

**TRƯỜNG ĐẠI HỌC THỦY LỢI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**GIÁO TRÌNH**  
**THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

Hà Nội, 2.2025

## MỤC LỤC

CHƯƠNG 1.	Làm quen .....	Error! Bookmark not defined.
Bài 1)	Tạo ứng dụng đầu tiên .....	Error! Bookmark not defined.
1.1)	Android Studio và Hello World .....	Error! Bookmark not defined.
1.2)	Giao diện người dùng tương tác đầu tiên.....	Error! Bookmark not defined.
1.3)	Trình chỉnh sửa bố cục.....	Error! Bookmark not defined.
1.4)	Văn bản và các chế độ cuộn.....	Error! Bookmark not defined.
1.5)	Tài nguyên có sẵn .....	Error! Bookmark not defined.
Bài 2)	Activities .....	Error! Bookmark not defined.
2.1)	Activity và Intent .....	Error! Bookmark not defined.
2.2)	Vòng đời của Activity và trạng thái.....	Error! Bookmark not defined.
2.3)	Intent ngầm định .....	Error! Bookmark not defined.
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ .....	Error! Bookmark not defined.
3.1)	Trình gỡ lỗi.....	Error! Bookmark not defined.
3.2)	Kiểm thử đơn vị.....	Error! Bookmark not defined.
3.3)	Thư viện hỗ trợ .....	Error! Bookmark not defined.
CHƯƠNG 2.	Trải nghiệm người dùng .....	Error! Bookmark not defined.
Bài 1)	Tương tác người dùng.....	Error! Bookmark not defined.
1.1)	Hình ảnh có thể chọn .....	Error! Bookmark not defined.
1.2)	Các điều khiển nhập liệu .....	Error! Bookmark not defined.
1.3)	Menu và bộ chọn .....	Error! Bookmark not defined.
1.4)	Điều hướng người dùng .....	Error! Bookmark not defined.
1.5)	RecyclerView .....	Error! Bookmark not defined.
Bài 2)	Trải nghiệm người dùng thú vị .....	Error! Bookmark not defined.
2.1)	Hình vẽ, định kiểu và chủ đề .....	Error! Bookmark not defined.
2.2)	Thẻ và màu sắc .....	Error! Bookmark not defined.
2.3)	Bố cục thích ứng .....	Error! Bookmark not defined.
Bài 3)	Kiểm thử giao diện người dùng.....	Error! Bookmark not defined.

3.1)	Espresso cho việc kiểm tra UI.....	<b>Error! Bookmark not defined.</b>
CHƯƠNG 3. Làm việc trong nền.....		<b>Error! Bookmark not defined.</b>
Bài 1)	Các tác vụ nền .....	<b>Error! Bookmark not defined.</b>
1.1)	AsyncTask .....	<b>Error! Bookmark not defined.</b>
1.2)	AsyncTask và AsyncTaskLoader .....	<b>Error! Bookmark not defined.</b>
1.3)	Broadcast receivers .....	<b>Error! Bookmark not defined.</b>
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	<b>Error! Bookmark not defined.</b>
2.1)	Thông báo.....	<b>Error! Bookmark not defined.</b>
2.2)	Trình quản lý cảnh báo.....	<b>Error! Bookmark not defined.</b>
2.3)	JobScheduler.....	<b>Error! Bookmark not defined.</b>
CHƯƠNG 4. Lưu dữ liệu người dùng.....		<b>Error! Bookmark not defined.</b>
Bài 1)	Tùy chọn và cài đặt.....	<b>Error! Bookmark not defined.</b>
1.1)	Shared preferences .....	<b>Error! Bookmark not defined.</b>
1.2)	Cài đặt ứng dụng.....	<b>Error! Bookmark not defined.</b>
Bài 2)	Lưu trữ dữ liệu với Room .....	<b>Error! Bookmark not defined.</b>
2.1)	Room, LiveData và ViewModel .....	<b>Error! Bookmark not defined.</b>
2.2)	Room, LiveData và ViewModel .....	<b>Error! Bookmark not defined.</b>
3.1)	Trình gỡ lỗi .....	

# CHƯƠNG 1: LÀM QUEN

## Bài 1: Tạo ứng dụng đầu tiên

### 1.1 Android và Hello world

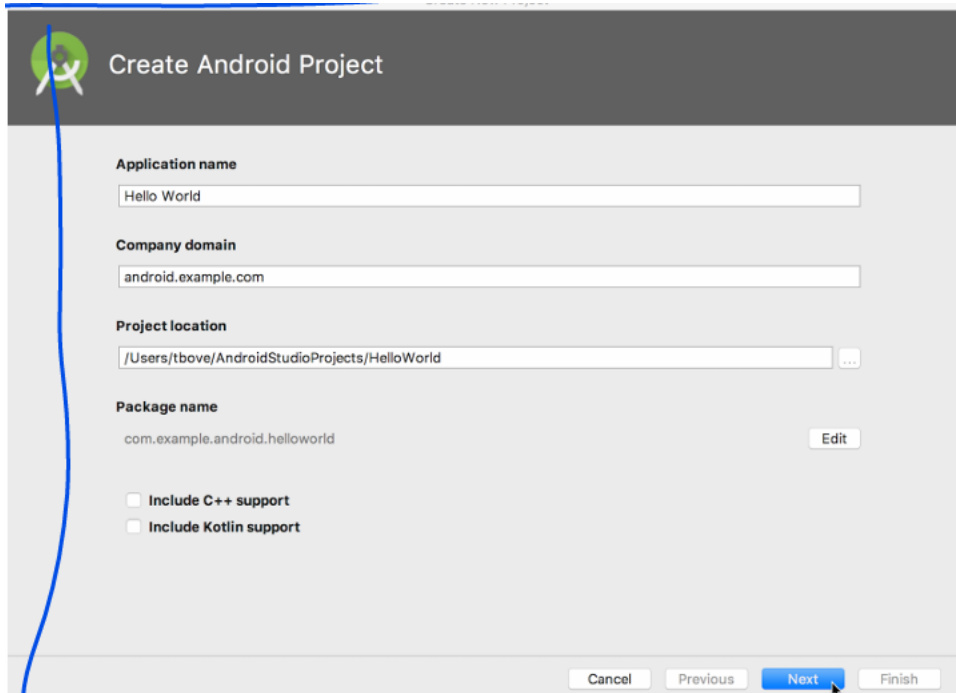
#### Giới thiệu

Trong bài thực hành này, bạn sẽ tìm hiểu cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, Hello World, trên một trình giả lập và trên một thiết bị vật lý.

#### Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.



## Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt mới nhất của Android Studio và Java lên máy tính của bạn.

## Những gì bạn sẽ học

- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

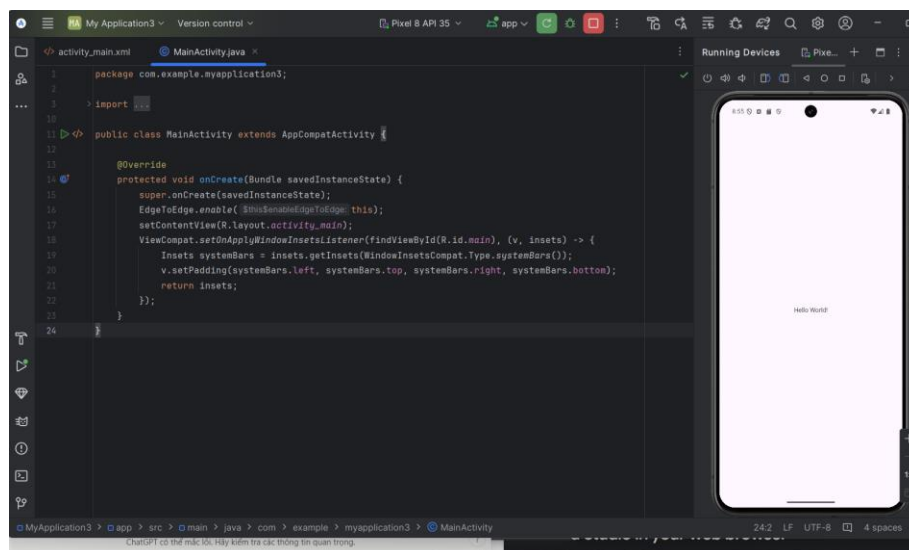
## Những gì bạn sẽ làm

- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một trình giả lập (thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.
- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

## Tổng quan về ứng dụng

Sau khi cài đặt thành công Android Studio, bạn sẽ tạo một dự án mới cho ứng dụng Hello World từ một mẫu. Ứng dụng đơn giản này hiển thị chuỗi "Hello World" trên màn hình của thiết bị ảo hoặc vật lý Android.

Đây là ứng dụng hoàn thành sẽ trông như thế nào:



## Nhiệm vụ 1: Cài đặt Android Studio

Android Studio cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh, bao gồm trình soạn thảo mã nâng cao và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ hỗ trợ phát triển, gỡ lỗi, kiểm thử và tối ưu hiệu suất, giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn.

Bạn có thể kiểm thử ứng dụng của mình trên nhiều trình giả lập được cấu hình sẵn hoặc trực tiếp trên thiết bị di động của mình, xây dựng ứng dụng để đưa vào sản xuất và xuất bản trên cửa hàng Google Play.

