الأكاديمية العربية الدولية
Arab International Academy

إسم المادة: أساسيات أندرويد

إسم الدكتور:حسام يونسو

الأكاديمية العربية الدولية – منصة أعد

الأكاديمية العربية الدولية
Arab International Academy

محاور المادة

- المقدمة
- أنظمة التشغيل
- التعريف بنظام Android
- شعبية وميزات نظام أندرويد
- تاريخ أندرويد
- التطورات الكبرى عبر النسخ المختلفة
- بنية نظام Android
- الأدوات المستخدمة لتطوير تطبيقات Android
- تعريف ببرنامج Android Studio
- دعم المحاكاة الافتراضية للاختبار
- مكونات التطبيقات في Android
- مفهوم ملفات الـ Manifest
- اللغات المستخدمة في تطوير Android
- تطوير التطبيقات باستخدام React Native
- مميزات React Native
- الفائدة من إستخدام React Native
- الفائدة من إستخدام Expo CLI
- مميزات Expo
- سهولة اختبار التطبيقات
- الإحتياجات الخاصة لإستخدام Expo
- تحميل وتثبيت Node.JS
- تثبيت محرر VSCode
- تثبيت Expoوإنشاء المشروع
- طريقة تشغيل سيرفر Expo
- اختبار على أجهزة فعلية
- إختبار على Android Studio

الأكاديمية العربية الدولية
Arab International Academy

المقدمة

في هذا العرض، سنشرح بالتفصيل أساسيات نظام Android، بدءًا من تعريفه وتاريخه وتطوره إلى كيفية عمله وبنيته الداخلية، بما في ذلك دورة حياة التطبيقات وكيفية التحكم في حالات التطبيق المختلفة .

سنتعرف أيضًا على بيئة التطوير (IDE)المستخدمة في بناء التطبيقات، مثل Android Studio، وكيفية إعداد واستخدام **React Native** و **Expo CLI** كأدوات تسهل عملية تطوير التطبيقات على أنظمة Android وiOS باستخدام كود مشترك يعتمد على JavaScript، مع شرح خطوات إعداد مشروع باستخدام **Expo CLI** وكيفية اختبار التطبيقات على الأجهزة الفعلية والمحاكيات، مما يوفر تجربة تطوير متكاملة ومبسطة.

أنظمة التشغيل

أنظمة التشغيل (Operating Systems)هي برامج أساسية تُدير جميع مكونات وعمليات الكمبيوتر أو الجهاز الذكي .تعمل كوسيط بين المستخدم ومكونات الجهاز، وتُشغِّل التطبيقات وتُدير الموارد مثل المعالج والذاكرة.

أهم وظائف أنظمة التشغيل:

إدارة الموارد :مثل المعالج، الذاكرة، والأجهزة الطرفية.

توفير واجهة للمستخدم :تتيح التفاعل مع الجهاز عبر واجهة رسومية (GUI)أو نصية.

تشغيل التطبيقات :يُمكّن التطبيقات من العمل والتواصل مع الجهاز.

التعريف بنظام Android

نظام **Android** هو نظام تشغيل مفتوح المصدر تم تطويره بواسطة شركة Google، ويستند إلى نواة نظام **لينكس (Linux)**. تم تصميمه بشكل أساسي للأجهزة المحمولة مثل الهواتف الذكية والأجهزة اللوحية، ولكنه يمتد أيضًا ليشمل أجهزة مثل التلفزيونات الذكية والساعات والسيارات وحتى الأجهزة المنزلية الذكية.

شعبية وميزات نظام أندرويد

منذ إطلاقه في عام 2008، حقق Androidانتشارًا واسعًا وأصبح نظام التشغيل الأكثر استخدامًا عالميًا .حاليًا، يُشغّل أكثر من 70%من الهواتف الذكية في العالم بفضل دعمه لعدد كبير من الشركات المصنعة مثل سامسونج وهواوي وشاومي.

ميزاته:

مفتوح المصدر :**Open Source**يُتيح للمطورين والمصنعين تخصيصه وفق احتياجاتهم.

تعدد الأجهزة :يعمل على مجموعة واسعة من الأجهزة بمواصفات وأسعار مختلفة.

متجر التطبيقات :يوفر **Google Play**ملايين التطبيقات التي تلبي مختلف احتياجات المستخدمين.

التكامل مع خدمات Google: مثل Gmailو Google Mapsو Google Assistant.

تاريخ أندرويد

تم تطوير نظام Androidفي البداية من قبل شركة **Android Inc.** عام 2003، والتي كانت تعمل على تطوير نظام تشغيل للأجهزة المحمولة .في عام 2005، استحوذت شركة **Google** على Android Inc. لتبدأ رحلتها في تطويره كنظام تشغيل للهواتف الذكية.

أُطلِق أول هاتف يعمل بنظام Androidفي عام **2008** وكان الهاتف هو **HTC Dream**. مع هذا الإطلاق، ظهر Androidكنظام تشغيل مفتوح المصدر يتيح للشركات المصنعة حرية تخصيصه.

التطورات الكبرى عبر النسخ المختلفة

2008: إصدار **Android 1.0**، الذي قدم ميزات أساسية مثل Gmail وGoogle Maps.

2011: إصدار **Android 3.0 (Honeycomb)**، الذي كان مخصصًا للأجهزة اللوحية مع واجهة مخصصة.

2014: إصدار **Android 5.0 (Lollipop)**، الذي قدم تصميمًا جديدًا يسمى **Material Design**.

2017: إصدار **Android 8.0 (Oreo)**، الذي جلب تحسينات في الأداء والبطارية.

