية (Hearing Disability).
- صعوبات الرؤية (Vision Difficulties).
- الصعوبات المعرفية (Cognitive Difficulties).

شكل 4.3: قابلية الوصول إلى الويب

اللغوية

الحركية

السمعية

البصرية

المعرفية

شكل 4.4: أنواع الإعاقات

يُعدُّ الوصول إلى الويب أمرًا ضروريًا لفئات أخرى من المُستخدِمين غير ذوي الاحتياجات الخاصة، وتشمل تلك الفئات كبار السن الذين يعانون من ضعف القدرات بسبب التقدم في العمر، وأولئك الذين يعانون من إعاقات دائمة أو مؤقتة بسبب التعرض لحوادث معينة مثل كسور الذراع وغيرها.

مبادئ تطوير موقع الويب من أجل قابلية الوصول
Principles of Website Development for Accessibility

تهدف قابلية الوصول إلى الويب إلى تلبية احتياجات كافة مُستخدِمي الموقع لاستخدامه حسب قدراتهم وحاجاتهم، ولذلك توجد بعض الشروط اللازمة لتحقيق هذا الهدف وهي:

وضوح المحتوى من خلال الاختيار السليم للألوان وزيادة التباين
Clarify Vision through Careful Color Selection and Increased Contrast

قد يجد الأشخاص الذين يعانون من إعاقات بصرية صعوبة في قراءة النصوص بدون خلفية عالية التباين، سواء كانت خلفية عادية أو نصًا مُدمجًا في صورة.

عدم الاعتماد على الألوان فقط لتوضيح المعلومات
Not Relying Only on Colors to Clarify Information

يُعدُّ استخدام التصميمات التي يقتصر اعتمادها على تمييز الألوان غير كافيًا لتوفير المحتوى للأشخاص الذين لا يستطيعون التمييز بين الألوان المختلفة. على سبيل المثال، يوفِّر موضع الضوء النشط في إشارة المرور المعلومات اللازمة للأفراد المصابين بعمى الألوان عندما يتعلق الأمر بالتوقف أو التقدم أثناء القيادة. يجب على المصممين استخدام أكثر من طريقة للتعبير عن المعنى المقصود بالتصميم.

التصفح باستخدام لوحة المفاتيح Browse Using the Keyboard

تُستخدَم الفأرة بشكلٍ واسع خلال تصفح المُستخدِم للويب، ولكن قد يصعُب استخدامها في بعض الأحيان، وبالتالي يجب أن توفّر لوحة المفاتيح خيارات للتنقل في صفحة الويب تناسب المُستخدِمين ذوي القدرة المحدودة على الحركة، كما يمكن استخدام طرق خاصة في تصميم الروابط، مثل إبرازها بالألوان، وتمييز التصميمات لحالات مختلفة، مثل الضغط والتمرير وغيرها؛ لتمكين المُستخدِمين من التنقل في المواقع باستخدام لوحة المفاتيح.

توفير تسمية مناسبة لكل حقل Provide the Correct Naming of the Fields

قد لا يتمكن بعض المُستخدِمين ذوي الإعاقات المعرفية من فهم معنى حقول النموذج، ولذلك يجب على المطوِّر توفير تسمية وصفية لجميع حقول النموذج.

التنوّع عند عرض التغذية الراجعة للأخطاء Variety of Feedback for Errors

يجب تنبيه مُستخدِمي موقع الويب عند ارتكابهم الأخطاء أثناء عملهم وذلك بطرقٍ متعددة من خلال استخدام النصوص، والأيقونات، والألوان بشكلٍ واضح وفوري. يمكن للمصمِّمين استخدام الألوان المختلفة حسب نوع التنبيه المطلوب للمستخدم.

وزارة التعليم
Ministry of Education
2023 - 1445

توفير بدائل عديدة للوسائط المُستخدَمة
Providing Several Alternatives for the Media Used

توفِّر الوسائط المختلفة في المواقع مثل: الصور، والصوت، والنصوص، والفيديو إمكانيات متكافئة للوصول إلى المعلومات للمُستخدِمين من ذوي الإعاقات المختلفة. النصوص المدمَجة بالصور والأصوات تجعل المحتوى أكثر جاذبية للمُستخدِمين الذين يعانون من إعاقة سمعية أو بصرية. من الجيد توفير نسخة نصية تحتوي على المعلومات الصوتية لمساعدة الأشخاص الصُم أو ضعاف السمع في فهم المحتوى، وينطبق الأمر نفسه على محركات البحث والتقنيات الأخرى.

كتابة نص بديل مفيد للصور والمحتويات الأخرى غير النصية
Write Useful Alt Text for Images and Other Non-Text Content

يستفيد الأشخاص ضعاف البصر من برامج قراءة الشاشة للحصول على المعلومات من خلال صفحات الويب بشكلٍ مسموع، حيث تحوِّل هذه الأدوات النصّ إلى كلام مسموع، عند توفر نصوص بديلة للصورة، يصف **القارئ التلقائي** (Auto Reader) الصورة للمُستخدِم وكذلك لمحركات البحث والتقنيات الأخرى باستخدام النصّ البديل بدلًا من مجرد الإشارة إلى وجود الصورة.

التصميم المرتكز على قابلية الاستخدام Usability-Focused Design

يرتكز التصميم الذي يتمحور على قابلية الاستخدام على نهج تصميم يضع المُستخدِم في قمة اعتباراته، حيث تكون احتياجات المُستخدِم وسلوكياته وتوقُّعاته مركزية في عملية التصميم. يتضمن هذا النهج تعاونًا وثيقًا بين المصمِّمين والمطوِّرين وأصحاب المصلحة لضمان تلبية البرمجيات لاحتياجات الجمهور المستهدَف. تتضمن عملية التصميم التي تركز على قابلية الاستخدام القيام بأبحاث المُستخدِمين وتصميم النماذج الأولية وإجراء الاختبار وتكرار الخطوات السابقة عند اللزوم. تتضمن أبحاث المُستخدِمين القيام باستطلاعات الرأي، أو بمجموعات مُركزة، أو إجراء اختبارات قابلية الاستخدام؛ لتحقيق الفهم الأفضل لاحتياجات المُستخدِم وأهدافه والمشاكل المُحتملة.

أمثلة على تطبيقات بقابلية الوصول Examples of Accessible Applications

تم إنشاء تطبيقات مختلفة بمعايير معينة لتكون متاحةً للمُستخدِمين الذين يعانون من أنواع مختلفة من الإعاقات الدائمة أو المؤقتة، وفيما يلي بعض الأمثلة:

آكسس ناو AccessNow

يوفر تطبيق آكسس ناو معلومات قابلية الوصول عن أماكن مختلفة حول العالم. يُمكن للمُستخدِم البحث في هذا التطبيق عن أنشطة تجارية أو أماكن معينة كالمطاعم والفنادق والمتاجر، أو تصفح خريطة لرؤية ميزات قابلية الوصول القريبة التي يحتاجها الشخص. على سبيل المثال، يُمكن لأي شخص يَستخدِم كرسيًا متحركًا الحصول على قائمة بالمطاعم المتاحة لمُستخدِمي الكراسي المتحركة في منطقة محدَّدة. إذا لم تكن المَعلومات موجودة بالفعل على الخريطة، يسمح التطبيق للمُستخدِمين بإضافتها والمساهمة في خدمة المجتمع في جميع أنحاء العالم.

روجر فويس RogerVoice

تم تصميم تطبيق روجر فويس لمساعدة الصُم على التواصل عبر الهاتف، حيث يسمح هذا التطبيق للصمّ، وذوي الإعاقة السمعيّة بأن يشاركوا في المحادثات والنقاشات باستخدام تقنيّة التعرف على الصوت، حيث يحوِّل الأصوات إلى نص مكتوب يمكن للشخص الأصمّ قراءته، مما يتيح إجراء محادثة ونقاش بين الطرفين بشكل فعَّال.

إنفجن إي آي Envision AI

يَستخدِم تطبيق إنفجن إي آي الكاميرا لتقديم وصف سمعي لما يحدث حول الشخص. على سبيل المثال: يمكنك توجيه هاتفك نحو رفيقك الجالس أمامك، وسيقوم التطبيق بإعلامك بوجود شخص يجلس هناك وسيصف لك البيئة المحيطة به. يمكن للتطبيق أيضًا قراءة المستندات والتعرف على الكتابة اليدوية ومسح **الرموز الشريطية** (Barcodes). يدعم البرنامج 60 لغة، ويمكن تدريبه للتعرُّف على صور أفراد عائلتك، وسيتذكرها في المستقبل.

تمرينات

1

حدّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. الفجوة الرقميّة هي مشكلة اقتصادية بحتة.	○	○
2. ليس للقيود الجغرافية أثر في تفاقم الفجوة الرقميّة.	○	○
3. تؤثر الفجوة الرقميّة على كل من الأميين والأشخاص ذوي الإعاقة.	○	○
4. تم تخصيص جميع التطبيقات التقَنيّة للغات المحلية.	○	○
5. يسهم ارتفاع تكلفة مكونات الأجهزة في تفاقم الفجوة الرقميّة.	○	○
6. يساعد توفير الوصول السريع للإنترنت في المناطق النائية على مكافحة الفجوة الرقميّة.	○	○
7. يساعد تقليص الفجوة الرقميّة في إنشاء المزيد من الأسواق للشركات.	○	○
8. يُعدُّ الوصول إلى البرمجيات الأمر الوحيد المطلوب لتقليص الفجوة الرقميّة للأشخاص ذوي الإعاقة.	○	○
9. لا توجد إرشادات قياسية لتصميم مواقع الويب التي يمكن الوصول إليها.	○	○
10. يتم استخدام التصميم المرتكز على قابلية الاستخدام فقط للأشخاص الذين يعانون من إعاقات بصرية.	○	○

2 صِفْ المقصود بالفجوة الرقميّة.

3 اشرح كيف يساعد خفض تكلفة المكونات التقَنيّة في سد الفجوة الرقميّة.

4 صَنِّف المزايا المجتمعية الرئيسة لتقليص الفجوة الرقميّة.

5 اشرح كيف يمكنك تطوير الأجهزة لتكون في متناول الأشخاص ذوي الإعاقة.

6 صِفْ ثلاثة مبادئ لتطوير موقع الويب من أجل قابلية الوصول.

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الثالث

ميزات قابلية الوصول في التطبيق

ملاءمة التطبيق للأشخاص ذوي الاحتياجات الخاصة

The Suitability of the Application for People with Special Needs

لا يمتلك كافة المُستخدِمين الاحتياجات نفسها، ولهذا السبب يجب أن تأخذ التطبيقات هذه الاختلافات بالاعتبار، ويجب تعديل واجهة المستخدِم ووظائفها بناءً عليها.

ستعمل على تحسين التطبيق الذي أنشأته في الوحدة السابقة لمساعدة كبار السن على التنقل من خلال شاشة التطبيق، وذلك ليتمكنوا من قراءة المعلومات حول المواقع السياحية المختلفة التي يمكنهم زيارتها في المملكة العربية السعودية.

نظرًا لأن كثيرًا من كبار السن يعانون من مشاكل في الرؤية، ستوفّر لهم القدرة على ضبط حجم النصّ ليتمكنوا من قراءته بسهولة، وقد يعاني بعضهم من مشاكل أخرى كارتعاش الأيدي، وبالتالي يصعب عليهم الضغط على أزرار الشاشة، ولهذا عليكَ أن توفّر لهم إمكانية ضبط حجم أزرار التطبيق حسب ما يناسبهم.

إن جعل تطبيقك في متناول الأشخاص الذين يعانون من ضعف الرؤية والصعوبات البصرية لا يتطلب بالضرورة قدرًا كبيرًا من العمل، ولكن الأمر المهم هنا هو أخذ احتياجات المُستخدِمين المختلفة بعين الاعتبار.

لتطوير التطبيق ليتكيف مع احتياجات المُستخدِمين الذين لديهم صعوبة في الرؤية، يجب تحسين التطبيق الأولي بإضافة الميزات التالية:

وظيفة التكبير والتصغير Zoom In and Out Function

يمكنك إضافة **خاصية التكبير** (Zoom In) **وخاصية التصغير** (Zoom Out) بحيث يُمكن للمُستخدِم ضبط حجم النصّ حسب احتياجاته.