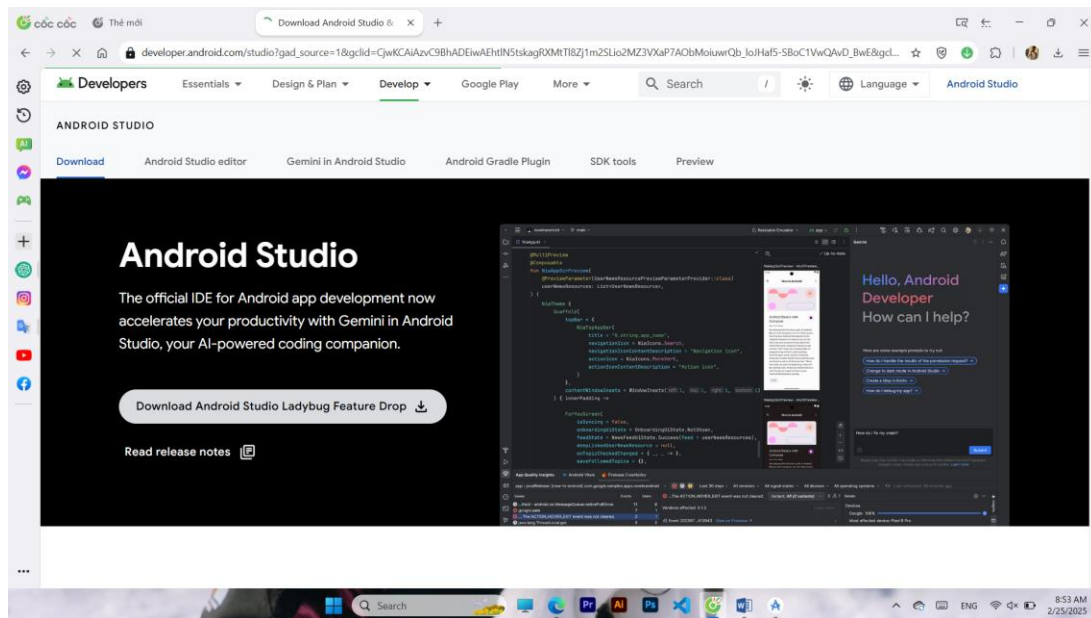
*Android Studio có sẵn cho các máy tính chạy Windows, Linux và macOS.*

Phiên bản mới nhất của OpenJDK (Java Development Kit) được tích hợp sẵn trong Android Studio.

Để bắt đầu với Android Studio, trước tiên hãy kiểm tra **yêu cầu hệ thống** để đảm bảo thiết bị của bạn đáp ứng đủ điều kiện. Quá trình cài đặt tương tự trên tất cả các nền tảng, chỉ có một số khác biệt nhỏ được ghi chú bên dưới.

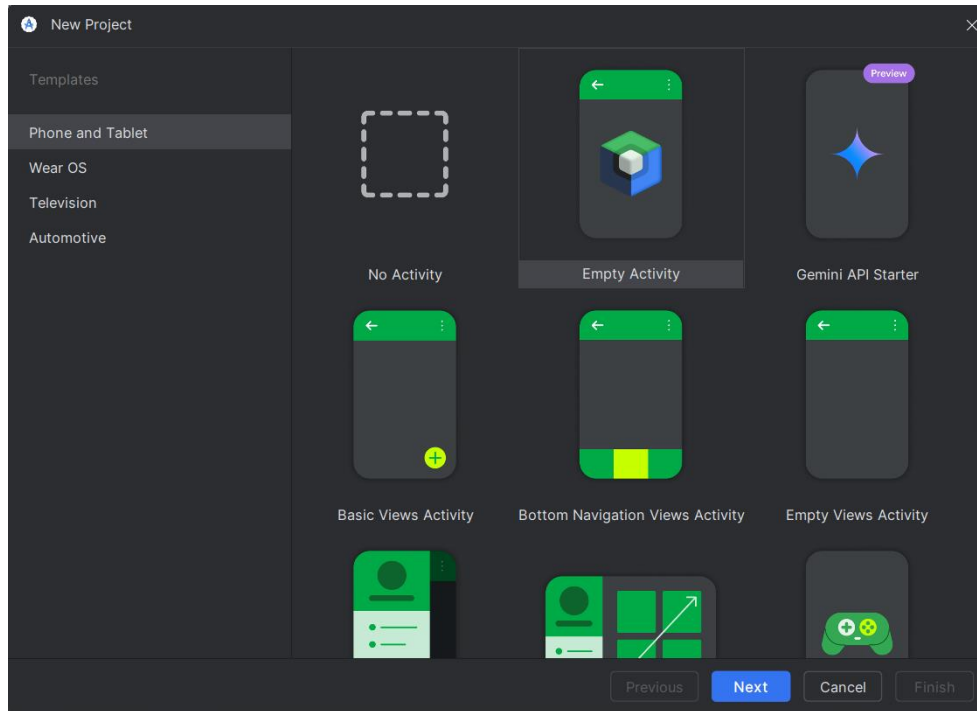
### Các bước cài đặt Android Studio:

1. Truy cập **trang web dành cho nhà phát triển Android** và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.



2. Chấp nhận các cấu hình mặc định trong tất cả các bước và đảm bảo rằng **tất cả các thành phần cần thiết** được chọn để cài đặt.
3. Sau khi hoàn tất cài đặt, **Trình hướng dẫn thiết lập (Setup Wizard)** sẽ tải xuống và cài đặt một số thành phần bổ sung, bao gồm **Android SDK**. Quá trình này có thể mất một khoảng thời gian tùy vào tốc độ Internet của bạn, và có thể một số bước sẽ trông giống nhau.

4. Khi quá trình tải xuống hoàn tất, **Android Studio sẽ khởi động**, và bạn đã sẵn sàng tạo dự án đầu tiên của mình.

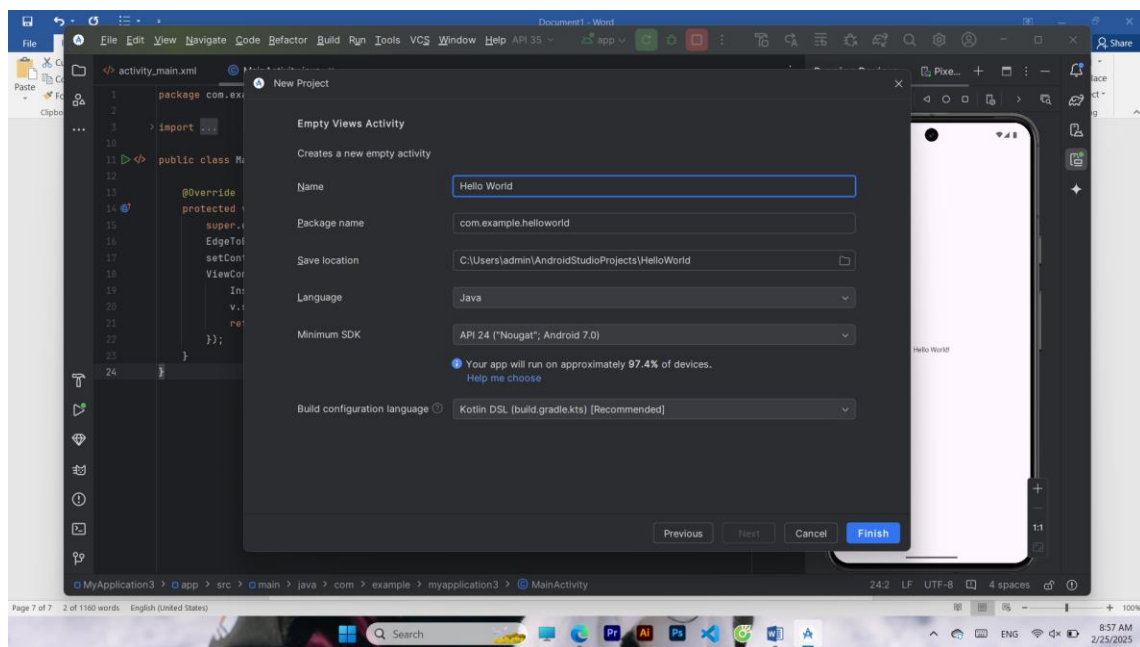


## Nhiệm vụ 2: Tạo ứng dụng "Hello World"

Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị dòng chữ **"Hello World"** để xác minh rằng Android Studio đã được cài đặt đúng cách, đồng thời làm quen với các bước phát triển ứng dụng cơ bản trong Android Studio.

### 2.1 Tạo dự án ứng dụng

1. **Mở Android Studio** nếu chưa mở.
2. Trong cửa sổ chính **"Welcome to Android Studio"**, nhấp vào **"Start a new Android Studio project"**.
3. Trong cửa sổ **"Create Android Project"**, nhập **"Hello World"** vào ô **Application name** (Tên ứng dụng).



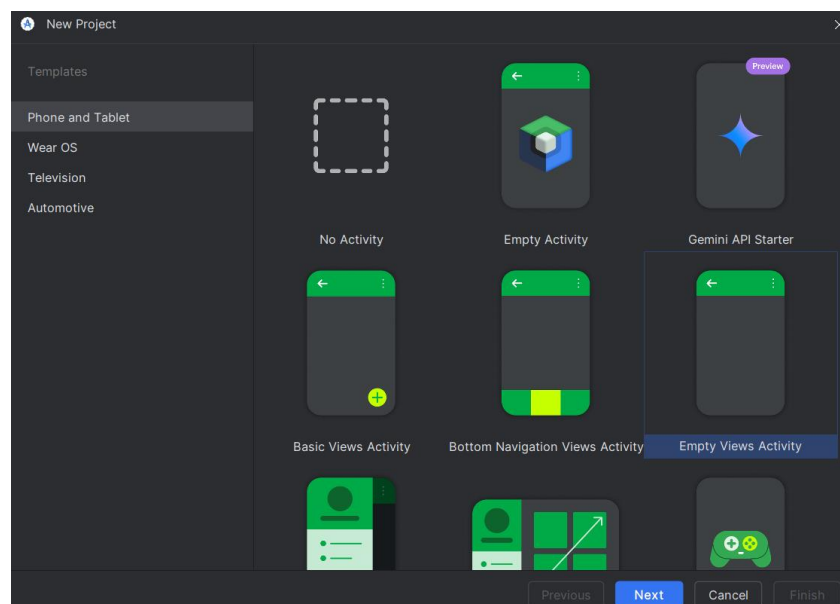
4. Xác nhận rằng **vị trí dự án mặc định (Project location)** là nơi bạn muốn lưu trữ ứng dụng **Hello World** và các dự án khác trong Android Studio. Nếu cần, hãy thay đổi sang thư mục bạn mong muốn.