2021: إصدار **Android 12**، الذي ركز على التخصيصات الشخصية وتعزيز الأمان.

2023: إصدار **Android 14**، الذي عزز التكامل مع تقنيات الذكاء الاصطناعي وتحسينات الخصوصية.

2024: إصدار **Android 15 (Vanilla Ice Cream)**

بنية نظام Android

نظام Androidيتكون من عدة طبقات تعمل معًا لتشغيل التطبيقات وتوفير الخدمات للمستخدم:

نواة لينكس (Linux Kernel):

- تقع في الأساس، وهي المسؤولة عن إدارة الأجهزة)مثل المعالج والذاكرة.(
- تتحكم في الاتصال بين التطبيقات والعتاد)الهاردوير (من خلال برامج التشغيل .(Drivers)

مكتبات النظام (System Libraries):

- تحتوي على مكتبات مفتوحة المصدر توفر الخدمات اللازمة لتشغيل التطبيقات، مثل التعامل مع الوسائط (Media)وقواعد البيانات .(SQLite)

إطار العمل (Application Framework):

- يوفر الأدوات والخدمات التي تساعد التطبيقات على التفاعل مع النظام ومع بعضها البعض، مثل إدارة النشاطات (Activities)والإشعارات (Notifications).

طبقة التطبيقات (Applications):

- التطبيقات التي يستخدمها المستخدمون مثل الرسائل والمتصفح.
- كل تطبيق يعمل بشكل مستقل عن الآخر في بيئة معزولة .(Sandbox)

الأدوات المستخدمة لتطوير تطبيقات Android

لتطوير تطبيقات Android، يحتاج المطورون إلى **بيئة تطوير متكاملة (IDE)** تساعدهم في كتابة الكود وتصحيح الأخطاء واختبار التطبيقات.

الأدوات الشائعة المستخدمة تشمل:

Android Studio: الأداة الرسمية لتطوير تطبيقات Android.

Visual Studio Code مع إضافات مخصصة مثل Flutter أو React Native.

Expo CLI: لتطوير تطبيقات باستخدام React Native.

تعريف ببرنامج Android Studio

Android Studioهو بيئة التطوير المتكاملة الرسمية التي طورتها Google خصيصًا لتطوير تطبيقات Android.

يعتمد على IntelliJ IDEA ويوفر مجموعة شاملة من الأدوات مثل:

محرر كود ذكي يدعم Java و Kotlin.

أدوات لتصميم الواجهات الرسومية باستخدام **XML**.

دعم كامل لـ **Gradle** لإدارة المشاريع.

دعم المحاكاة الافتراضية للاختبار

Android Studioيحتوي على محاكي Android (Emulator)الذي يسمح للمطورين بتشغيل واختبار التطبيقات على مجموعة متنوعة من الأجهزة الافتراضية المختلفة.

المحاكي يوفر إمكانية اختيار أجهزة بمواصفات متنوعة)مثل حجم الشاشة وإصدار Android)واختبار الأداء والوظائف دون الحاجة إلى جهاز فعلي.

يدعم أيضًا ميزات مثل تغيير الموقع الجغرافي وإعدادات الشبكة لاختبار التطبيقات في بيئات مختلفة.

باستخدام **Android Studio**والمحاكي المدمج، يمكن للمطورين تجربة تطبيقاتهم واكتشاف الأخطاء وتحسين الأداء قبل نشرها.

مكونات التطبيقات في Android

تطبيقات Androidتتكون من عدة مكونات رئيسية تعمل معًا لتوفير تجربة متكاملة للمستخدم .هذه المكونات تشمل:

الأنشطة :(Activities)

- كل شاشة في التطبيق تُعتبر نشاطًا.
- تمثل النشاطات واجهات المستخدم حيث يتفاعل المستخدم مع التطبيق.
- كل تطبيق يحتوي عادةً على عدة أنشطة، ويتم إدارة التنقل بينها بواسطة النظام.

الخدمات :(Services)

- تعمل في الخلفية لتنفذ عمليات طويلة دون التفاعل المباشر مع المستخدم.
- يمكن أن تكون خدمات مثل تشغيل الموسيقى أو تنزيل البيانات في الخلفية.

المستقبلات :(Broadcast Receivers)

- تتلقى وتتعامل مع رسائل النظام أو التطبيقات الأخرى.
- مثال :يمكن للتطبيق استقبال إشعار عند اكتمال تنزيل ملف، أو عند تغير حالة الشبكة.

الموفرون :(Content Providers)

- يُستخدمون لتبادل البيانات بين التطبيقات.
- مثل توفير الوصول إلى جهات الاتصال أو الصور، أو مشاركة البيانات مع تطبيقات أخرى بشكل منظم وآمن.

مفهوم ملفات الـ Manifest

ملف الـ Manifest هو ملف مهم في كل تطبيق Android. يُسمى **AndroidManifest.xml** ويحتوي على معلومات أساسية عن التطبيق مثل:

- **تعريف الأنشطة والخدمات والمستقبلات** المستخدمة في التطبيق.
- **الأذونات (Permissions)** التي يحتاجها التطبيق للوصول إلى ميزات النظام مثل الكاميرا أو الإنترنت.
- **الحد الأدنى لإصدار Android** الذي يدعمه التطبيق.
- **نقطة الدخول الأساسية للتطبيق** (النشاط الرئيسي الذي يُفتح عند تشغيل التطبيق.)
- يعمل ملف الـ Manifest كجسر بين تطبيقك ونظام Android، ويحدد كيفية تفاعل التطبيق مع النظام والموارد المتاحة.