واجهة ملونة Colored Interface

يمكنك إضافة خيار تبديل الواجهة ما بين ملونة أو بالأبيض والأسود فقط، من أجل تسهيل القراءة للمُستخدِمين الأكبر سنًا أو الأشخاص ذوي الإعاقة وكذلك للمُستخدِمين عمومًا وبشكل خاص على الأجهزة المحمولة.

تحويل النصّ إلى كلام Text to Speech

من المهم أن يدعم التطبيق أكبر قدر ممكن من التفاعل مع حواس الإنسان، لكي يتمكن المُستخدِمون من فهم المعلومات الموجودة في الوسائط المختلفة واستيعابها، مثل: الصور، والصوت، والفيديو، والرسوم المتحركة، والعروض التقديمية. لذلك ستعمل على تعديل تطبيقك لدعم حاسة أخرى غير البصر، من خلال إضافة خيار للمُستخدِم يستطيع من خلاله الاستماع إلى المعلومات إذا كان غير قادر على قراءتها لأي سبب من الأسباب.

تعديل النموذج الأولي لتحسين قابلية الوصول للتطبيق

Adjusting the Prototype to Improve the Accessibility of the App

قبل البدء بإجراء التغييرات المناسبة على تطبيق الهاتف المحمول في **مخترع التطبيقات** (App Inventor)، يجب إجراء التغييرات على النموذج الأولي الذي أنشأته باستخدام **بنسل بروجكت** (Pencil Project).

افتح **بنسل بروجكت** (Pencil Project) لإعادة تصميم الشاشة الأخيرة من التطبيق.

ستُضيف أربع صور كما تعلمت سابقًا، وستُنشئ الشاشة التي تعرض مَعْلَم المَصمك كما هو موضَّح أدناه:

شكل 4.5: تعديل التّصميم لتحسين قابلية الوصول في بينسل بروجكت

تحسين واجهة المُستخدِم بميزات قابلية الوصول
Enhancing the UI with Accessibility Features

ستعمل الآن على تحسين شاشة مَعْلَم Al Masmak (المصمك) بميزات قابلية الوصول، حيث ستضيف أزرارًا لتكبير حجم النصّ أو تصغيره، وزرًّا لتغيير نَسَق (Theme) لون الشاشة، وزرًّا لإضافة وظيفة تحويل النصّ إلى كلام على الشاشة. ستُضيف أولًا المكونات إلى قسم Designer (المُصمِّم)، ثم تُبرمِج وظائفها من قسم Blocks (اللبنات البرمجية).

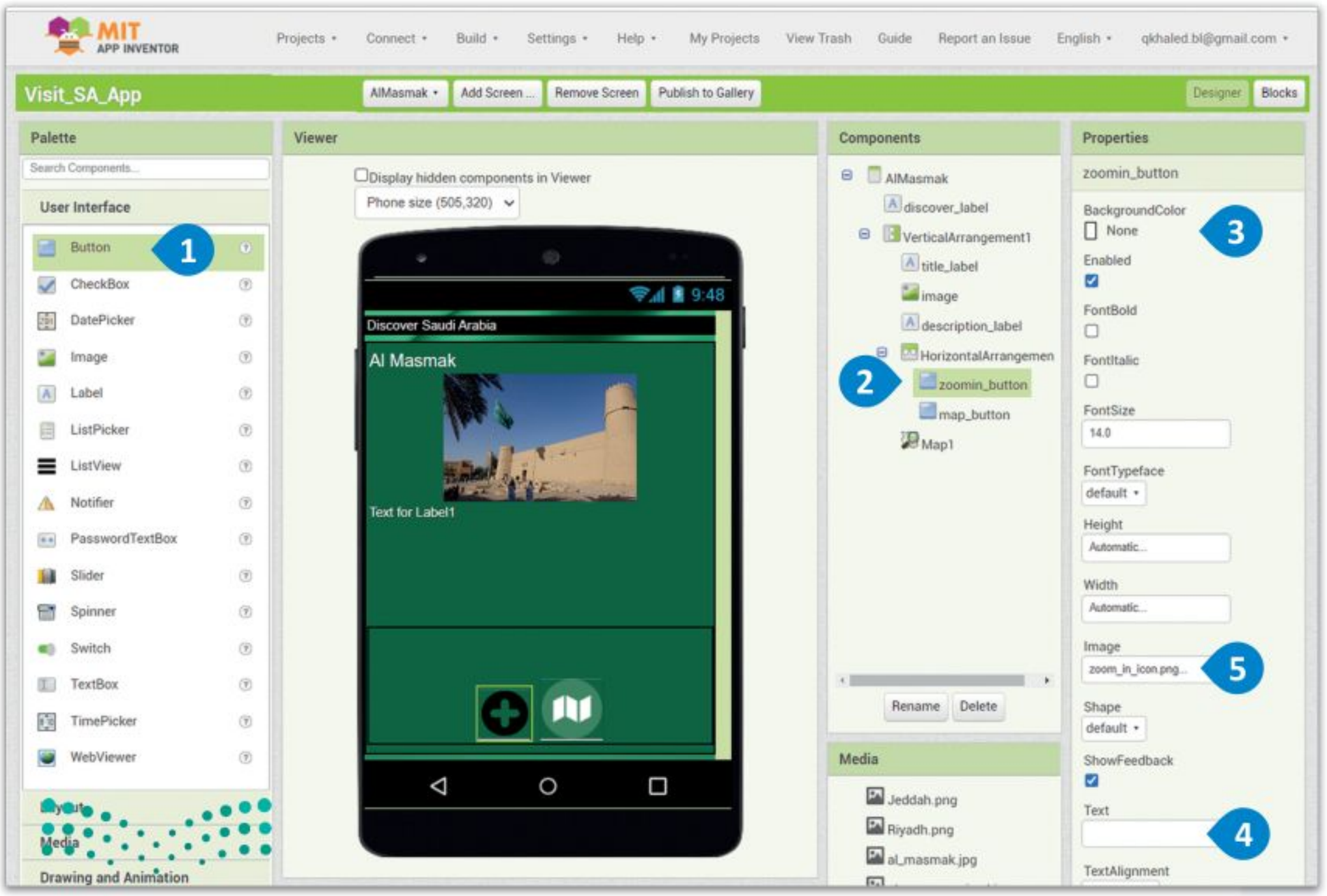
إضافة زر تكبير للتطبيق Adding a Zoom In Button to the Application

ستُضيف الآن زرًّا لتكبير حجم الخط لجميع مكونات النصّ في كل مرة تضغط عليه.

لإضافة زرّ التكبير:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Button (زرّ) إلى Screen1 (شاشة 1)، (1) وأعد تسميته إلى zoomin_button (زرّ_التكبير). (2)

> من المُكوِّن zoomin_button (زرّ_التكبير)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (3) ثم امسح خاصية Text (النصّ)، (4) واضبط خاصية Image (الصورة) إلى أيقونة علامة زائد. (5)