Name	Hello World
Package name	com.example.helloworld
Save location	C:\Users\admin\AndroidStudioProjects\HelloWorld
Language	Java
Minimum SDK	API 24 ("Nougat"; Android 7.0)
<p>📘 Your app will run on approximately 97.4% of devices.</p>	



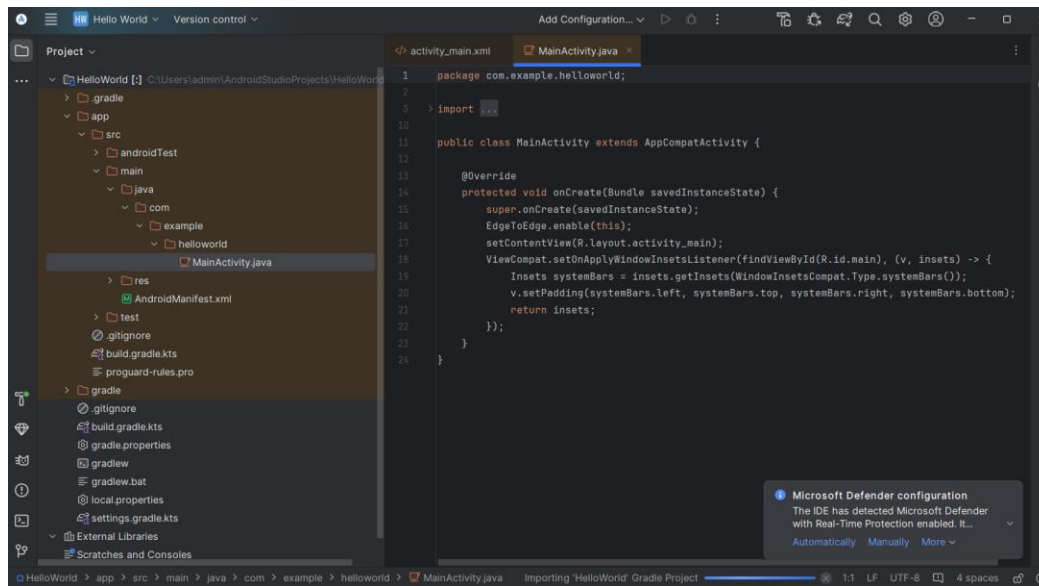
5. Chấp nhận giá trị mặc định android.example.com cho Company Domain hoặc tạo một miền công ty riêng.
  - Nếu bạn không có ý định xuất bản ứng dụng, bạn có thể giữ nguyên mặc định.
  - Hãy lưu ý rằng thay đổi tên gói (package name) sau này sẽ tốn thêm công sức.
6. Bỏ chọn các tùy chọn Include C++ support và Include Kotlin support, sau đó nhấn Next.
7. Trên màn hình Target Android Devices, hãy đảm bảo Phone and Tablet được chọn.
  - Đặt API 15: Android 4.0.3 IceCreamSandwich làm Minimum SDK.
  - Nếu API không đúng, hãy sử dụng menu thả xuống để chọn lại.
  - Tính đến thời điểm viết tài liệu này, các cài đặt này giúp ứng dụng **Hello World** của bạn **tương thích với 97% thiết bị Android** đang hoạt động trên **Google Play Store**.
8. **Bỏ chọn** tùy chọn **Include Instant App support** và tất cả các tùy chọn khác, sau đó nhấn **Next**.
  - Nếu dự án của bạn yêu cầu thêm các thành phần cho **target SDK** đã chọn, Android Studio sẽ tự động cài đặt chúng.
9. Màn hình **Add an Activity** sẽ xuất hiện.
  - **Activity** là một thành phần quan trọng của ứng dụng Android, đại diện cho một tác vụ cụ thể mà người dùng có thể thực hiện.
  - Mỗi **Activity** thường có một **layout** đi kèm, giúp xác định cách các phần tử giao diện người dùng (**UI elements**) hiển thị trên màn hình.
  - Android Studio cung cấp các **mẫu Activity** để giúp bạn bắt đầu nhanh chóng.

☞ **Đối với dự án Hello World**, hãy chọn **Empty Activity**, như hình bên dưới, rồi nhấn **Next**.



10. Màn hình **Configure Activity** sẽ xuất hiện (màn hình này có thể khác nhau tùy thuộc vào mẫu bạn đã chọn ở bước trước).

- Mặc định, **Activity** trông được cung cấp bởi mẫu được gọi là **MainActivity**.
- Bạn có thể thay đổi tên này nếu muốn, nhưng trong bài học này, chúng ta sẽ sử dụng **MainActivity**.



11. Đảm bảo rằng tùy chọn **Generate Layout file** được chọn. Tên mặc định của layout là **activity\_main**. Bạn có thể thay đổi tên này nếu muốn, nhưng trong bài học này, chúng ta sẽ sử dụng **activity\_main**.

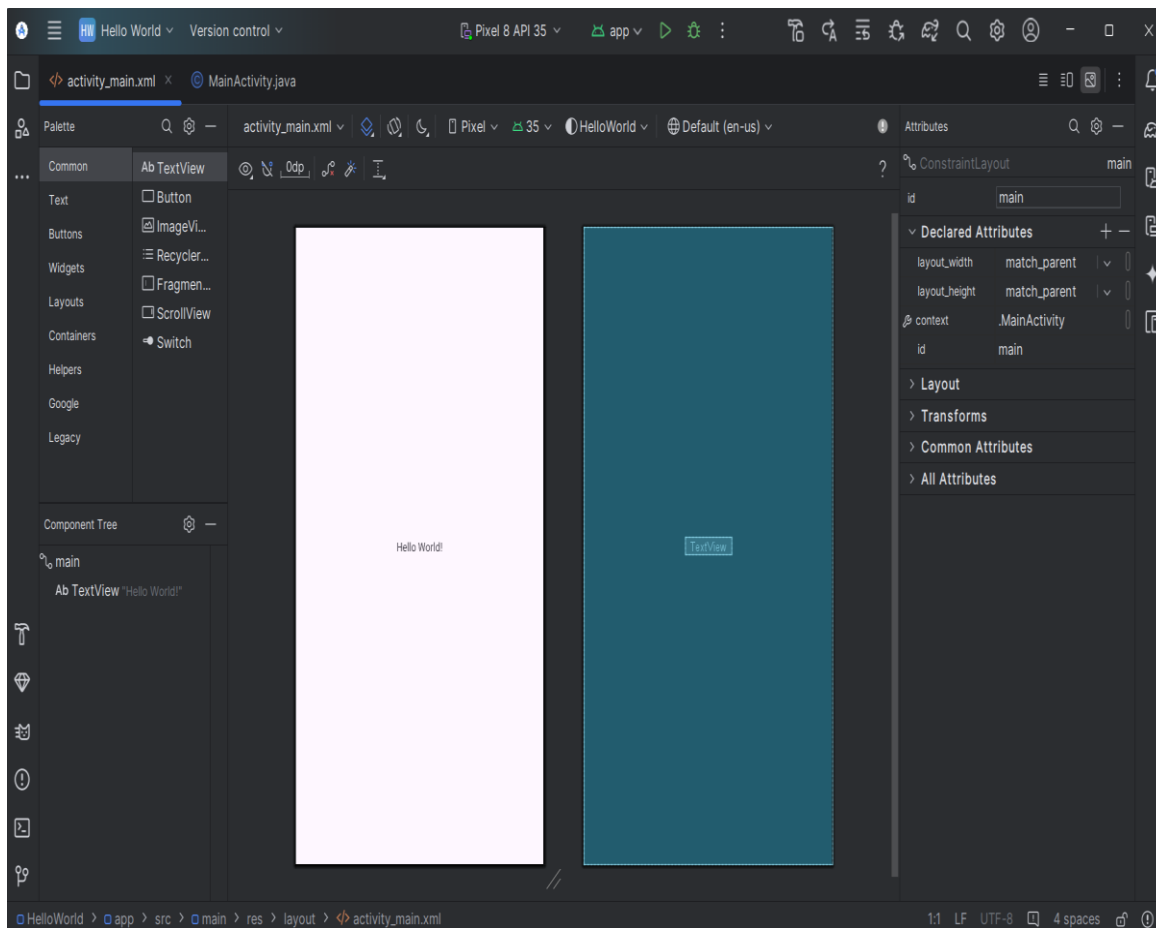
12. Đảm bảo rằng tùy chọn **Backwards Compatibility (App Compat)** được chọn. Điều này đảm bảo rằng ứng dụng của bạn sẽ tương thích ngược với các phiên bản Android trước.

13. Nhấn **Finish**.

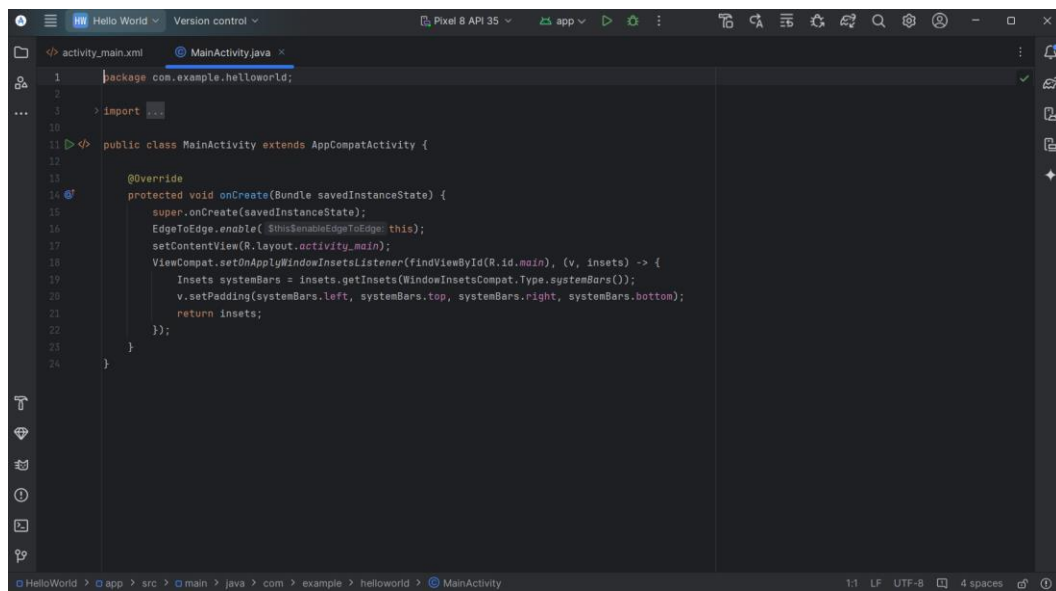
Android Studio sẽ tạo một thư mục cho dự án của bạn và xây dựng dự án bằng **Gradle** (quá trình này có thể mất một vài phút).