اللغات المستخدمة في تطوير Android

1. Java: اللغة التقليدية لتطوير تطبيقات Android

Java هي اللغة الأساسية التي استُخدمت لتطوير تطبيقات Android منذ إطلاق النظام.

تم تطوير Android في البداية باستخدام Java نظرًا لكونها لغة برمجة قوية، ومشهورة، وتدعم مفهوم البرمجة الكائنية (OOP).

ميزات **Java** في تطوير Android:

- **قابلة للنقل**: يمكن تشغيلها على أي نظام يدعم Java.
- مجتمع كبير من المطورين ودعم قوي.
- تكامل سلس مع Android Studio وأدوات التطوير.

اللغات المستخدمة في تطوير Android

2. Kotlin: اللغة الحديثة الموصى بها من Google

في عام 2017، أعلنت **Google** دعم **Kotlin** كلغة رسمية لتطوير تطبيقات Android.

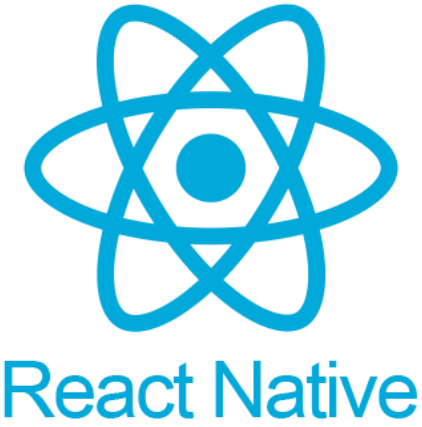
Kotlin هي لغة حديثة، متعددة المنصات، وتعتبر أكثر أمانًا وفعالية من Java.

ميزات **Kotlin**:

- **أقل تعقيدًا**: كتابة الكود في Kotlin تكون مختصرة وأكثر وضوحًا.
- **توافق كامل مع Java**: يمكن للمطورين استخدام Java وKotlin معًا في مشروع واحد.
- **أمان أكبر**: تقلل من الأخطاء البرمجية.
- **أداء أفضل**: تشجع على استخدام الموارد بفعالية.

تطوير التطبيقات باستخدام React Native

تعريف بـ React Native

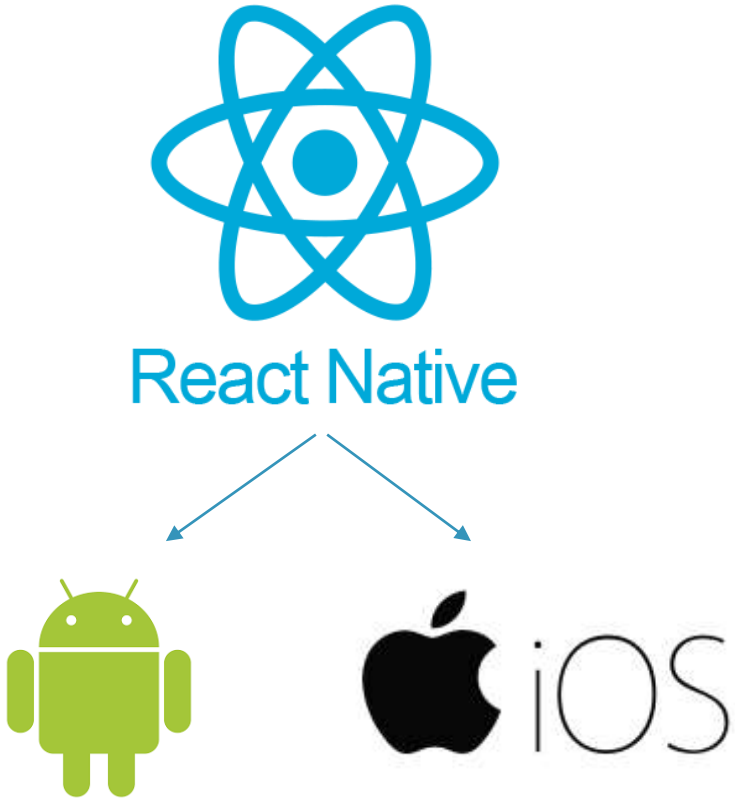


React Native هو (Framework) إطار عمل مفتوح المصدر تم تطويره بواسطة **Facebook** ويُستخدم لتطوير تطبيقات الهواتف الذكية باستخدام لغة **JavaScript**.

يتيح للمطورين بناء تطبيقات أصلية (Native) لأنظمة **Android** و**iOS** من خلال استخدام كود مشترك يعتمد على مكتبة **React**.