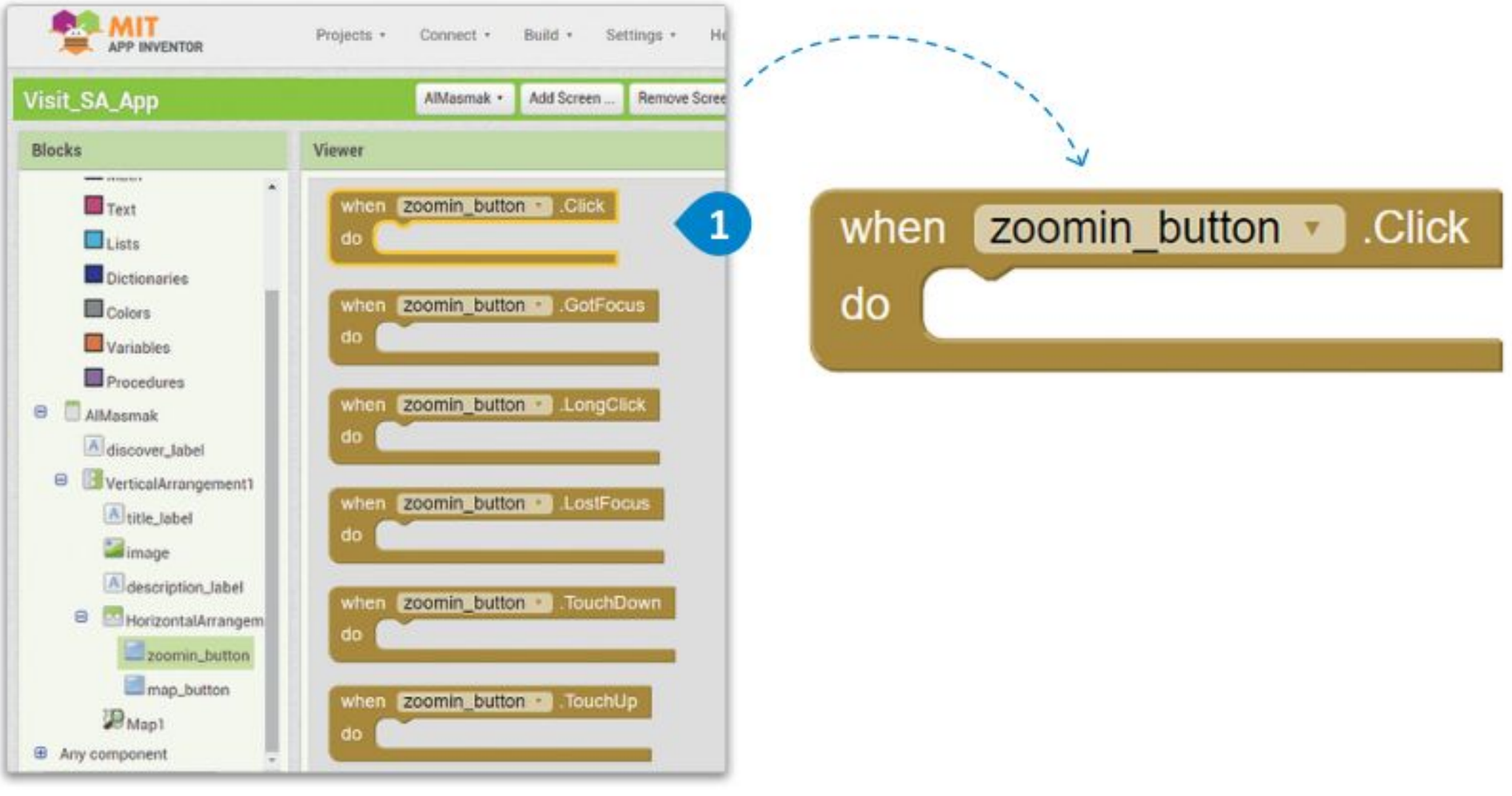
شكل 4.6: إضافة زرّ التكبير

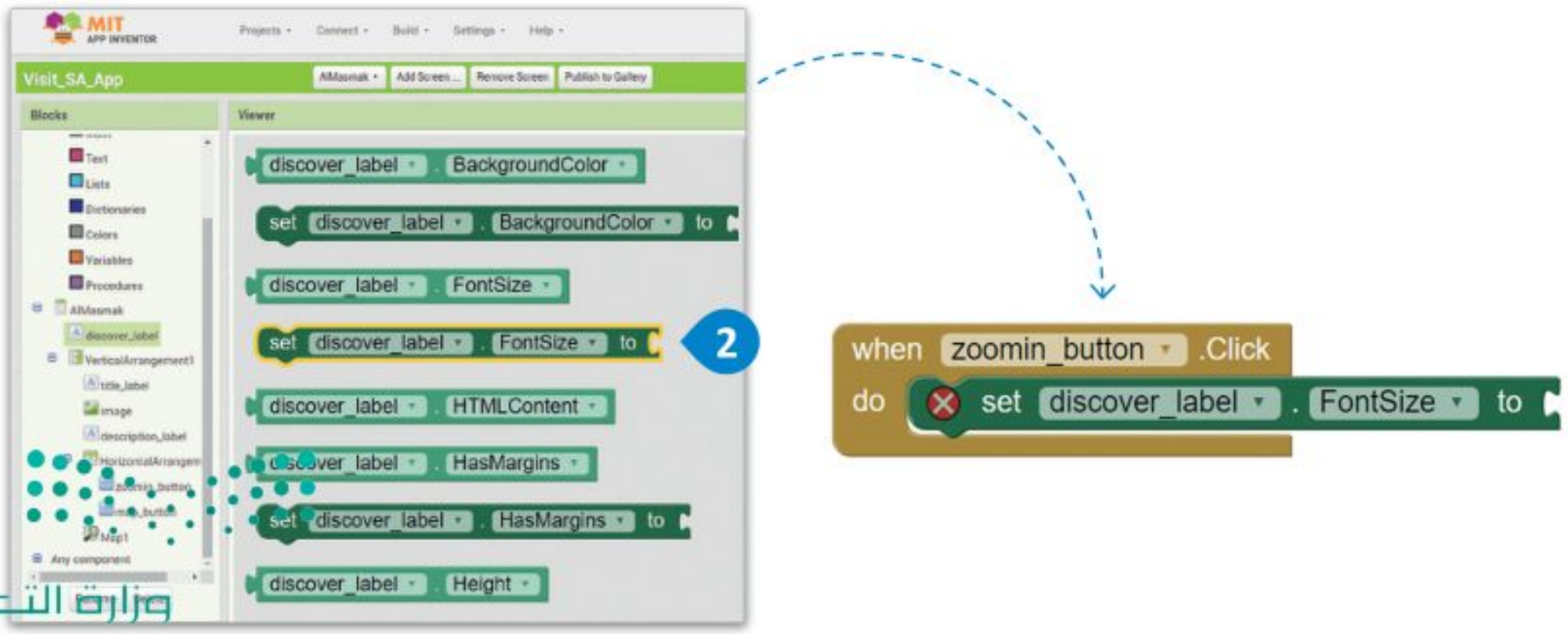
برمجة زرّ التكبير Programming the Zoom In Button

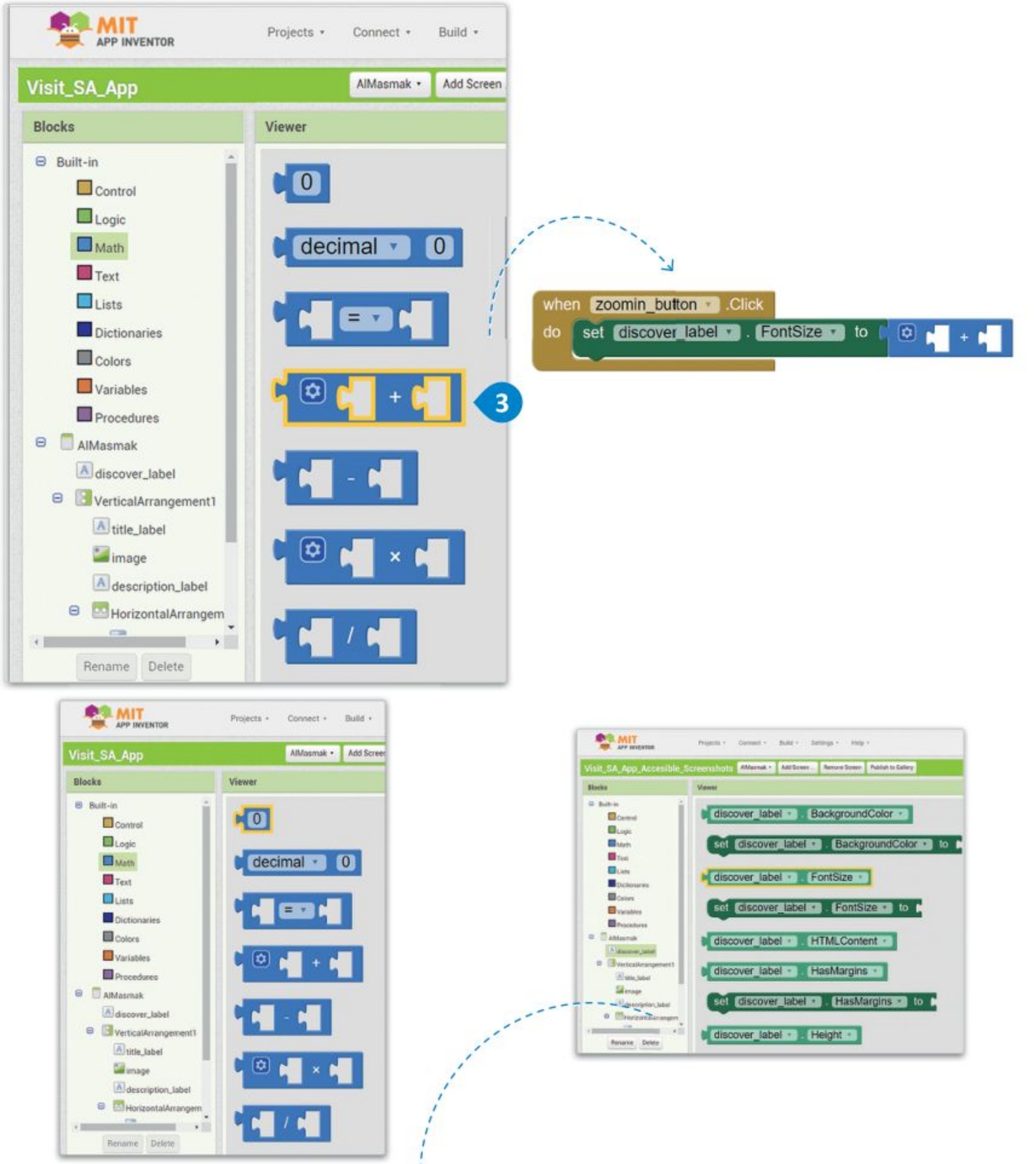
سيقوم هذا الزر بزيادة القيمة الحالية لخاصية FontSize (حجم الخط) لكافة مكونات النصّ بمقدار نقطة واحدة في كل مرة يتم الضغط عليه.

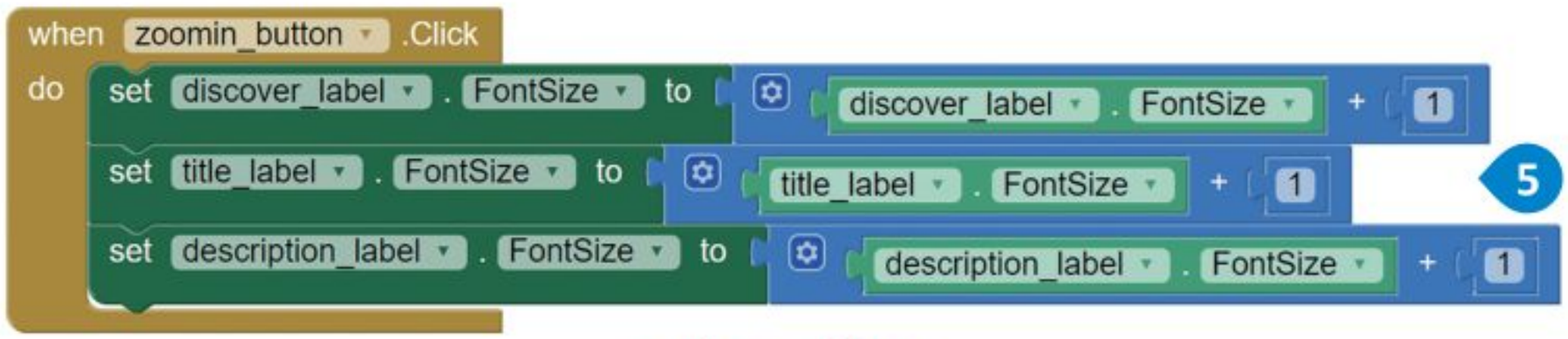
لبرمجة زرّ التكبير:

> حدِّد حدث Click (الضغط) لمُكوِّن zoomin_button (زرّ_التكبير). 1
> حدِّد الأمر Set FontSize (ضبط حجم الخط) للمُكوِّن discover_label (اكتشف_ التسمية). 2
> حدِّد لبنة Addition (الجمع) من مجموعة أوامر Math (الحساب). 3
> أضف القيمة 1 إلى خاصية FontSize (حجم الخط) الموجودة لمُكوِّن discover_label (اكتشف_ التسمية). 4
> كرِّر العملية نفسها لمكونات النصّ المتبقية على الشاشة. 5






MIT APP INVENTOR
Projects
Connect
Build
Visit_SA_App
AlMasmak
Add Screen
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
AlMasmak
discover_label
VerticalArrangement1
title_label
image
description_label
HorizontalArrangem
Rename
Delete
0
decimal 0
=
+
3
-
×
/
when zoomin_button .Click
do set discover_label . FontSize to + 1
4



شكل 4.7: برمجة زرّ التكبير

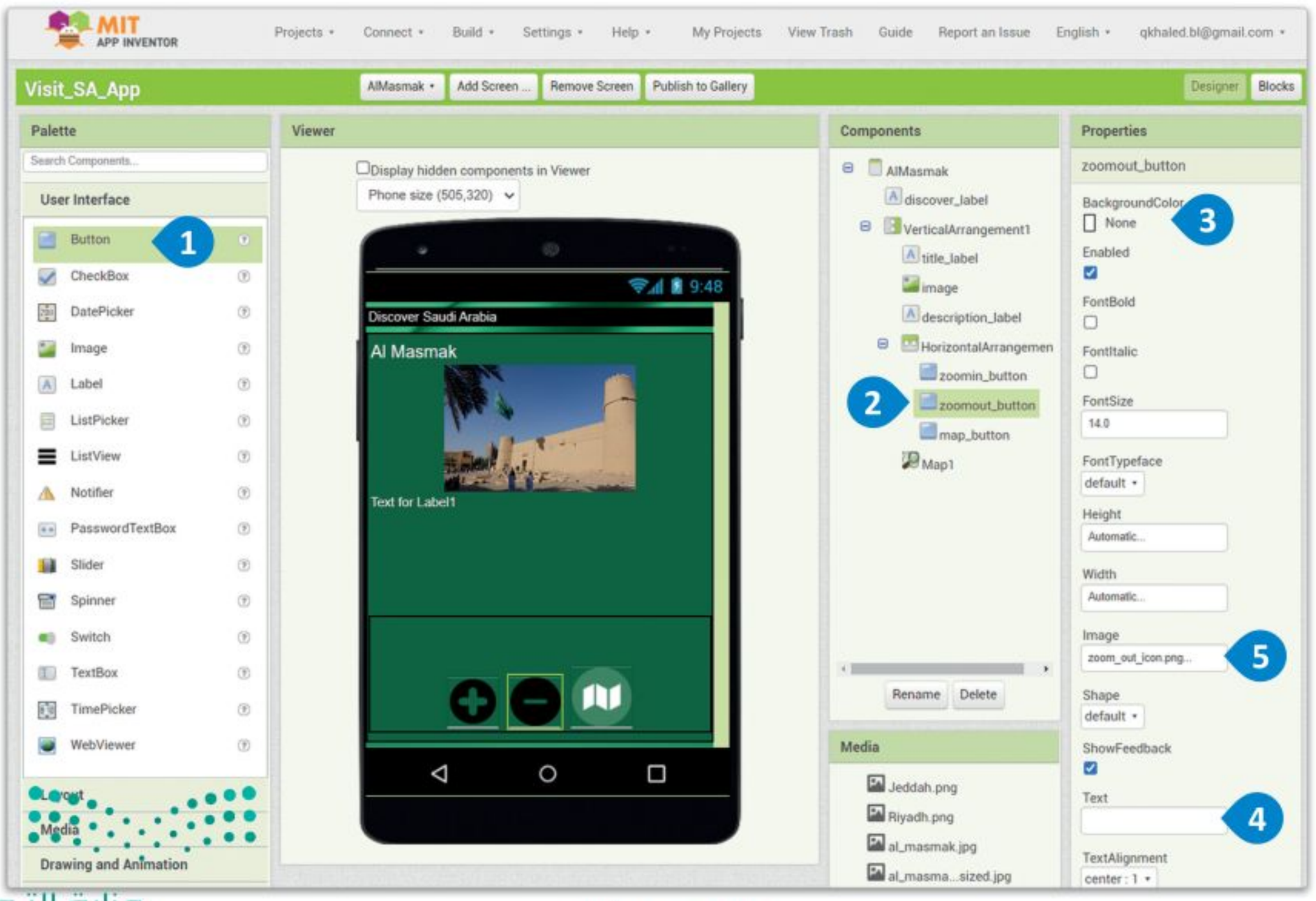
إضافة زر التصغير للتطبيق Adding a Zoom Out Button to the Application

ستُضيف الآن زرًّا لتصغير حجم الخط لجميع مكونات النصّ في كل مرة يتم الضغط عليه.