## Sử dụng Android Studio Editor

1. Nhấn vào thẻ **activity\_main.xml** để xem **layout editor**.
2. Nhấn vào thẻ **Design** trong **layout editor** (nếu chưa chọn), để xem một phiên bản đồ họa của layout như hình dưới đây.



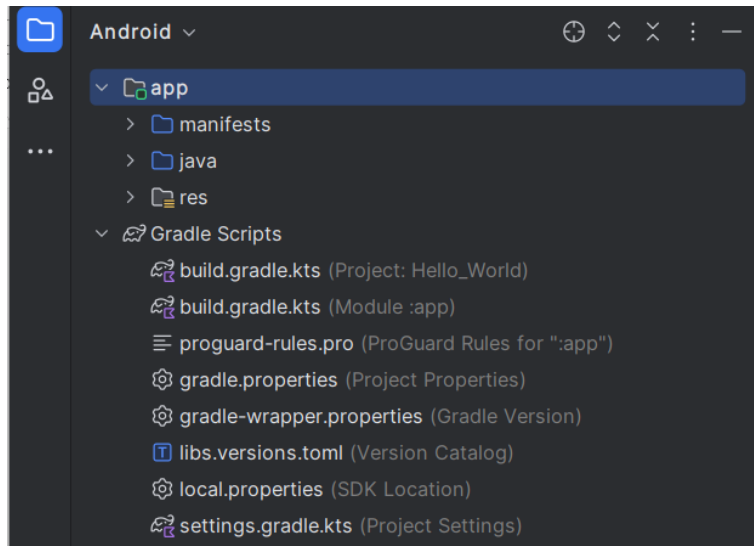
Nhấp vào thẻ **MainActivity.java** để xem **code editor** như hình dưới đây.



## 2.2 Khám phá Project > Android pane

Trong phần thực hành này, bạn sẽ khám phá cách dự án được tổ chức trong Android Studio.

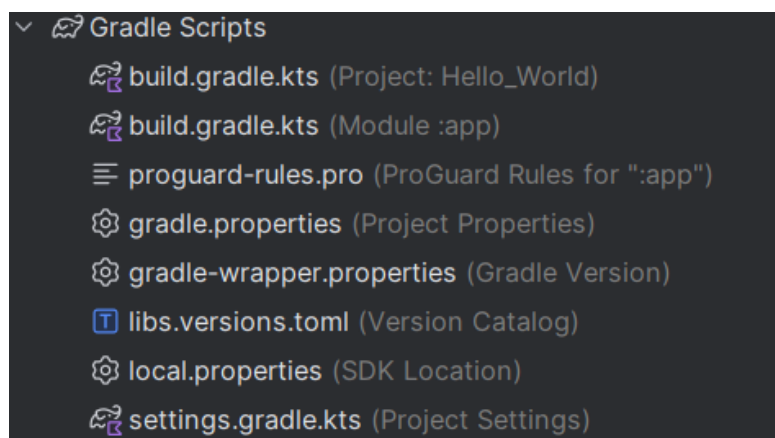
1. Nếu **Project tab** chưa được chọn, nhấp vào **Project** trong cột tab dọc ở phía bên trái của cửa sổ Android Studio. **Project pane** sẽ xuất hiện.
2. Để xem dự án theo cấu trúc dự án Android chuẩn, chọn **Android** từ menu thả xuống ở trên cùng của **Project pane**, như hình dưới đây.



## 2.3 Khám phá thư mục Gradle Scripts

Hệ thống xây dựng **Gradle** trong Android Studio giúp bạn dễ dàng bao gồm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào dự án của bạn dưới dạng các phụ thuộc (dependencies).

Khi bạn lần đầu tạo một dự án ứng dụng, **Project > Android pane** sẽ hiển thị với thư mục **Gradle Scripts** được mở rộng như hình dưới đây.



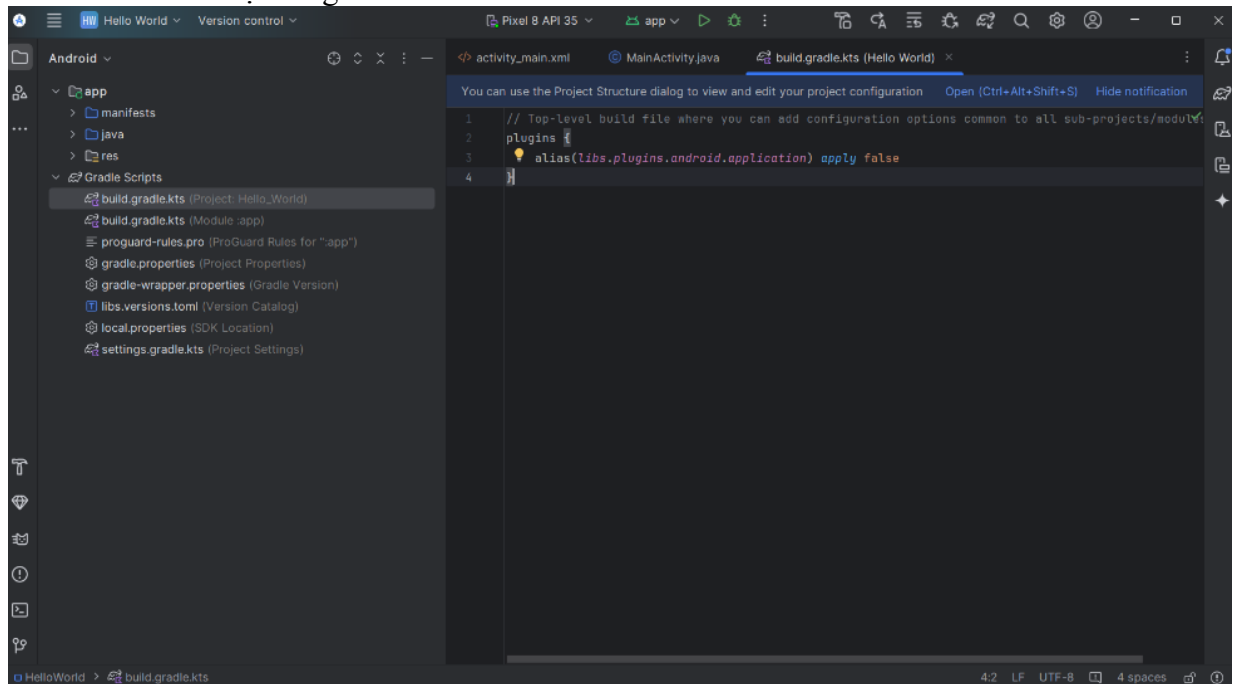
## Khám phá hệ thống Gradle

1. Nếu thư mục **Gradle Scripts** chưa được mở rộng, nhấp vào dấu **tam giác** để mở rộng nó. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng.

## 2. Tìm tệp **build.gradle (Project: HelloWorld)**.

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun tạo thành dự án của bạn.

- Mỗi dự án Android Studio đều có một tệp Gradle xây dựng ở cấp độ cao nhất.
- Hầu hết thời gian, bạn sẽ không cần thay đổi tệp này, nhưng vẫn hữu ích khi hiểu về nội dung của nó.



Theo mặc định, tệp build cấp cao nhất sử dụng khối **buildscript** để định nghĩa các kho lưu trữ và phụ thuộc Gradle chung cho tất cả các mô-đun trong dự án. Khi phụ thuộc của bạn không phải là thư viện hoặc cây tệp cục bộ, Gradle sẽ tìm các tệp trong các kho lưu trữ trực tuyến được chỉ định trong khối **repositories** của tệp này.

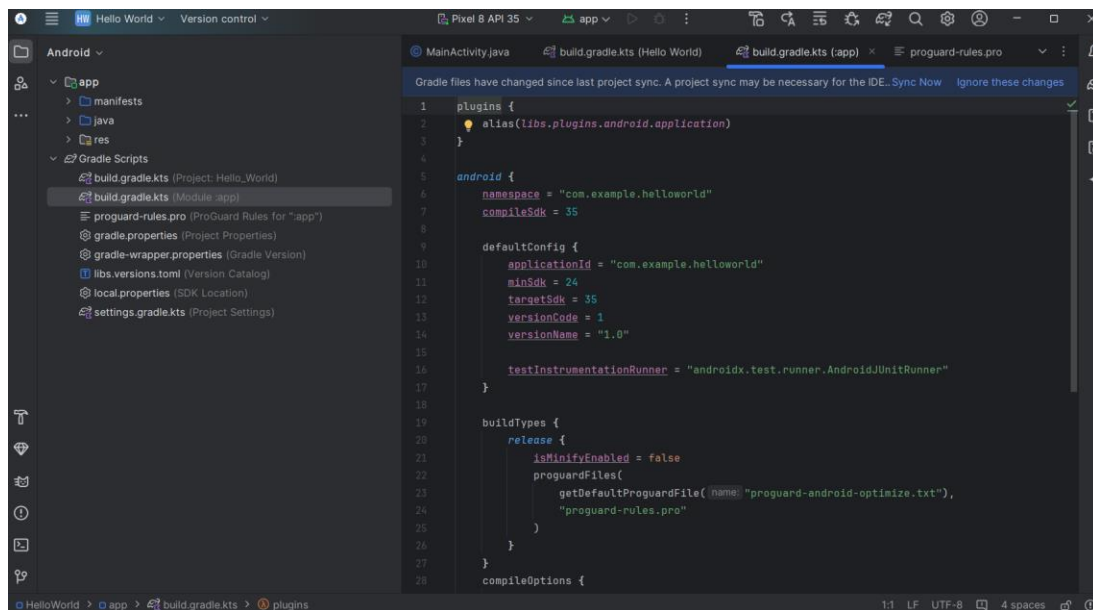
Theo mặc định, các dự án Android Studio mới khai báo **JCenter** và **Google** (bao gồm kho **Google Maven**) là vị trí kho lưu trữ:

```

1  // Top-level build file where you can add configuration
2  plugins {
3      alias(libs.plugins.android.application) apply false
4  }
5  allprojects {
6      repositories {
7          google()
8          jcenter()
9      }
10 }