مميزات React Native

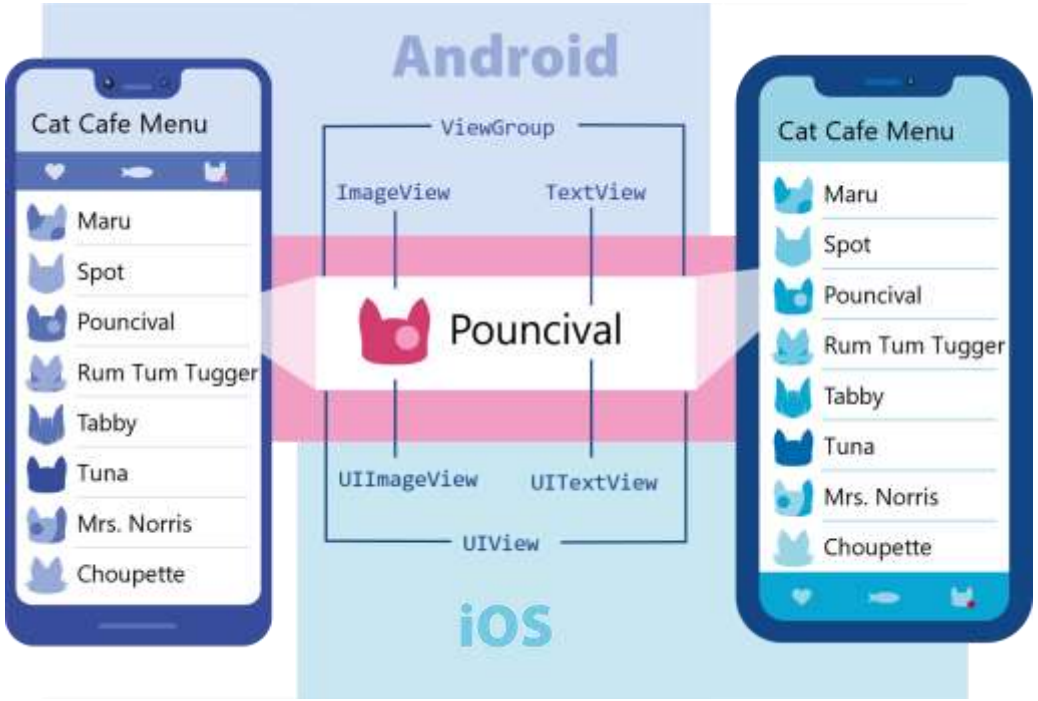
كود مشترك بين Android وiOS: يتيح إمكانية كتابة كود واحد يُستخدم لتطوير تطبيقات لكل من Android وiOS، مما يُقلل الجهد المطلوب لتطوير وصيانة التطبيقات.



مميزات React Native

مكونات أصلية (Native Components): يقدم مكونات أصلية تُترجم إلى كود أصلي في iOSوAndroid، مما يتيح أداءً عاليًا وتجربة مستخدم تشبه التطبيقات الأصلية.

دعم واسع للمكتبات: بفضل مجتمع مطوري React Native، هناك مكتبات وأدوات جاهزة تسهل التكامل مع ميزات النظام مثل الكاميرا أو الموقع.



الفائدة من إستخدام React Native

تطوير أسرع :كتابة كود واحد لنظامين يوفر الوقت والجهد.

تجربة مستخدم عالية الجودة :على الرغم من أنه يعتمد على JavaScript، فإن التطبيقات المُطورة باستخدام React Native تقدم تجربة استخدام مماثلة للتطبيقات الأصلية.

التوافق مع Android وiOS

باستخدام **React Native**، يمكن للمطورين كتابة كود مشترك لكل من **Android**و**iOS**، مما يتيح نشر التطبيقات على كلا النظامين بسهولة.

تُترجم الأكواد المكتوبة بـ React Nativeإلى مكونات وواجهات أصلية لكل نظام، مما يحسن الأداء والتوافق مع ميزات الأجهزة.

الفائدة من إستخدام Expo CLI

Expo CLIهو مجموعة أدوات مفتوحة المصدر مصممة لتسهيل عملية تطوير تطبيقات .**React Native**

يوفر بيئة تطوير متكاملة تمكن المطورين من بناء تطبيقات الهواتف الذكية بسرعة باستخدام JavaScriptوReact Native، دون الحاجة للتعامل مع إعدادات معقدة للمشاريع الأصلية (Native Projects).

مميزات Expo

إعداد سريع وبسيط Expo :يلغي الحاجة إلى إعدادات معقدة للمشاريع ويتيح لك بدء تطوير التطبيق فورًا باستخدام أوامر بسيطة.

مكتبات جاهزة :يأتي مع مجموعة واسعة من المكتبات والمكونات الجاهزة مثل الكاميرا، GPS، والإشعارات، مما يسهل التكامل مع ميزات الأجهزة.

مناسب للمبتدئين والمحترفين Expo :مثالي للمطورين من جميع المستويات لأنه يقدم واجهة بسيطة وسهلة الاستخدام مع ميزات قوية.

سهولة اختبار التطبيقات

اختبار على أجهزة فعلية :باستخدام تطبيق **Expo Go** على الهاتف، يمكن للمطورين اختبار التطبيقات بشكل مباشر على أجهزة Android و iOS من خلال مسح رمز QR.

دعم المحاكيات Expo CLI :يتكامل بسهولة مع المحاكيات الافتراضية مثل **Android Emulator** و **iOS Simulator** لاختبار التطبيقات على الكمبيوتر.

التحديث الفوري (Hot Reloading): أي تغيير في الكود يظهر فورًا على الجهاز أو المحاكي دون الحاجة إلى إعادة تشغيل التطبيق، مما يجعل عملية التطوير أسرع وأكثر كفاءة.

الإحتياجات الخاصة لإستخدام Expo

جهاز كمبيوتر:

يعمل بنظام تشغيل **Windows**، **macOS**، أو **Linux**.