لإضافة زرّ التصغير:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Button (زرّ) إلى Screen1 (شاشة 1)، (1) وأعد تسميته إلى Zoomout_button (زرّ _ التصغير). (2)

> من المكوِّن Zoomout_button (زرّ _ التصغير)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (3) ثم امسح خاصية Text (النصّ)، (4) واضبط خاصية Image (صورة) إلى أيقونة علامة ناقص. (5)



شكل 4.8: إضافة زرّ التصغير

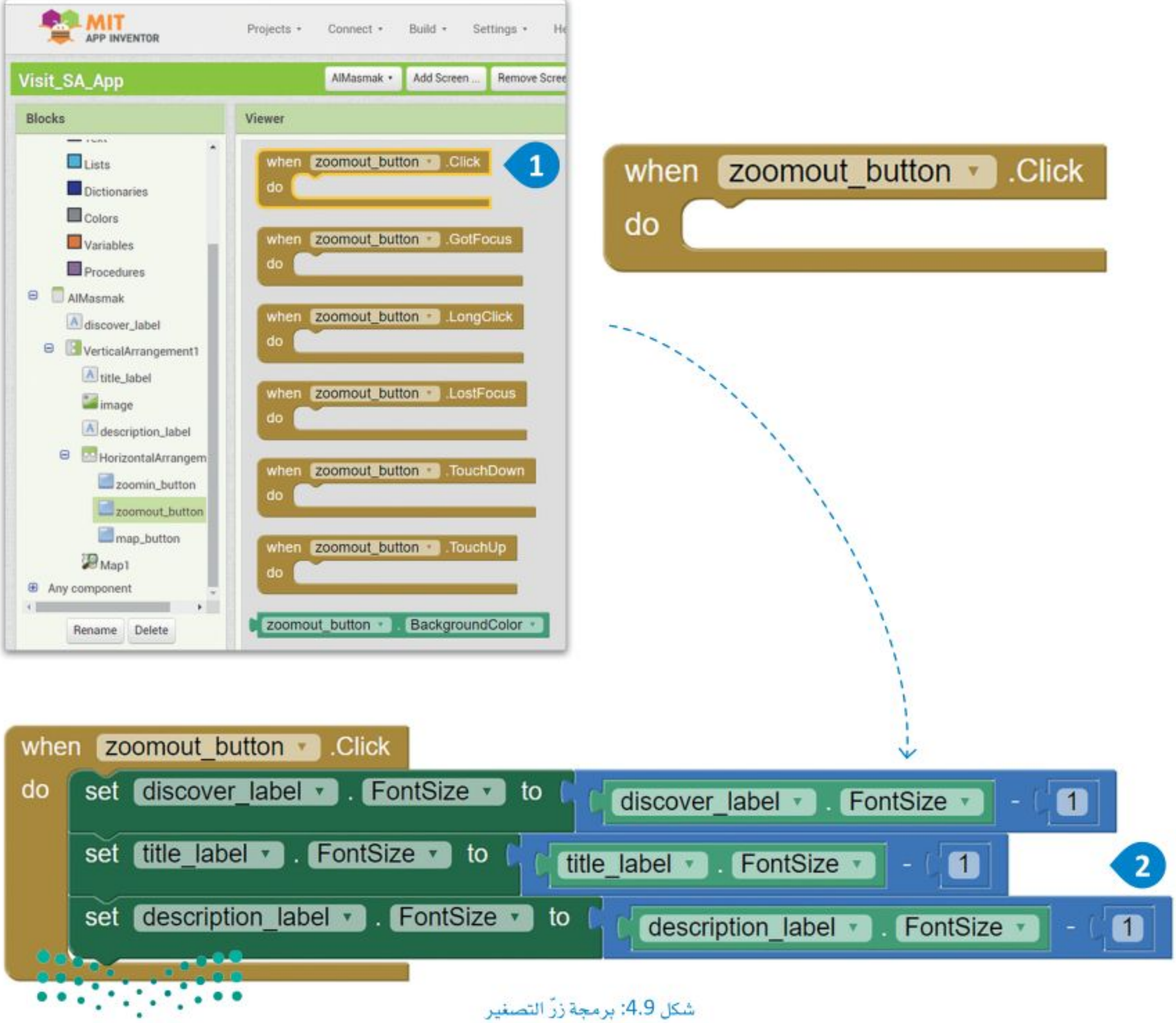
برمجة زر التصغير Programming the Zoom Out Button

سيؤدي هذا الزر إلى تقليل القيمة الحالية لخاصية FontSize (حجم الخط) لكافة مُكوِّنات النصّ بمقدار نقطة واحدة في كل مرة يتم الضغط عليه.

لبرمجة زرّ التصغير:

> حدِّد حدث Click (الضغط) لمُكوِّن Zoomout_button (زرّ التصغير). 1

> كرِّر نفس العملية السابقه لمُكوِّن zoomin_button (زرّ التكبير)، وغيِّر العملية الحسابية فقط من addition (الجمع) إلى subtraction (الطرح). 2



شكل 4.9: برمجة زرّ التصغير

إضافة زر تحويل النصّ إلى كلام للتطبيق

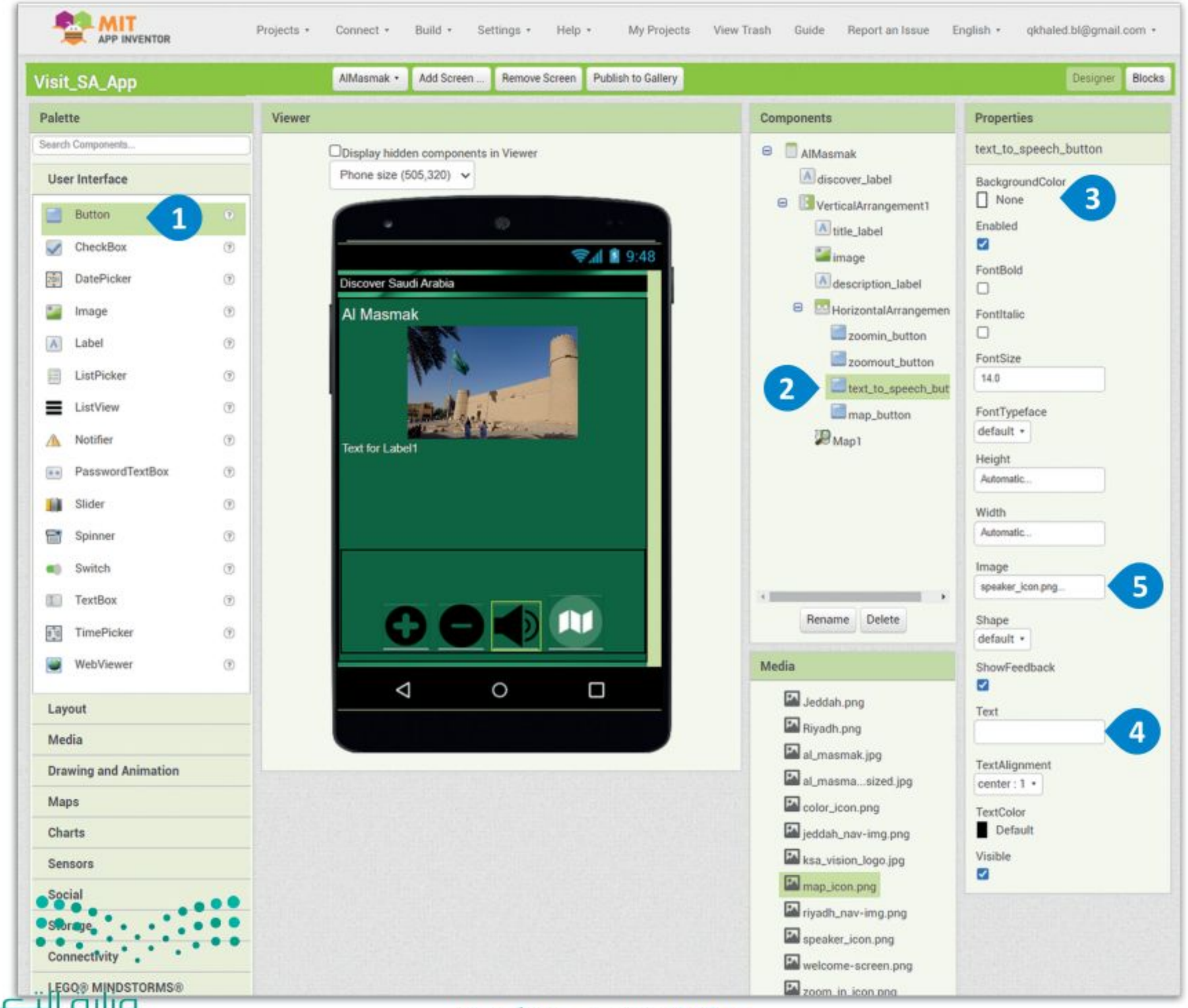
Adding a Text-To-Speech Button to the Application

ستُضيف الآن زرًّا يقوم بتنشيط مُكوِّن TextToSpeech (تحويل النصّ إلى كلام) الذي يَستخدِم نظام صوت جهاز الهاتف المحمول لقراءة نص مُحدَّد بصوت عالٍ.

لإضافة زر تحويل النصّ إلى كلام :

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Button (زرّ) إلى Screen1 (شاشة 1)، (1) وأعد تسميته إلى text_to_speech_button (زرّ_تحويل النصّ_إلى_كلام). (2)

> من مُكوِّن text_to_speech_button (زرّ_تحويل النصّ_إلى_كلام)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، (3) ثم امسح خاصية Text (النصّ)، (4) واضبط خاصية Image (الصورة) إلى أيقونة مكبر صوت. (5)