```

### 3. Khám phá tệp build.gradle (Module:app)



Ngoài tệp **build.gradle** cấp dự án, mỗi mô-đun cũng có một tệp **build.gradle** riêng, cho phép bạn cấu hình các cài đặt build cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc cấu hình các cài đặt build này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại build (build types) bổ sung và các biến thể sản phẩm (product flavors). Bạn cũng có thể ghi đè các cài đặt trong tệp **AndroidManifest.xml** hoặc tệp **build.gradle** cấp cao nhất.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi các cấu hình cấp ứng dụng, chẳng hạn như khai báo các phụ thuộc trong phần **dependencies**. Bạn có thể khai báo một thư viện phụ thuộc bằng một trong số nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ, câu lệnh:

**implementation files('libs/\*.jar')**

Dưới đây là nội dung của tệp build.gradle(Module: app) cho ứng dụng HelloWorld:

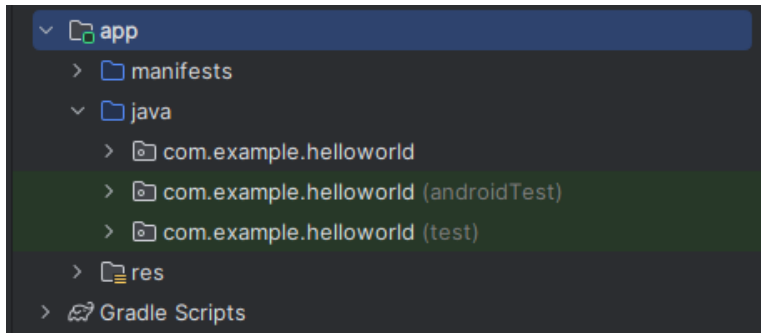
```
54 dependencies {
55
56     implementation(libs.appcompat)
57     implementation(libs.material)
58     implementation(libs.activity)
59     implementation(libs.constraintlayout)
60     testImplementation(libs.junit)
61     androidTestImplementation(libs.ext.junit)
62     androidTestImplementation(libs.espresso.core)
63     // Cập nhật androidx.appcompat lên phiên bản mới nhất
64     implementation("androidx.appcompat:appcompat:1.7.0")
65
66     // Cập nhật androidx.constraintlayout lên phiên bản mới nhất
67     implementation("androidx.constraintlayout:constraintlayout:2.2.0")
68
69     // Cập nhật JUnit lên phiên bản mới nhất
70     testImplementation("junit:junit:4.13.3")
71
72     // Cập nhật androidx.test.ext.junit lên phiên bản mới nhất
73     androidTestImplementation("androidx.test.ext:junit:1.1.6")
74
75     // Cập nhật espresso-core lên phiên bản mới nhất
76     androidTestImplementation("androidx.test.espresso:espresso-core:3.5.2")
77 }
```

4. Nhấp vào dấu **tam giác** để đóng **Gradle Scripts**.

## 2.4 Khám phá các thư mục app và res

Tất cả mã nguồn và tài nguyên của ứng dụng đều nằm trong các thư mục **app** và **res**.

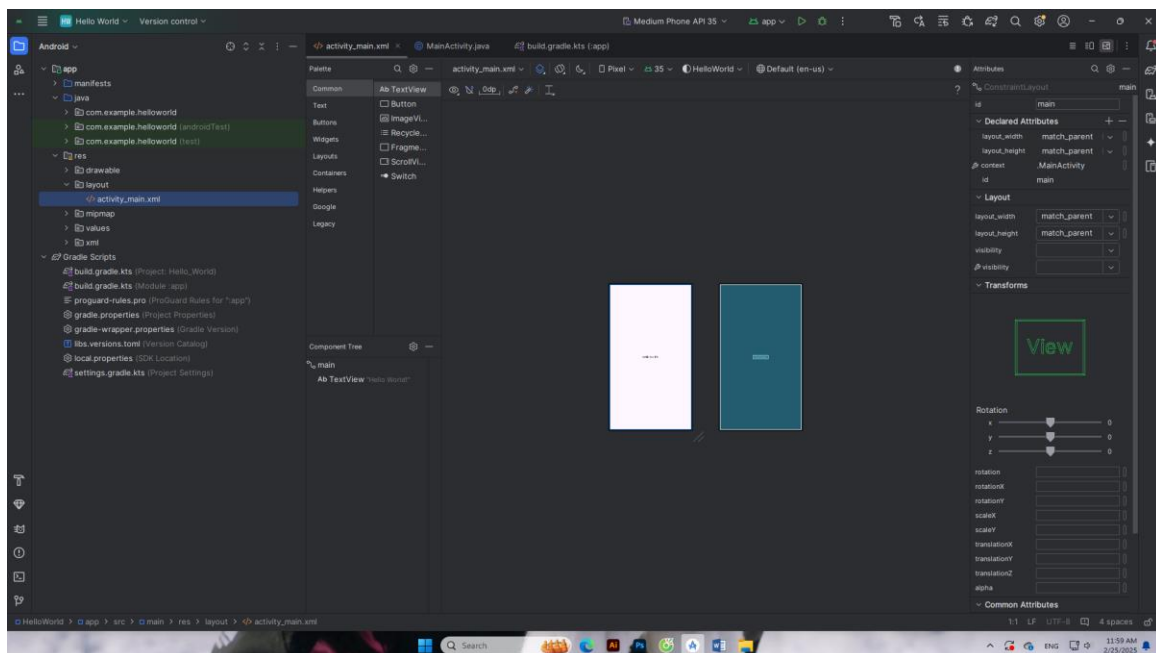
1. Mở rộng thư mục **app**, thư mục **java** và thư mục **com.example.android.helloworld** để xem tệp **java MainActivity**. Nhấp đúp vào tệp để mở tệp trong trình chỉnh sửa mã.



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như được chỉ ra trong hình trên.

Thư mục **com.example.hello.helloworld** (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp cho một gói ứng dụng. Hai thư mục còn lại được sử dụng cho mục đích kiểm tra và sẽ được mô tả trong bài học khác. Đối với ứng dụng Hello World, chỉ có một gói và nó chứa tệp **MainActivity.java**. Tên của **Activity** đầu tiên (màn hình mà người dùng nhìn thấy) cũng khởi tạo các tài nguyên toàn cục của ứng dụng, thường được gọi là **MainActivity** (phần mở rộng tệp bị bỏ qua trong bảng **Project > Android**).

2. Mở rộng thư mục **res** và thư mục **layout**, sau đó nhấp đúp vào tệp **activity\_main.xml** để mở nó trong **layout editor**.



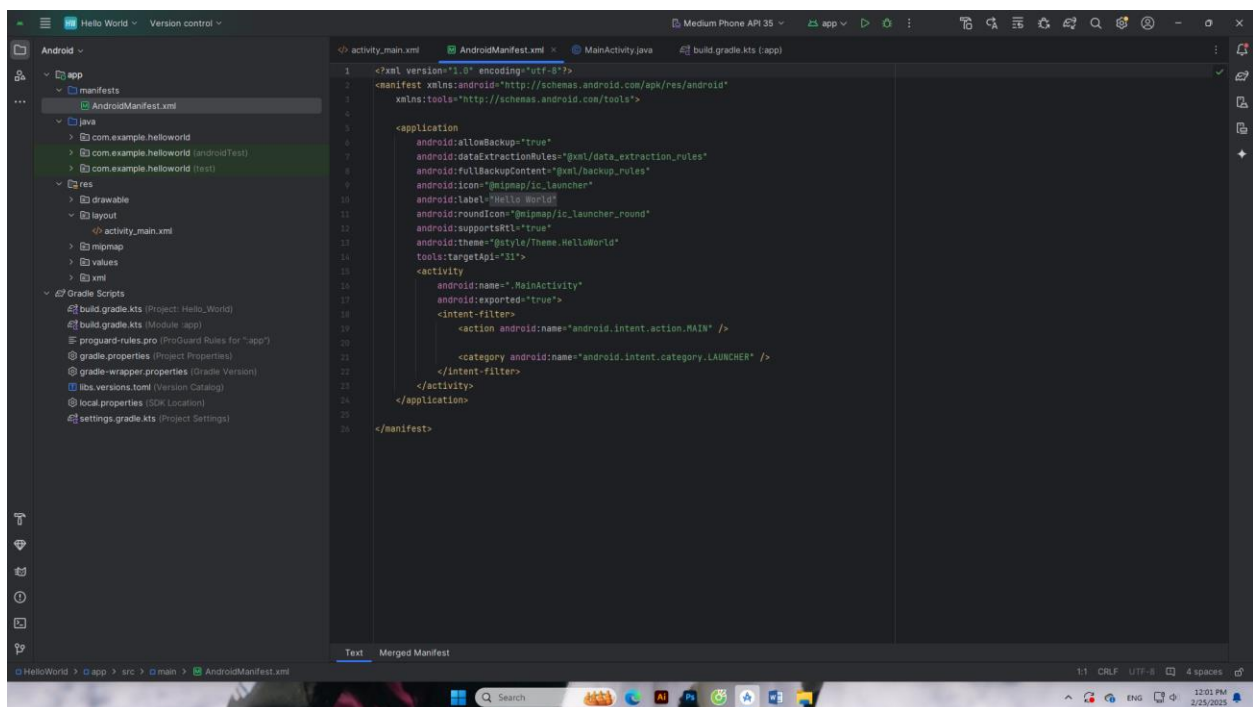


Thư mục **res** chứa các tài nguyên, chẳng hạn như các bố cục (layouts), chuỗi (strings), và hình ảnh (images). Một **Activity** thường được liên kết với một bố cục giao diện người dùng (UI) được định nghĩa dưới dạng một tệp XML. Tệp này thường có tên giống với tên của **Activity**.