تحميل وتثبيت Node.JS

Node.js:

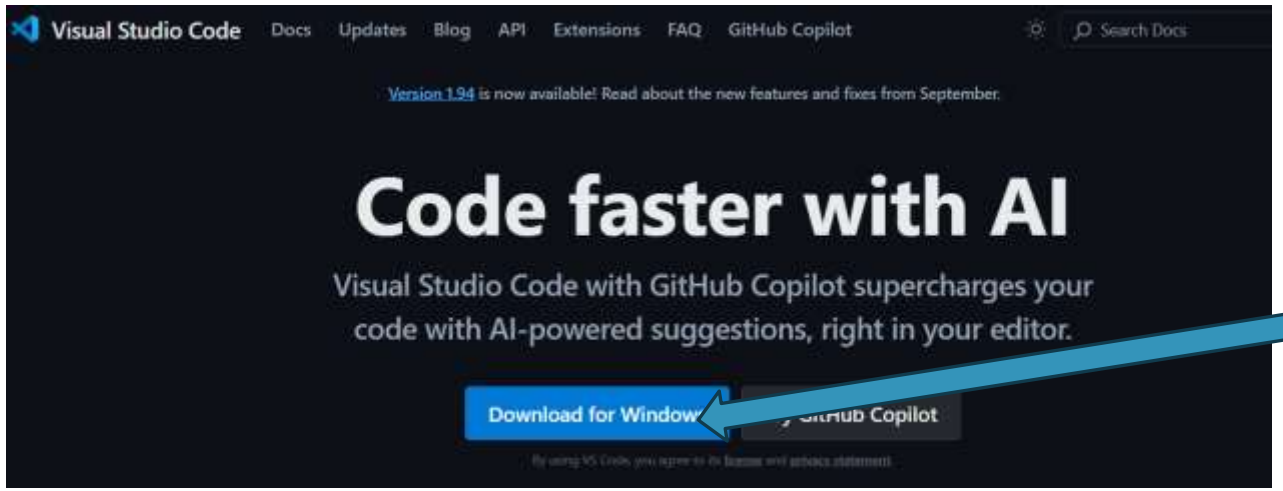
تحتاج إلى تثبيت **Node.js** على جهازك لأنه يعتمد عليه كل من Expo و React Native.

يمكنك تنزيله من الموقع الرسمي : https://nodejs.org/.

بعد الدخول إلى الموقع
إضغط على Download Node.js (LTS)
وقم بتثبيته على جهازك.

تثبيت محرر VSCode

يمكنك تنزيل **VS Code** من الموقع الرسمي :
https://code.visualstudio.com/.

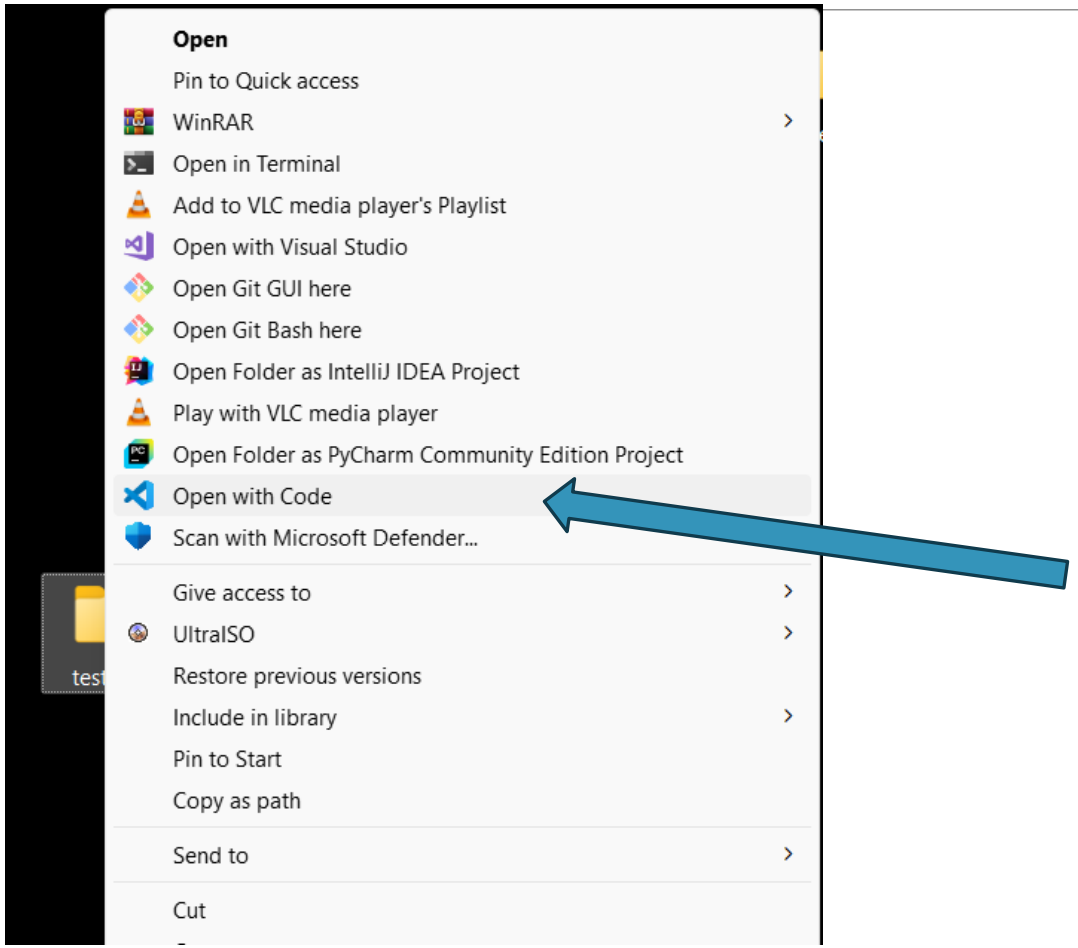


إضغط على

Download for Windows

وقم بتنصيب البرنامج

إنشاء المجلد الخاص بالمشروع



بعد تنصيب برنامج VSCodeنقوم بإنشاء مجلد جديد في ال الجهاز الخاص بنا..