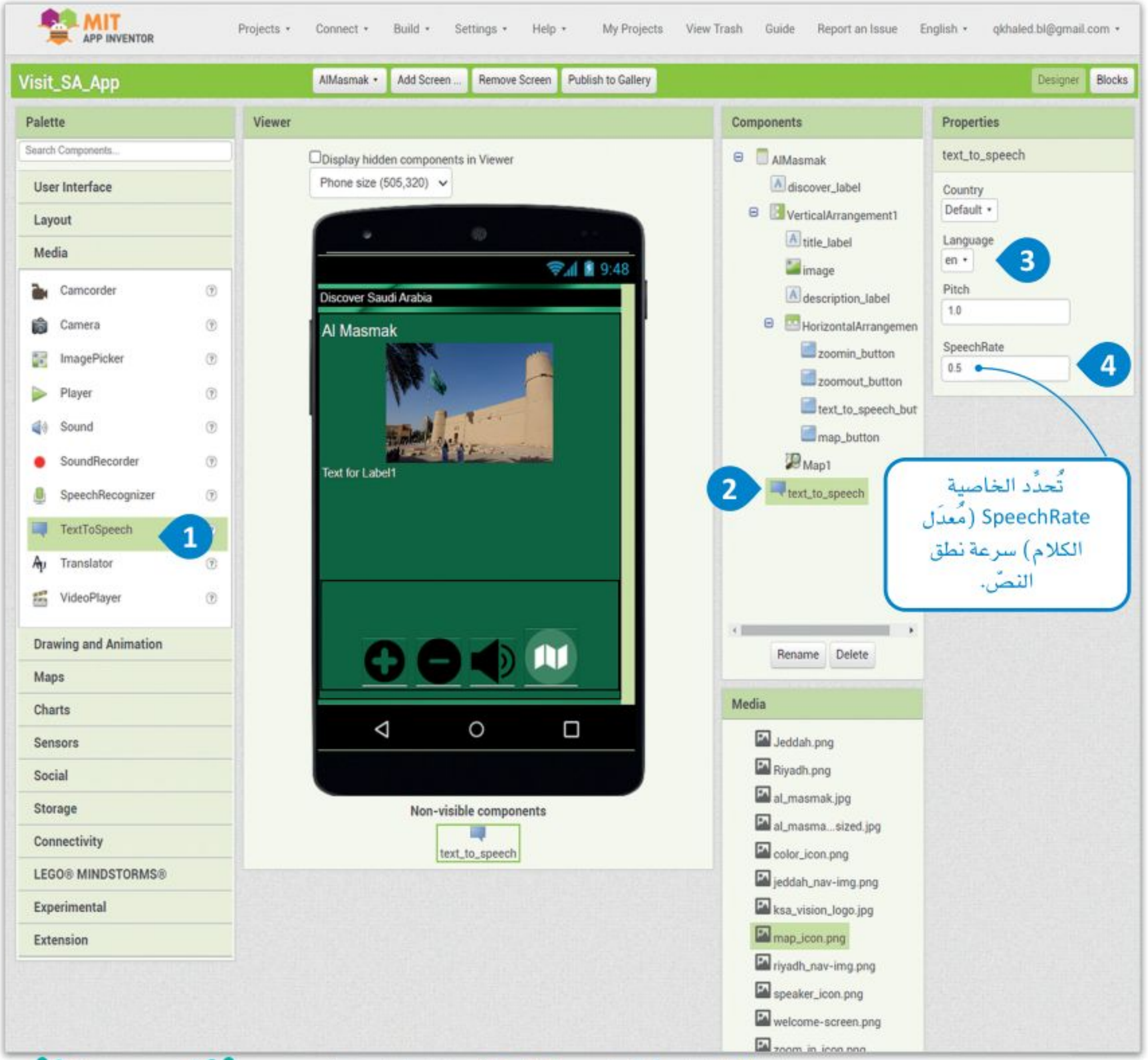


شكل 4.10: إضافة زر تحويل النصّ إلى كلام

لإضافة مُكوِّن TextToSpeech (تحويل النصّ إلى كلام):

> من مجموعة Media (الوسائط)، أضف مُكوِّن TextToSpeech (تحويل النصّ إلى كلام) إلى الشاشة، 1 وأعد تسميته إلى text_to_speech (تحويل النصّ _إلى _كلام). 2

> في مُكوِّن text_to_speech (تحويل النصّ _إلى _كلام)، اضبط خاصية Language (اللغة) إلى en (الإنجليزية)، 3 ثم اضبط خاصية SpeechRate (مُعدَل الكلام) إلى 0.5. 4



شكل 4.11: إضافة مُكوِّن TextToSpeech (تحويل النصّ إلى كلام)

برمجة زر تحويل النصّ إلى كلام

Programming the Text-To-Speech Button

سيُنشئ هذا الزر نصًا من محتويات جميع مكونات النصّ على الشاشة والتي سيتمّ توفيرها لمُكوِّن TextToSpeech (تحويل النصّ إلى كلام) لإنتاج الرسالة المنطوقة.

لبرمجة زرّ تحويل النصّ إلى كلام:

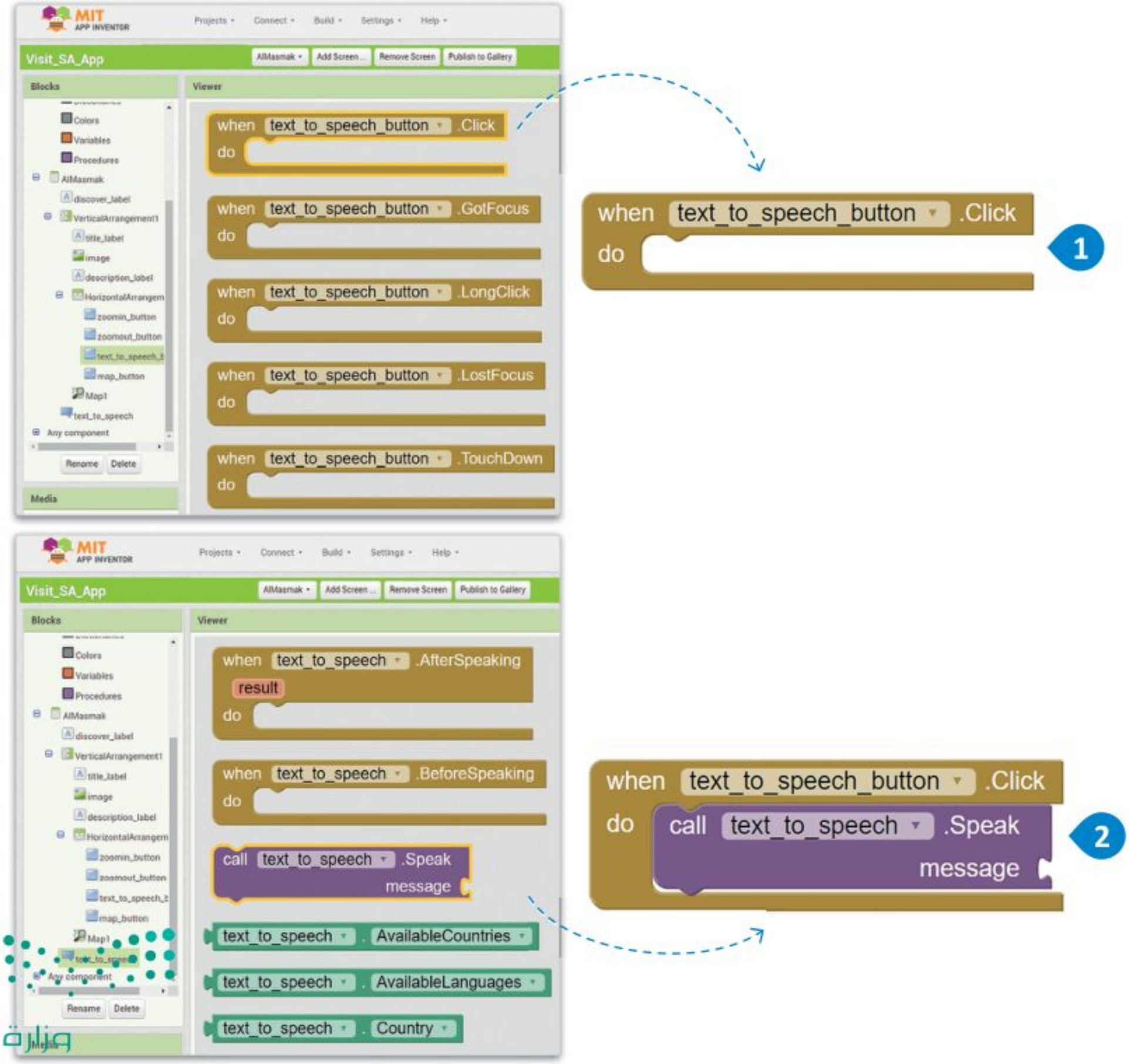
> حدِّد حدث Click (الضغط) لمُكوِّن text_to_speech_button (زرّ_ تحويل النصّ _إلى _كلام). 1

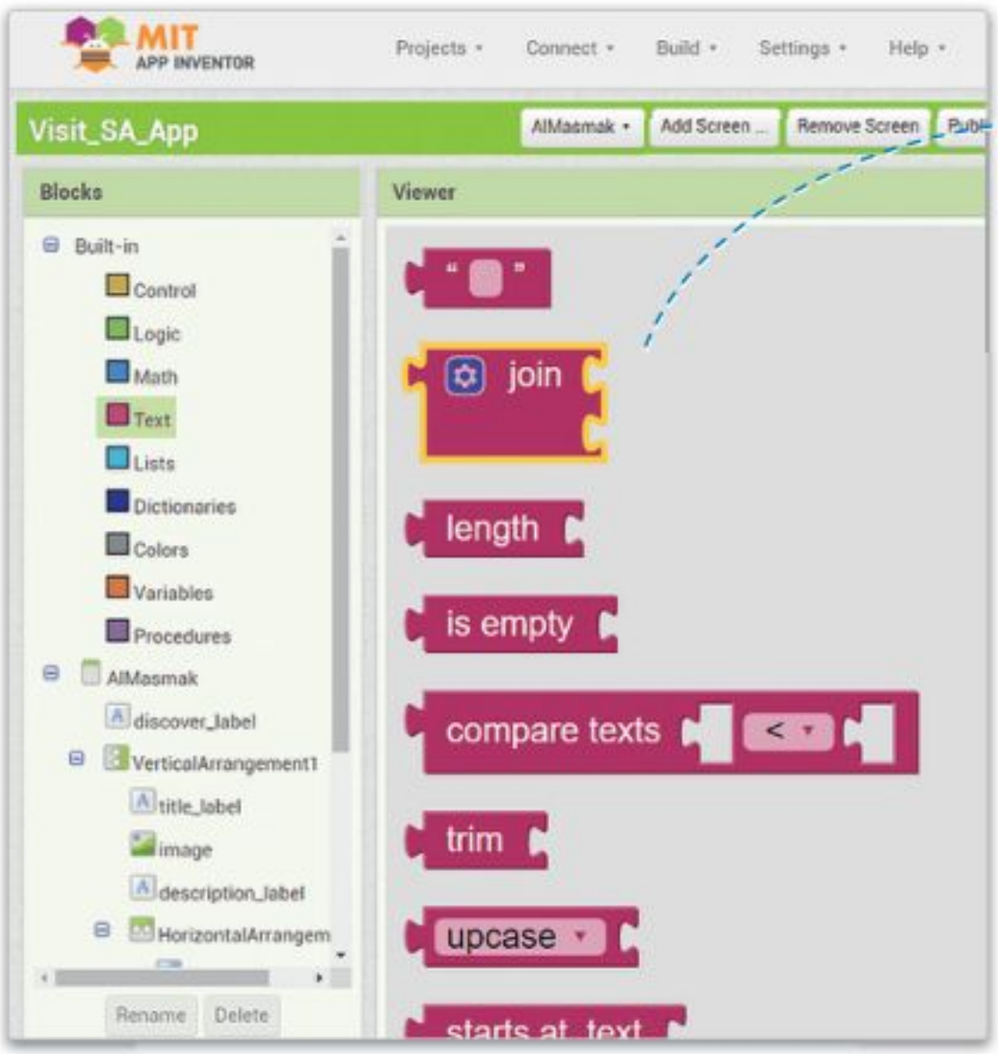
> حدِّد الأمر Speak message (رسالة صوتية) لمُكوِّن text_to_speech (تحويل النصّ _إلى _كلام). 2

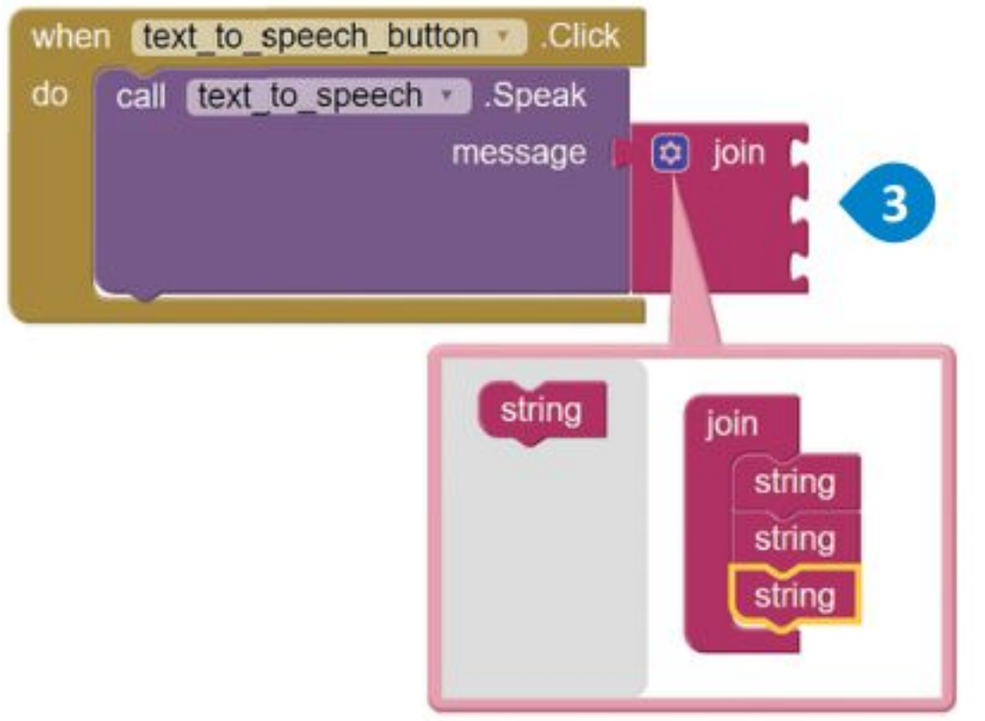
> حدِّد الأمر Join (ربط) من مجموعة Text (النصّ) لربط النصوص معًا. 3