## 2.5 Khám phá thư mục manifests

Thư mục **manifests** chứa các tệp cung cấp thông tin quan trọng về ứng dụng của bạn cho hệ thống Android, những thông tin này mà hệ thống cần có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục **manifests**.
2. Mở tệp **AndroidManifest.xml**.



Tệp **AndroidManifest.xml** mô tả tất cả các thành phần của ứng dụng Android của bạn. Mọi thành phần trong ứng dụng, chẳng hạn như mỗi **Activity**, phải được khai báo trong tệp XML này. Trong các bài học sau, bạn sẽ chỉnh sửa tệp này để thêm các tính năng và quyền truy cập tính năng.

## Nhiệm vụ 3: Sử dụng thiết bị ảo (emulator)

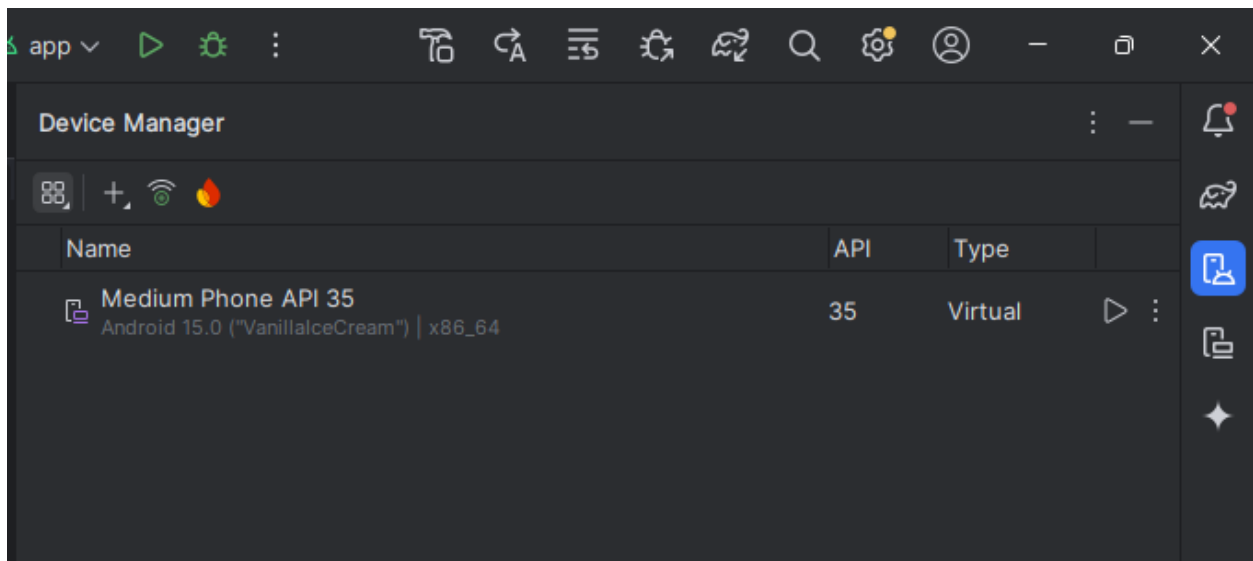
Trong nhiệm vụ này, bạn sẽ sử dụng **Android Virtual Device (AVD) Manager** để tạo một thiết bị ảo (hay còn gọi là emulator) mô phỏng cấu hình của một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng **Android Emulator** có những yêu cầu bổ sung ngoài các yêu cầu hệ thống cơ bản của **Android Studio**.

Sử dụng **AVD Manager**, bạn định nghĩa các đặc điểm phần cứng của thiết bị, mức API, bộ nhớ, skin và các thuộc tính khác và lưu lại dưới dạng một thiết bị ảo. Với các thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các cấu hình thiết bị khác nhau (chẳng hạn như máy tính bảng và điện thoại) với các mức API khác nhau mà không cần sử dụng thiết bị vật lý.

### 3.1 Tạo một thiết bị ảo Android (AVD)

Để chạy một emulator trên máy tính của bạn, bạn cần tạo một cấu hình mô tả thiết bị ảo.

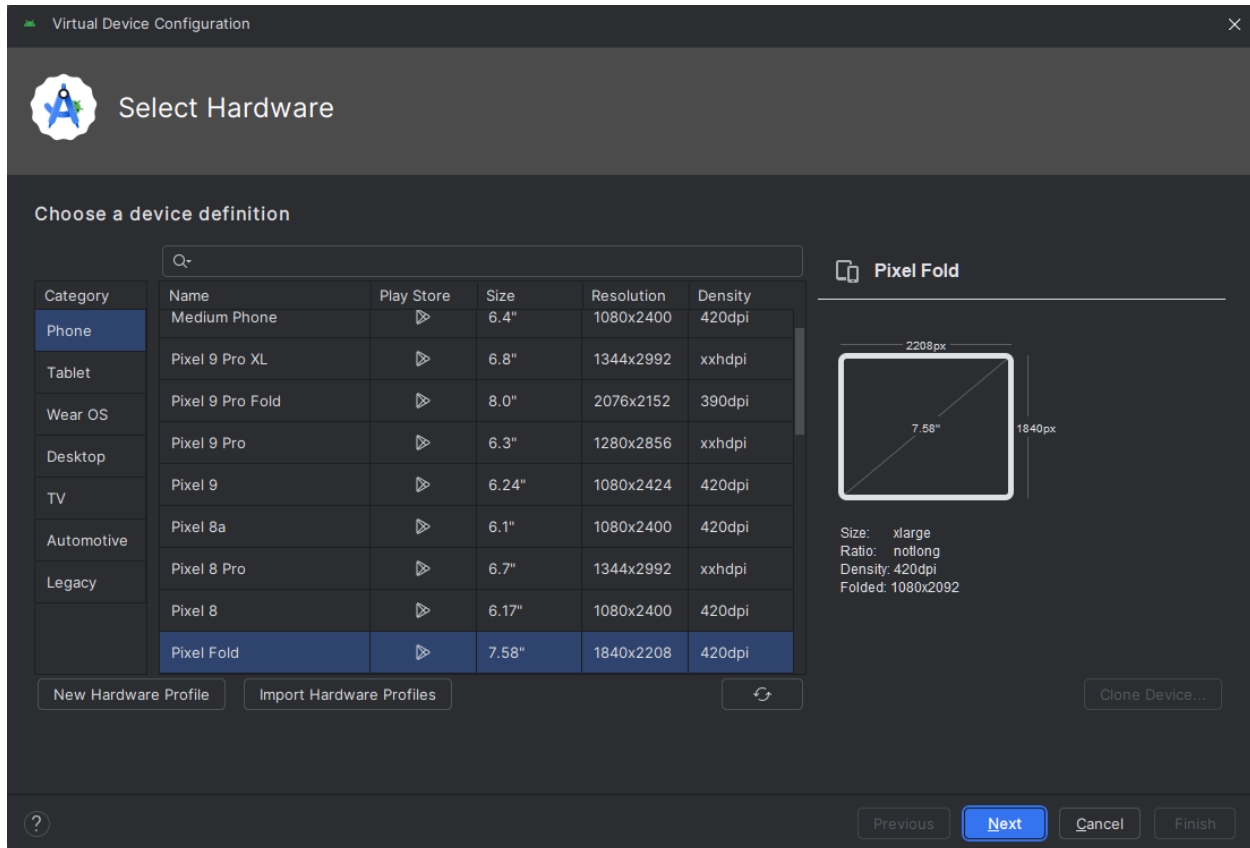
1. Trong **Android Studio**, **Device Manager**, hoặc nhấp vào biểu tượng **AVD Manager** trên thanh công cụ. Màn hình **Device Manager** sẽ xuất hiện.
  - Nếu bạn đã tạo các thiết bị ảo, màn hình sẽ hiển thị chúng (như hình dưới đây); nếu chưa, bạn sẽ thấy một danh sách trống.



2. Nhấp vào **+Create Virtual Device**. Cửa sổ **Select Hardware** sẽ xuất hiện, hiển thị danh sách các thiết bị phần cứng đã được cấu hình sẵn. Đối với mỗi thiết bị, bảng sẽ cung cấp các cột thông tin về:
  - **Size**: Kích thước đường chéo của màn hình.
  - **Resolution**: Độ phân giải màn hình tính bằng pixel.
  - **Density**: Mật độ điểm ảnh (pixel density).

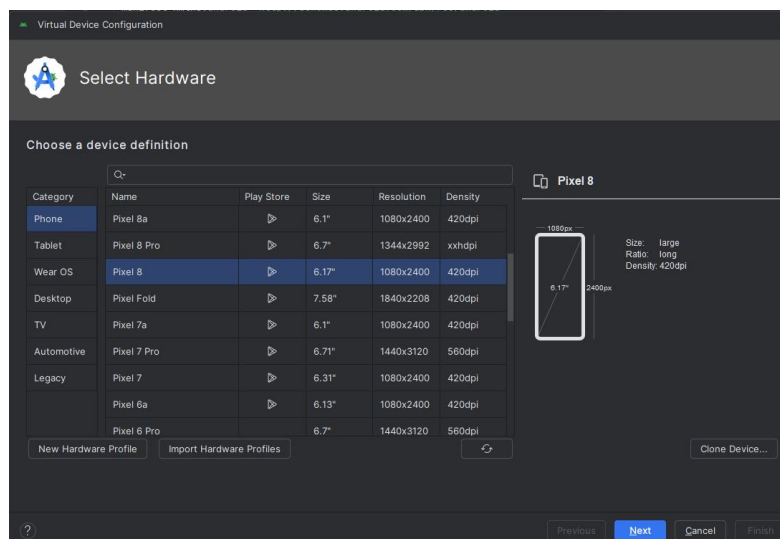


Từ cửa sổ này, bạn có thể chọn một thiết bị mà bạn muốn mô phỏng cho thiết bị ảo của mình.