ونضغط بالزر الأيمن في الماوس على المجلد

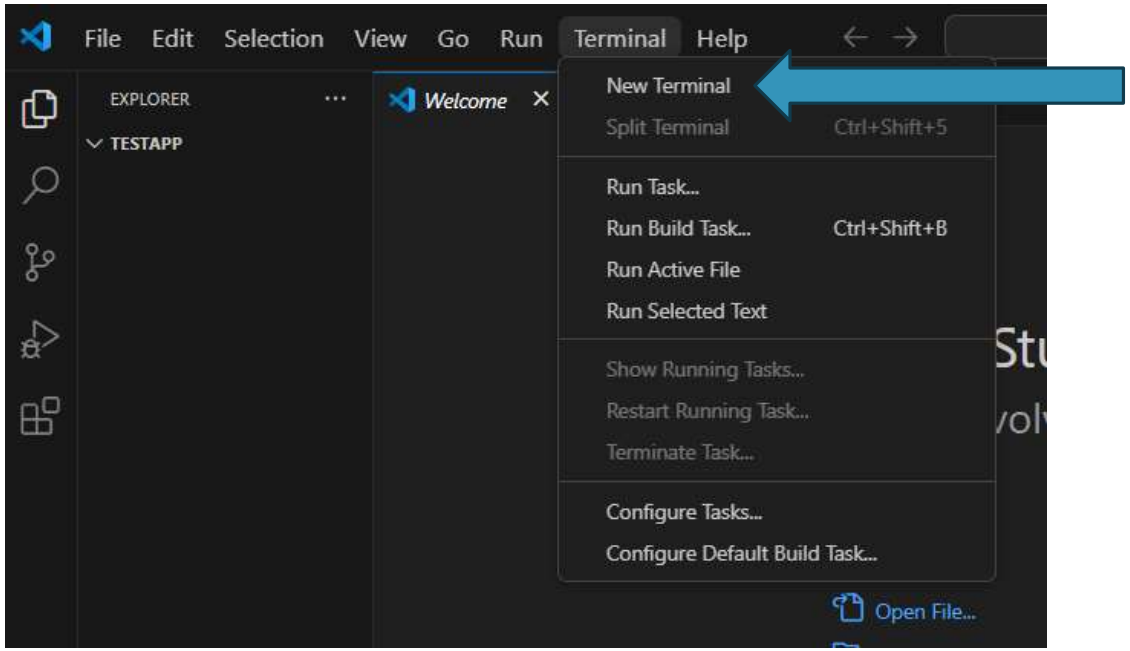
ونبحث عن خيار

Open With Code

فتح ال Terminal

نقوم بالضغط على Terminal

ونقوم بفتح Terminalجديد



تثبيت Expo وإنشاء المشروع

قم بتثبيت Expo CLI باستخدام npm:

```
PS C:\Users\Mulan\Desktop\testApp> npm install -g expo-cli
```

وبعدها نقوم بإنشاء المشروع الخاص بنا وكتابة اسم المشروع بدل كلمة

StickerSmash

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Mulan\Desktop\testApp> npx create-expo-app@latest StickerSmash --template blank
```

الإنتهاء من إنشاء المشروع

عند الحصول على نتيجة مماثلة للتالي ,فذلك يعني نجاح عملية إنشاء المشروع

```
Run `npm audit` for details.

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd StickerSmash
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
PS C:\Users\Mulan\Desktop\testApp>
```

طريقة تشغيل سيرفر Expo

يجب إستخدام الأمر التالي للدخول إلى مسار المشروع:

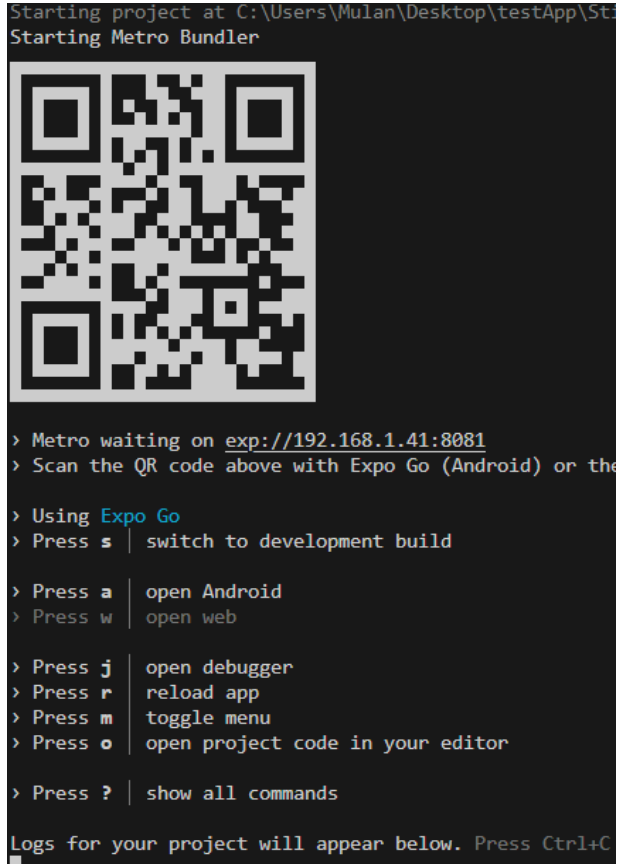
```
cd StickerSmash
```

بعد الإنشاء، يمكنك تشغيل التطبيق مباشرة على محاكي أو جهاز فعلي باستخدام:

```
PS C:\Users\Mulan\Desktop\testApp\StickerSmash> npx expo start
```

اختبار على أجهزة فعلية

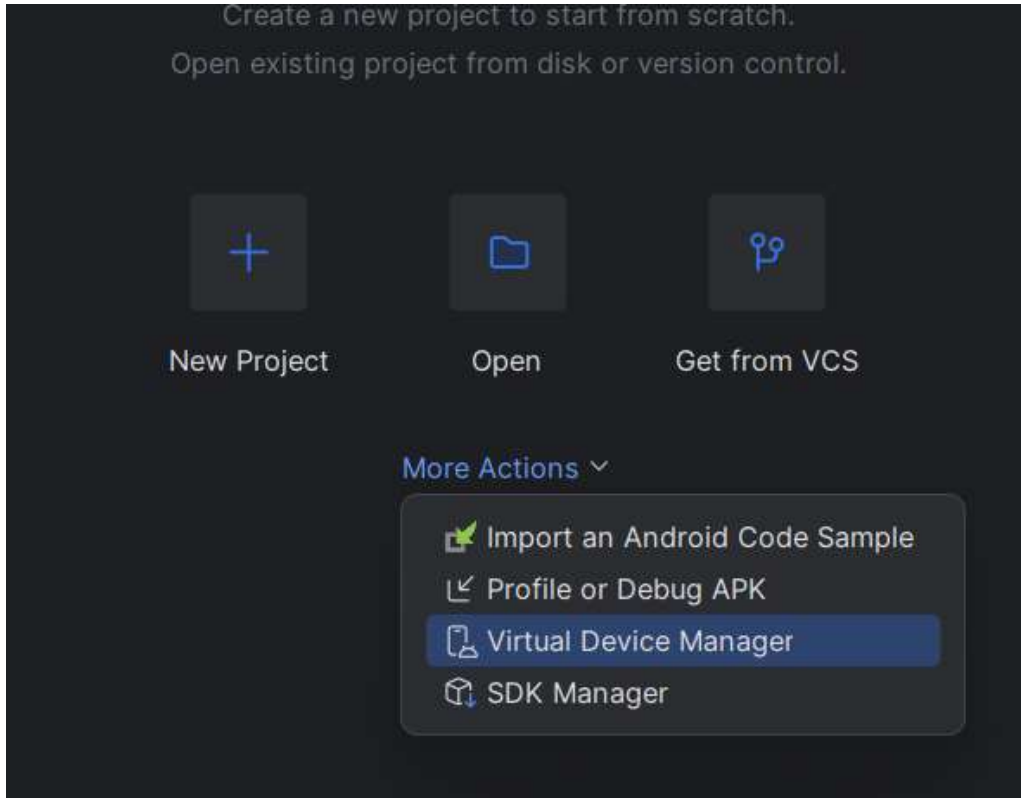
باستخدام تطبيق **Expo Go** على الهاتف، يمكن للمطورين اختبار التطبيقات بشكل مباشر على أجهزة Android و iOS من خلال مسح رمز QR.



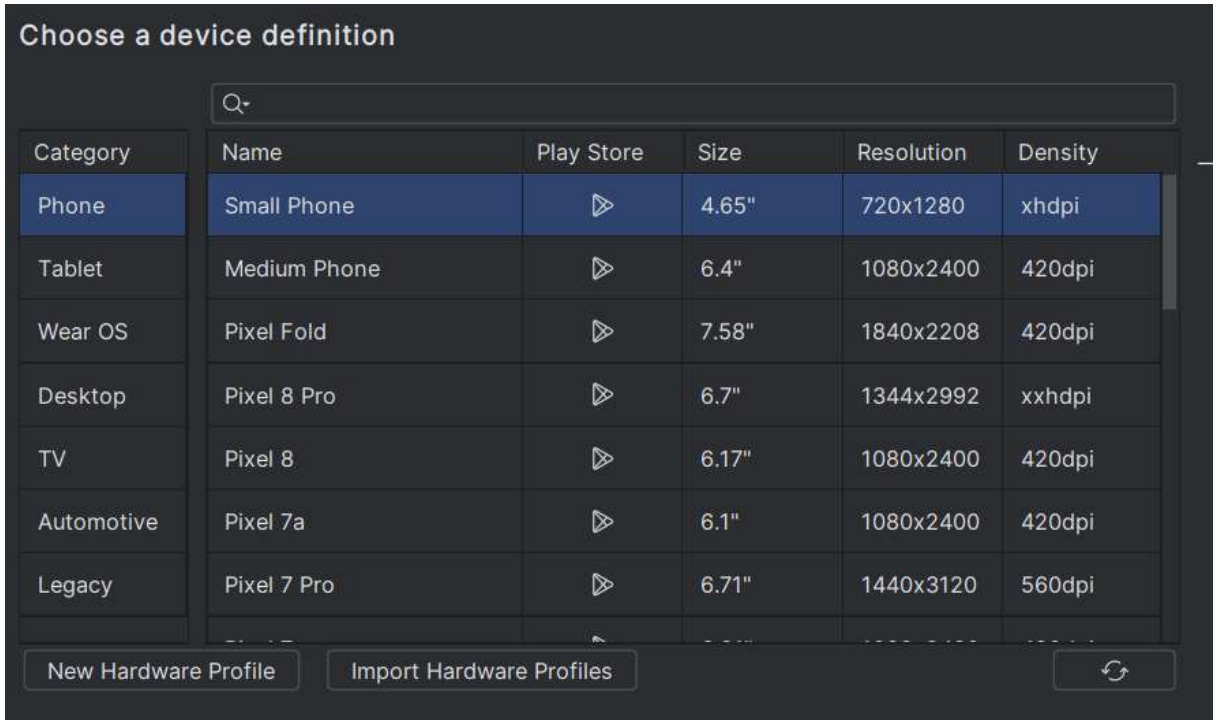
إختبار على Android Studio

لتشغيل التطبيق على المحاكي الخاص ب Android Studio يجب علينا إنشاء جهاز إفتراضي.

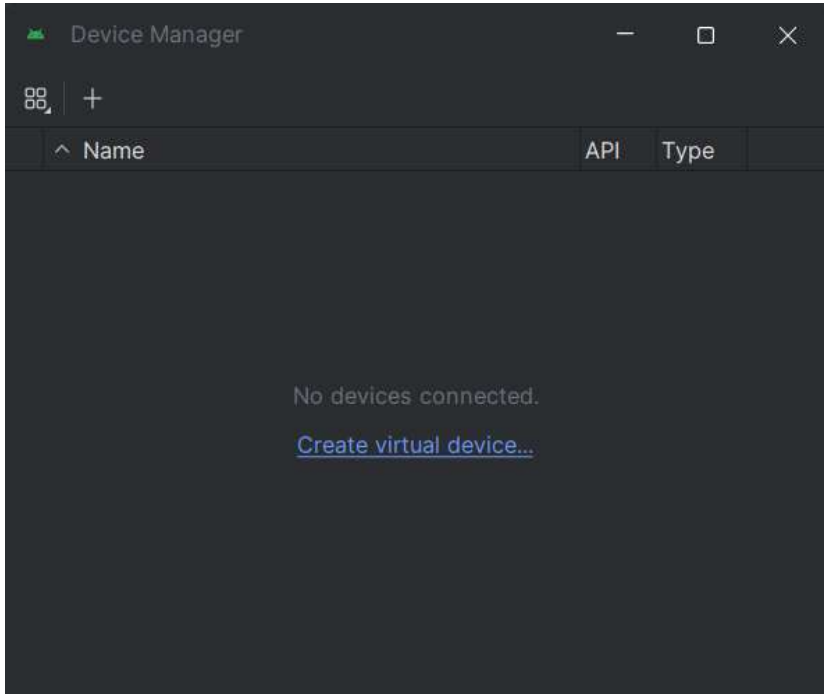
نقوم بالضغط على Virtual Device Manager



إنشاء جهاز إفتراضي على Android Studio

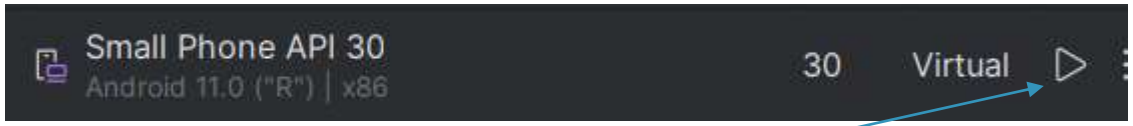


نختار نوع الجهاز وحجمه وبعدها نضغط على **Next**



نضغط على **Create virtual device**