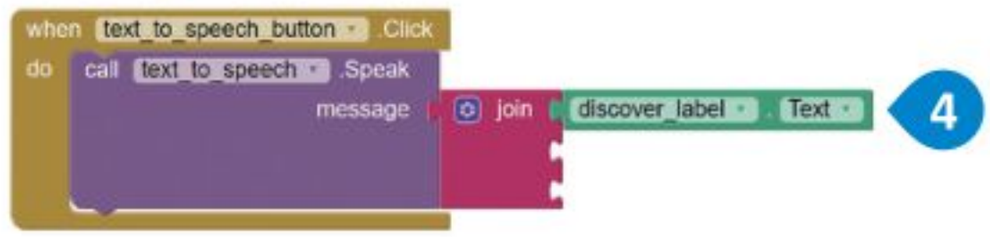
> أضف خاصية Text (النصّ) لمُكوِّن Discover_label (اكتشف_ التسمية) إلى المُدخَل الأول لأمر Join (الربط). 4

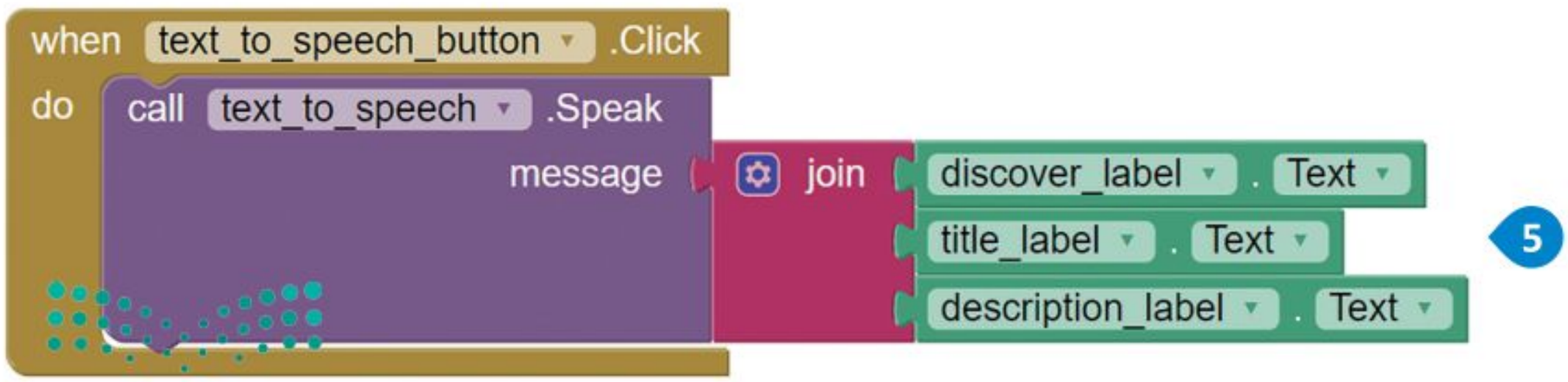
> كرِّر العملية المذكورة أعلاه للمكونات النصية الأخرى. 5











شكل 4.12: برمجة وظيفة تحويل النصّ إلى كلام

إضافة زرّ لتبديل النَّسَق اللوني للتطبيق

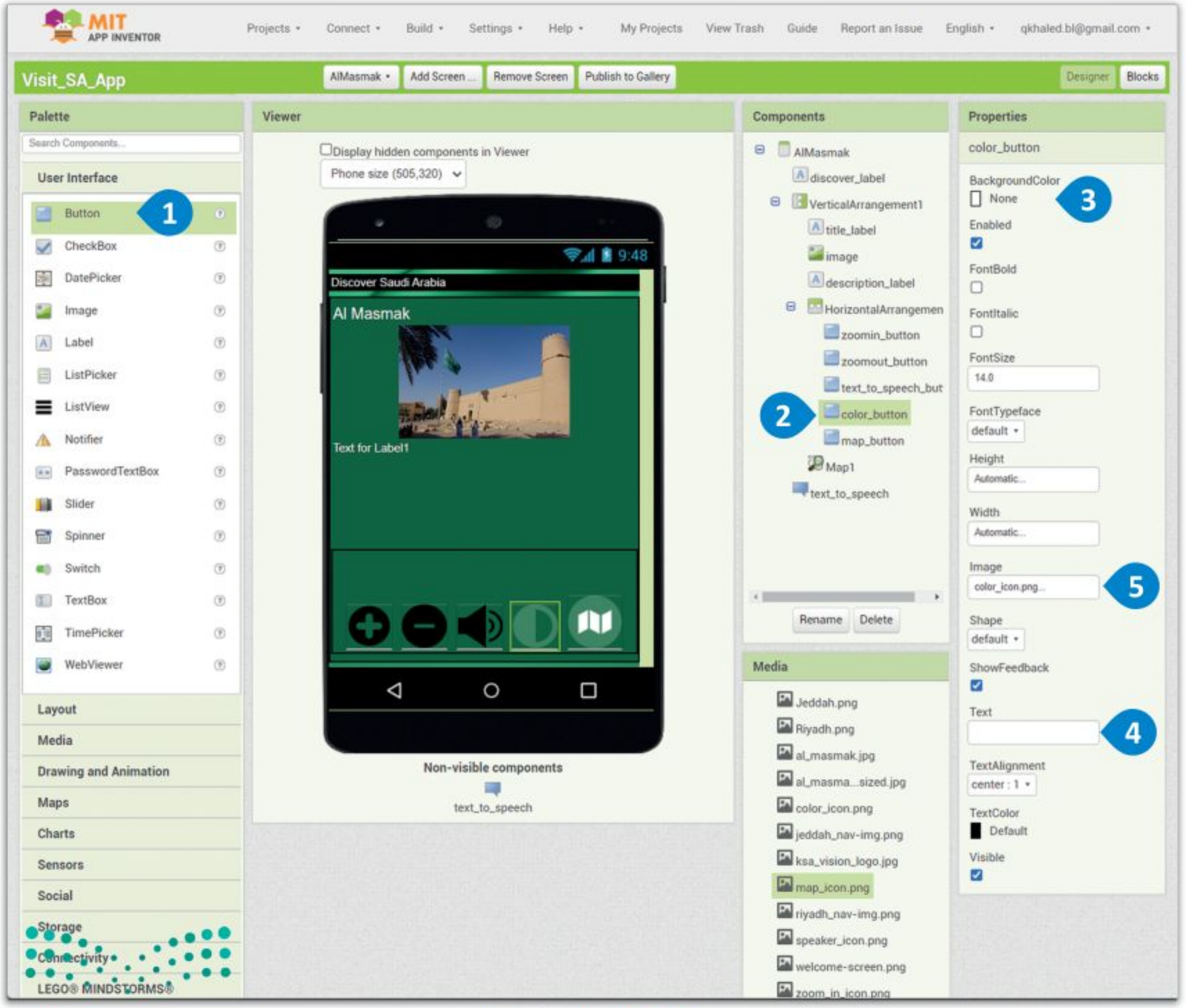
Adding a Button to Change the Color Theme of the Application

ستُضيف الآن زرًّا يُغيِّر نَسَق الشاشة بين نَسَقٍ فاتح وآخر أخضر.

لإضافة زرّ نَسَق اللون:

> من مجموعة User Interface (واجهة المُستخدِم)، أضف مُكوِّن Button (زرّ) إلى Screen1 (شاشة 1)، 1 وأعد تسميته إلى color_button (زرّ_ اللون). 2

> في مُكوِّن color_button (زرّ_ اللون)، اضبط خاصية BackgroundColor (لون الخلفية) إلى None (بدون)، 3 ثم امسح خاصية Text (النصّ)، 4 واضبط خاصية Image (صورة) إلى أيقونة نصف دائرة. 5



شكل 4.13: إضافة زرّ نَسَق اللون

برمجة زرّ نَسَق اللون Programming the Color Theme Button

عند ضغط هذا الزر ستُصبح خلفية الشاشة بيضاء وسيصبح لون النصّ أسود. وعند ضغطه لمدة طويلة، ستُصبح خلفية الشاشة خضراء داكنة وسيصبح لون النصّ أبيض.

لبرمجة زرّ نَسَق اللون:

> حدِّد حدث Click (الضغط) لمُكوِّن color_button (زرّ_ اللون). 1

> اضبط خاصية BackgroundColor (لون الخلفية) لمُكوِّن VerticalArrangement1 (الترتيب العمودي1) إلى White (أبيض). 2

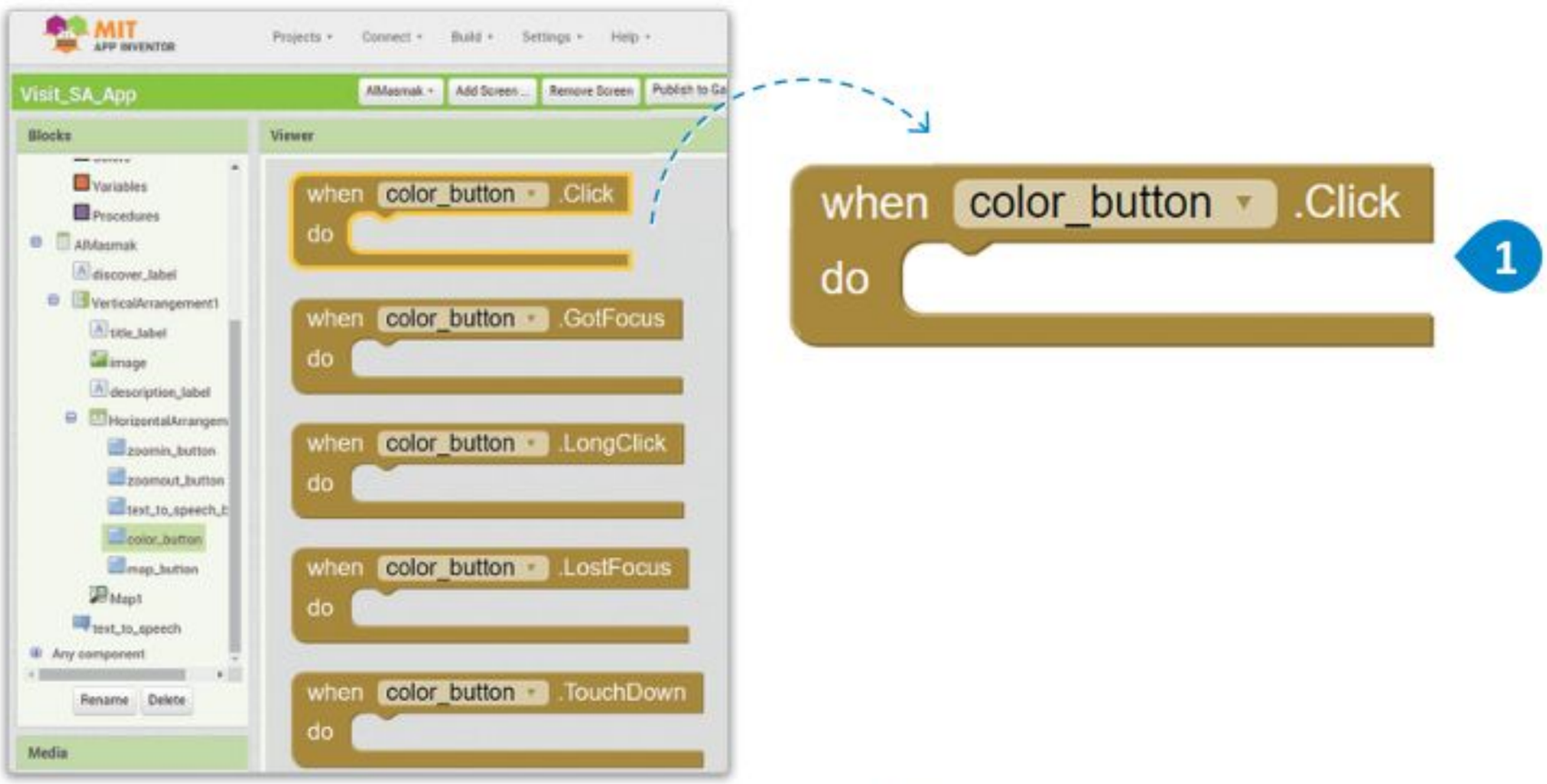
> اضبط خاصية BackgroundColor (لون الخلفية) لمُكوِّن discover_label (اكتشف_ التسمية) إلى Light Gray (رمادي فاتح). 3