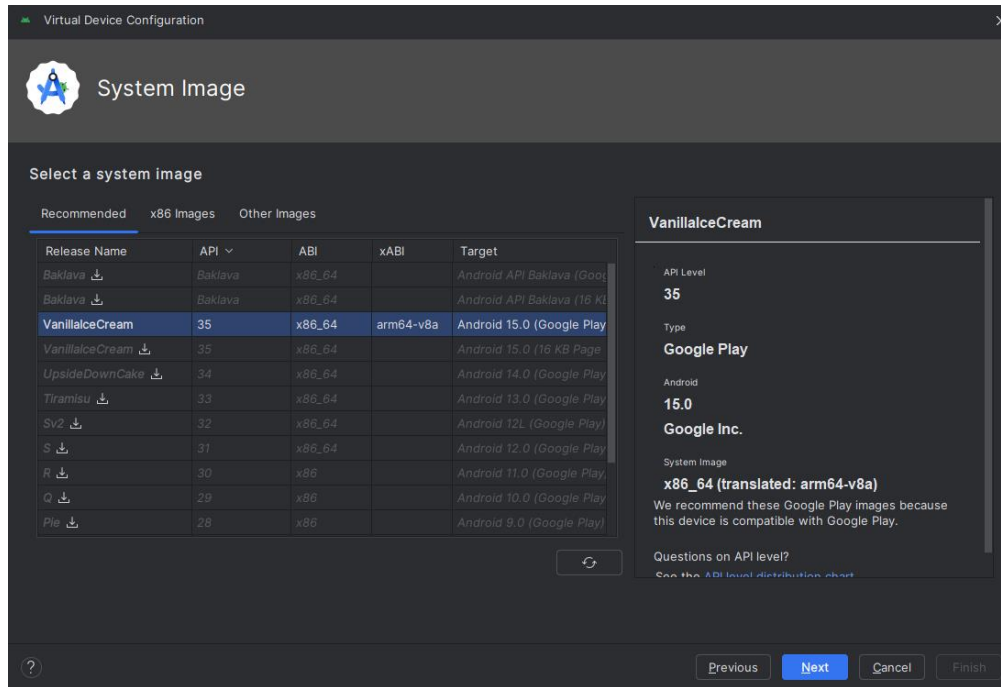


### 3. Chọn thiết bị và cài đặt hệ điều hành cho thiết bị ảo

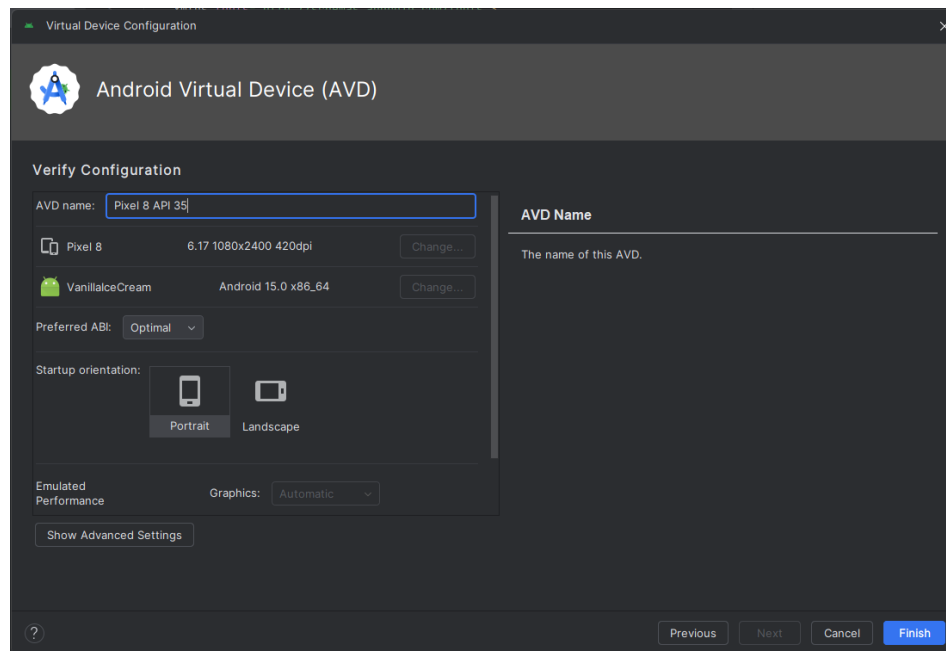
1. Chọn một thiết bị như **Pixel 8** hoặc **Pixel XL**, rồi nhấp vào **Next**. Màn hình **System Image** sẽ xuất hiện.



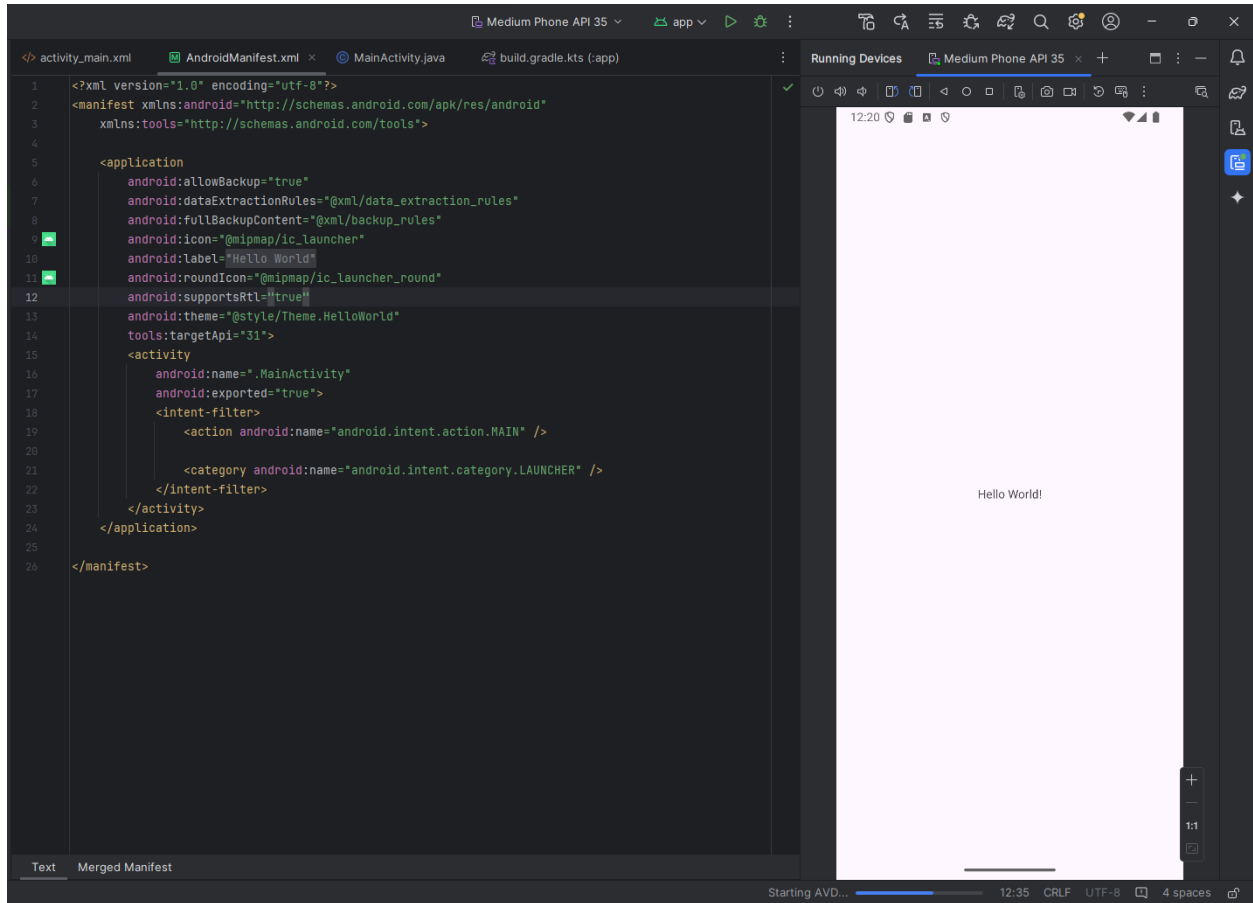
2. Nhấp vào tab **Recommended** (nếu chưa được chọn) và chọn phiên bản **Android** mà bạn muốn chạy trên thiết bị ảo (ví dụ, **Oreo**).



3. Nếu có một liên kết **Download** hiển thị bên cạnh một hệ điều hành mà bạn muốn sử dụng, điều đó có nghĩa là hệ điều hành đó chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải về, và nhấp vào **Finish** khi tải xong.
4. Sau khi chọn hệ điều hành, nhấp vào **Next**. Màn hình **Android Virtual Device (AVD)** sẽ xuất hiện.



Sau khi hoàn thành, thiết bị ảo sẽ được tạo và bạn có thể sử dụng nó để chạy và kiểm tra ứng dụng của mình.



## Task 4: (Tùy chọn) Sử dụng thiết bị vật lý

Trong nhiệm vụ cuối cùng này, bạn sẽ chạy ứng dụng của mình trên một thiết bị di động vật lý như điện thoại hoặc máy tính bảng. Bạn nên luôn kiểm tra ứng dụng của mình trên cả thiết bị ảo và thiết bị vật lý.

*Những gì bạn cần:*

- Một thiết bị Android như điện thoại hoặc máy tính bảng.
- Cáp dữ liệu để kết nối thiết bị Android của bạn với máy tính thông qua cổng USB.
- Nếu bạn đang sử dụng hệ thống Linux hoặc Windows, có thể bạn sẽ cần thực hiện thêm các bước để chạy trên thiết bị phần cứng. Tham khảo tài liệu **Using Hardware Devices**. Bạn cũng có thể cần cài đặt driver USB thích hợp cho thiết bị của mình. Đối với driver USB trên hệ điều hành Windows, tham khảo **OEM USB Drivers**.