إختيار إصدار أندرويد الخاص بالجهاز



بعد إنشاء الجهاز نضغط على زر التشغيل

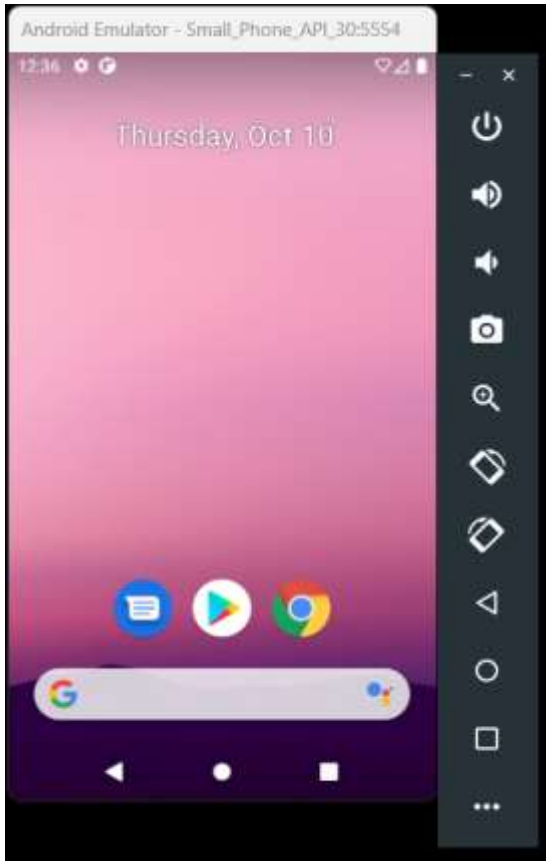
إختر إصدار الأندرويد وبعدها إضغط على Nextوبعدها Finish

نجاح عملية تشغيل التطبيق

بعد تشغيل الجهاز الإفتراضي ,يمكننا الذهاب إلى ال Terminalفي VSCodeوالضغط على الحرف Aلتشغيل التطبيق

```
› Opening on Android...
√ Select a device/emulator » Small_Phone_API_30 (emulator)
› Opening exp://192.168.1.41:8081 on Small_Phone_API_30
Downloading the Expo Go app [=============                                ] 20% 141.0s
```

سيتم تحميل تطبيق Expo Goعلى الجهاز بشكل أوتوماتيكي ,وسيتم تشغيل التطبيق بنجاح.



الأكاديمية العربية الدولية
Arab International Academy

شكراً لكم