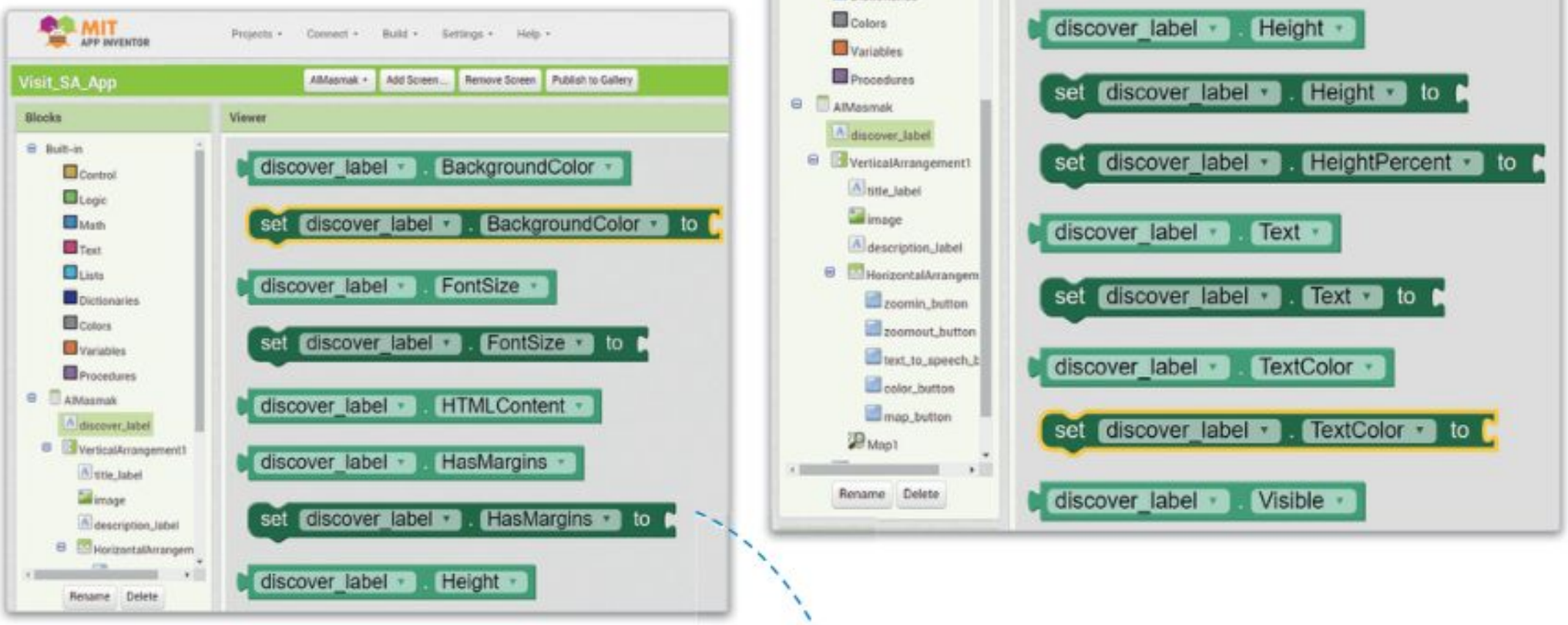
> اضبط خاصية TextColor (لون النصّ) لمُكوِّن discover_label (اكتشف_ التسمية) إلى Black (أسود). 4

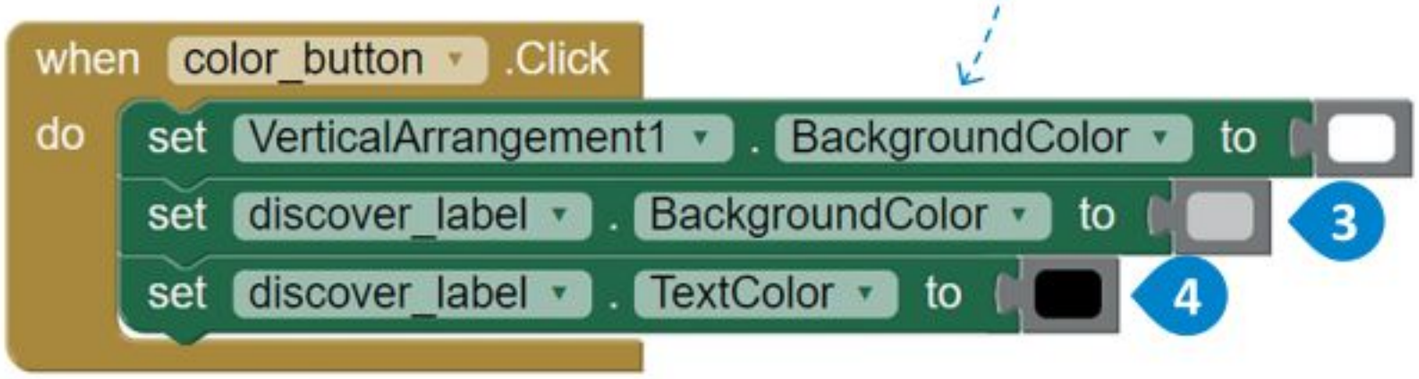
> كرِّر العملية المذكورة أعلاه لمكوّنات النصّ المتبقية. 5

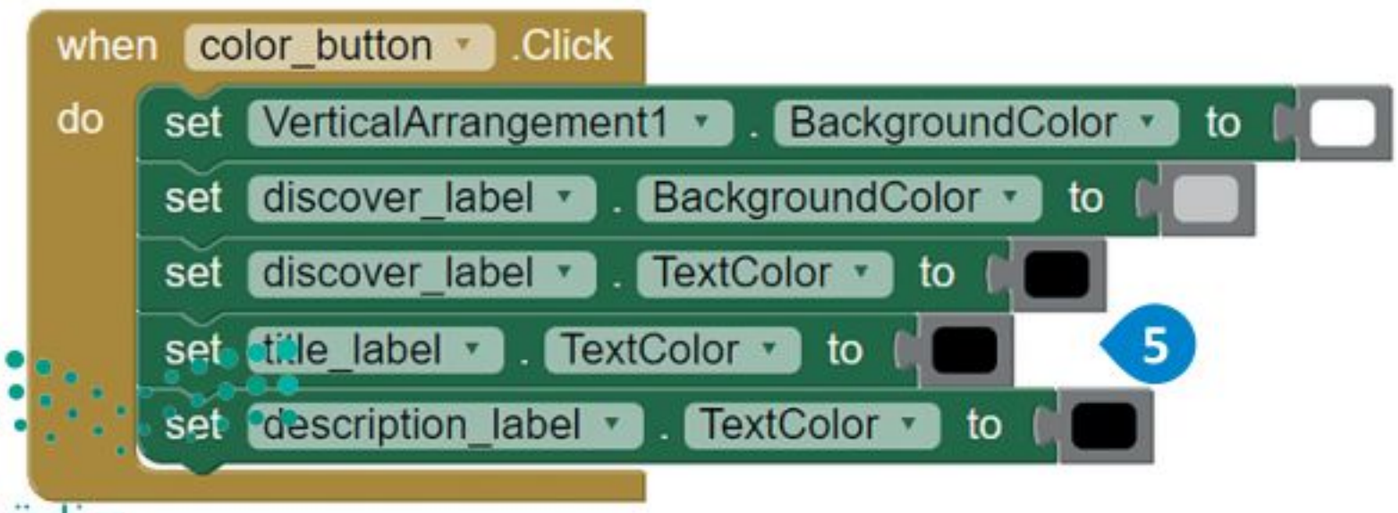
> حدِّد حدث Long Click (ضغطة طويلة) لمُكوِّن color_button (زرّ_ اللون). 6

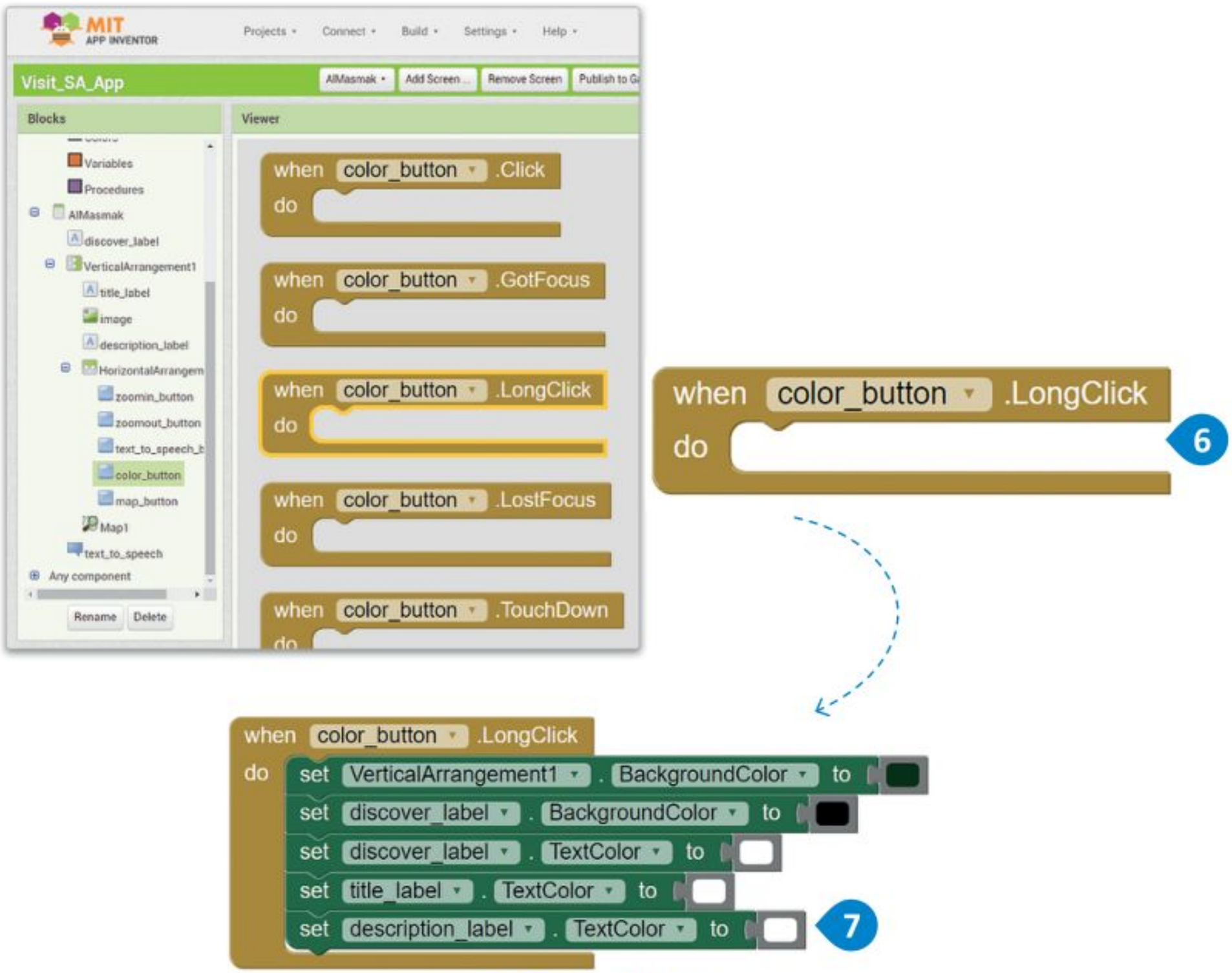
> اضبط الألوان المناسبة للمكونات كما في العملية أعلاه. 7




MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
Visit_SA_App
AlMasmak
Add Screen ...
Remove Screen
Publish to Gallery
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
AlMasmak
discover_label
VerticalArrangement1
title_label
image
description_label
HorizontalArrangem
Rename
Delete
discover_label . BackgroundColor
set discover_label . BackgroundColor to
discover_label . FontSize
set discover_label . FontSize to
discover_label . HTMLContent
discover_label . HasMargins
set discover_label . HasMargins to
discover_label . Height
zoomin_button
zoomout_button
color_button
map_button
Map1
discover_label . Height
set discover_label . Height to
set discover_label . HeightPercent to
discover_label . Text
set discover_label . Text to
discover_label . TextColor
set discover_label . TextColor to
discover_label . Visible


when color_button .Click
do
set VerticalArrangement1 . BackgroundColor to
set discover_label . BackgroundColor to
3
set discover_label . TextColor to
4

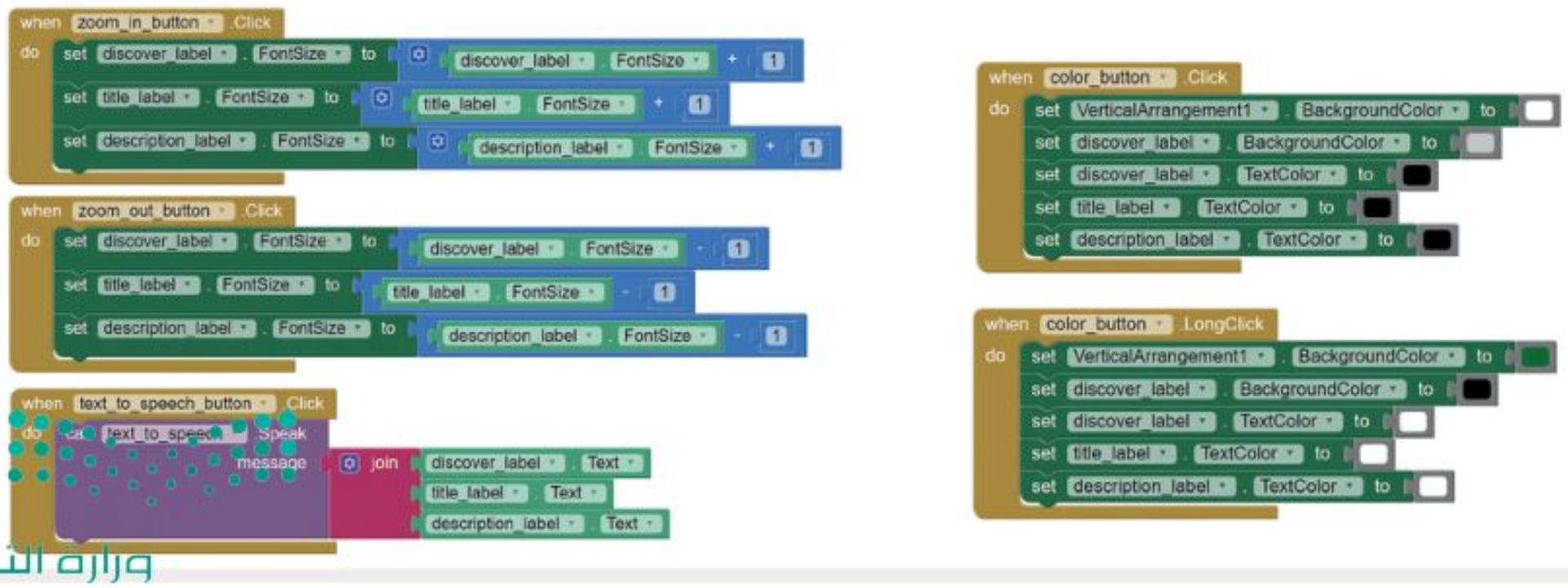

when color_button .Click
do
set VerticalArrangement1 . BackgroundColor to
set discover_label . BackgroundColor to
set discover_label . TextColor to
set title_label . TextColor to
5
set description_label . TextColor to



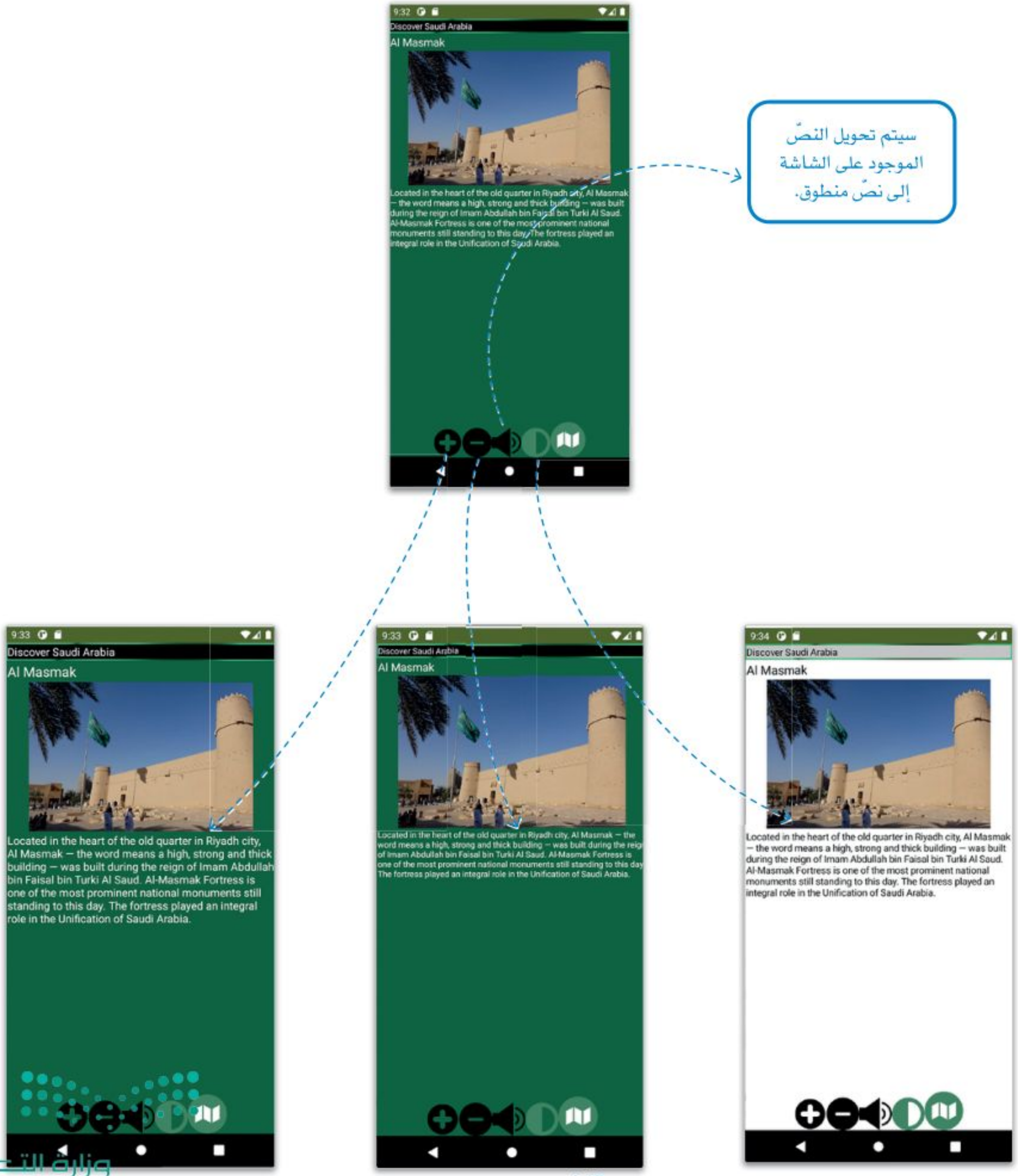
شكل 4.14: برمجة زرّ نَسَق اللون

البرنامج كاملًا بميزات قابلية الوصول في الشاشة الثالثة (المَصمك)

The Complete Code for the Accessibility Features on the Third Screen (Al Masmak)



أصبح البرنامج جاهزًا وعليك اختباره. شغِّل التطبيق باستخدام **محاكي الأندرويد** (Android Emulator) أو امسح رمز الاستجابة السريعة باستخدام جهاز الأندرويد لمعاينته. توضح الصور أدناه كيفية ظهور ميزات قابلية الوصول على جهاز الهاتف المحمول.



شكل 4.15: ميزات قابلية الوصول في المحاكي

تمرينات

1 صف كيف تساعد ميزات قابلية الوصول التي تتمّ إضافتها الأشخاصَ ذوي الاحتياجات المختلفة على استخدام التطبيق.

2 أضف زِرَّي التكبير والتصغير إلى الشاشات المتبقية من التطبيق.

3 قم بالتوسُّع في وظيفة زرّ التكبير والتصغير لتكبير مُكوِّني Button (زرّ) و ListPicker (قائمة الخيارات) في كل شاشة.

4 أضف زرًّا لتغيير نَسَق اللون إلى الشاشات المتبقية من التطبيق.

5 أضف زرّ تحويل النصّ إلى كلام للشاشات المتبقية من التطبيق.

6 وسّع وظيفة زرّ المتحدث لإضافة رسائل مُحدَّدة مسبقًا على كل شاشة لتقديم كل مُكوِّن قبل التحدث عن محتواه.

المشروع

1. أنشأت في الوحدة السابقة تطبيقًا للهاتف المحمول عن السياحة في المملكة العربية السعودية كجزء من رؤية 2030، ستعمل الآن على تحسينه لجعله قابلًا للوصول بسهولة.

2. يحتاج الأشخاص ذوو الاحتياجات الخاصة إلى الوصول إلى المعلومات المتعلقة بأحدث مشاريع المدن الكبرى في المملكة. ستُضيف الميزات الضرورية التي تجعل تطبيقك أكثر شمولًا للأشخاص ذوي الاحتياجات الخاصة المختلفة.

3. بشكل أكثر تحديدًا، ستُضيف ميزات لتمكين المُستخدِمين المصابين بعمى الألوان أو صعوبة الرؤية أو فقدان القدرة على الإبصار.

4. فكِّر في الميزات التي يجب دمجها في تطبيقك لمساعدة المُستخدِمين المذكورين أعلاه، وأضفها إلى التطبيق.

وزارة التعليم
Ministry of Education
2023 - 1445

ماذا تعلّمت

- التعرُّف على عمليات ومنهجيات اختبار تطبيقات البرمجيات.
- إنشاء تطبيق للهاتف المحمول، ونشره واختباره بواسطة مخترع التطبيقات MIT.
- التعرُّف على مدى تأثير الفجوة الرقميّة على المجتمع.
- تحديد كيفية تقليص الفجوة الرقميّة بتوظيف بعض الحلول.
- إضافة بعض الحلول لتوفير ميزات قابلية الوصول في الأجهزة والبرمجيات.
- تطوير التطبيق مع إضافة ميزات قابلية الوصول بالاعتبار.

المصطلحات الرئيسة

قابلية الوصول إلى الأجهزة	Hardware Accessibility
قابلية الوصول للبرمجيات	Software Accessibility
جودة البرمجيات	Software Quality
الاختبار	Testing
تحويل النصّ إلى كلام	Text-To-Speech
تصميم يركز على قابلية الاستخدام	Usability-Focused Design
قابلية الوصول إلى الويب	Web Accessibility

حزمة تطبيقات الأندرويد	Android App Bundle
حزمة الأندرويد	Android Package
تعيين إصدار التطبيق	Application Versioning
الاختبار المؤتمت	Automated Testing
تصحيح الأخطاء	Debugging
الفجوة الرقميّة	Digital Divide
التضمين الرقمي	Digital Inclusion

ملاحظات

وزارة التعليم
Ministry of Education
2023 - 1445

ملاحظات

وزارة التعليم
Ministry of Education
2023 - 1445

ملاحظات

وزارة التعليم
Ministry of Education
2023 - 1445