### 4.1 Bật chế độ USB Debugging

Để cho phép Android Studio giao tiếp với thiết bị của bạn, bạn phải bật **USB Debugging** trên thiết bị Android của mình. Tính năng này được kích hoạt trong phần **Developer options** trên thiết bị của bạn.

Trên các thiết bị Android từ phiên bản 4.2 trở lên, màn hình **Developer options** mặc định bị ẩn. Để hiển thị các tùy chọn phát triển và bật **USB Debugging**, làm theo các bước sau:

1. Trên thiết bị của bạn, mở **Settings** (Cài đặt), tìm kiếm **About phone** (Giới thiệu về điện thoại), nhấp vào **About phone**, sau đó chạm vào **Build number** bảy lần.
2. Quay lại màn hình trước đó (**Settings** / **System**). **Developer options** sẽ xuất hiện trong danh sách. Nhấn vào **Developer options**.
3. Chọn **USB Debugging** và bật nó lên.

Sau khi bật chế độ USB Debugging, bạn có thể kết nối thiết bị của mình với máy tính và chạy ứng dụng trực tiếp trên thiết bị vật lý.

### 4.2 Chạy ứng dụng của bạn trên thiết bị

Giờ đây, bạn có thể kết nối thiết bị của mình và chạy ứng dụng từ Android Studio.

1. Kết nối thiết bị của bạn với máy phát triển bằng cáp USB.
2. Nhấp vào nút **RUN** trên thanh công cụ. Cửa sổ **Select Deployment Target** mở ra với danh sách các trình giả lập có sẵn và các thiết bị được kết nối.
3. Chọn thiết bị của bạn và nhấp vào **OK**.

Android Studio cài đặt và chạy ứng dụng trên thiết bị của bạn.

### *Troubleshooting*

Nếu Android Studio không nhận dạng thiết bị của bạn, hãy thử các cách sau:

1. Rút phích cắm và cắm lại thiết bị của bạn.
2. Khởi động lại Android Studio.

Nếu máy tính của bạn vẫn không tìm thấy thiết bị hoặc tuyên bố thiết bị là "không được phép", hãy làm theo các bước sau:

1. Rút phích cắm của thiết bị.
2. Trên thiết bị, mở **Developer Options in Settings app**.
3. Nhấn vào **USB Debugging** gỡ lỗi USB.
4. Kết nối lại thiết bị với máy tính của bạn.
5. Khi được nhắc, hãy cấp ủy quyền.

Bạn có thể cần cài đặt trình điều khiển USB thích hợp cho thiết bị của mình. Xem [tài liệu về Thiết bị phần cứng U sing](#).



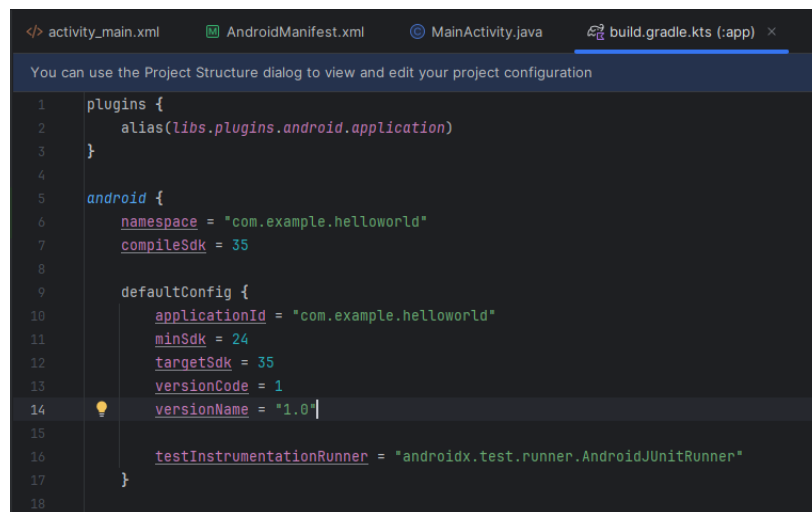
## Nhiệm vụ 5: Thay đổi cấu hình Gradle của ứng dụng

Trong nhiệm vụ này, bạn sẽ thay đổi điều gì đó về cấu hình ứng dụng trong `build.gradle(Module:app)` để tìm hiểu cách thực hiện các thay đổi và đồng bộ hóa chúng với dự án Android Studio của bạn.

### 5.1 Thay đổi phiên bản SDK tối thiểu cho ứng dụng

Làm theo các bước sau:

1. Mở rộng thư mục Gradle Scripts nếu nó chưa được mở và nhấp đúp vào tập tin `build.gradle(Module:app)`.
2. Nội dung của tệp xuất hiện trong trình soạn thảo mã.
3. Trong khối `defaultConfig`, thay đổi giá trị của `minSdkVersion` thành 17 như hình bên dưới (ban đầu nó được đặt thành 15). (Giữ nguyên giá trị default của nó là 24)



```
1 plugins {
2     alias(libs.plugins.android.application)
3 }
4
5 android {
6     namespace = "com.example.helloworld"
7     compileSdk = 35
8
9     defaultConfig {
10         applicationId = "com.example.helloworld"
11         minSdk = 24
12         targetSdk = 35
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18 }
```

Trình chỉnh sửa mã hiển thị thanh thông báo ở trên cùng với **liên kết** Sync Now.

### 5.2 Đồng bộ hóa cấu hình Gradle mới

Khi bạn thực hiện các thay đổi đối với các tệp cấu hình bản dựng trong một dự án, Android Studio yêu cầu bạn thực hiện *các* tệp dự án để có thể nhập các thay đổi về cấu hình bản dựng và chạy một số kiểm tra để đảm bảo cấu hình sẽ không tạo ra lỗi bản dựng.

Để đồng bộ hóa các tệp dự án, hãy nhấp vào **Sync Now** trong thanh thông báo xuất hiện khi thực hiện thay đổi

(như trong hình trước) hoặc nhấp vào **biểu tượng** Dự án Sync với Tệp Gradle  trên thanh công cụ.

Khi quá trình đồng bộ hóa Gradle hoàn tất, thông báo Gradle build completed sẽ xuất hiện ở góc dưới cùng bên trái của cửa sổ Android Studio.

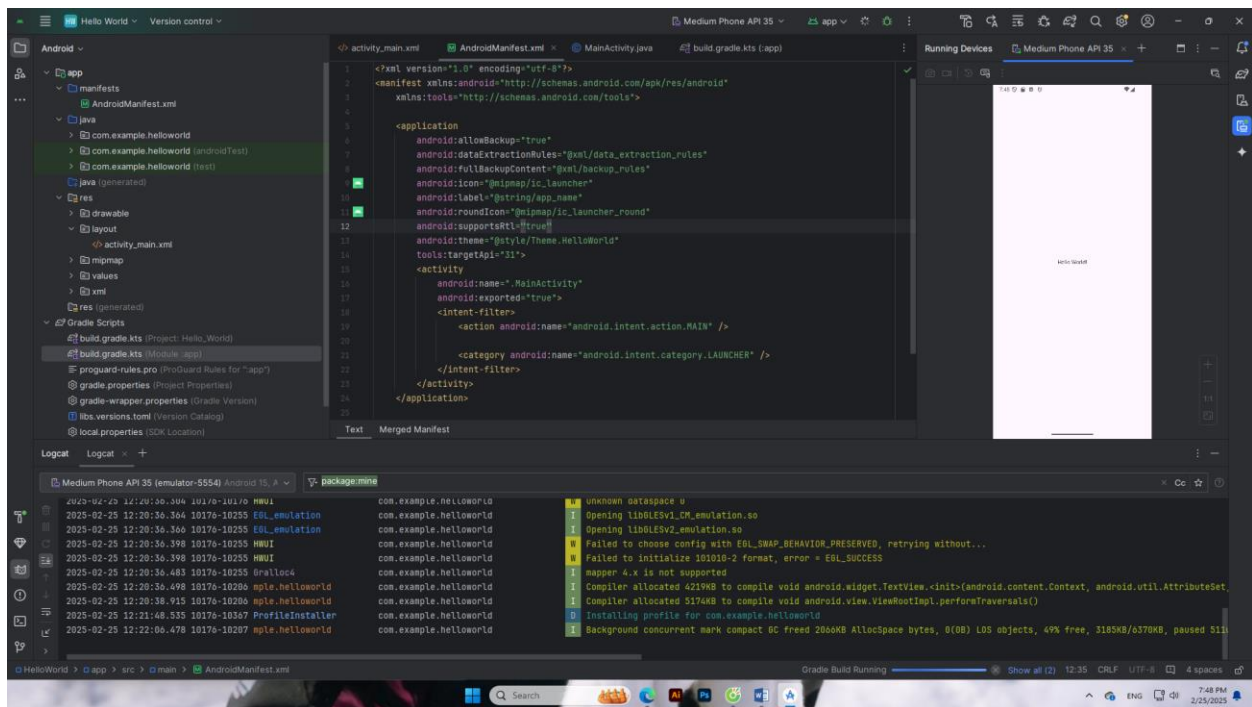
Để hiểu rõ hơn về Gradle, hãy xem tài liệu về [Tổng quan về hệ thống Build](#) và [Cài đặt về bản dựng Gradle](#).

## Nhiệm vụ 6: Thêm câu lệnh nhật ký vào ứng dụng của bạn

Trong nhiệm vụ này, bạn sẽ thêm [câu lệnh Log](#) vào ứng dụng của mình, câu lệnh này hiển thị thông báo trong **ngăn Logcat**. Thông báo nhật ký là một công cụ gỡ lỗi mạnh mẽ mà bạn có thể sử dụng để kiểm tra các giá trị, đường dẫn thực thi và báo cáo ngoại lệ.

### 6.1 View ngăn Logcat

Để xem ngăn **Logcat**, hãy nhấp vào tab **Logcat** ở cuối cửa sổ Android Studio như trong hình bên dưới.



Trong hình trên:

1. Tab **Logcat** để mở và đóng ngăn **Logcat**, hiển thị thông tin về ứng dụng của bạn khi ứng dụng đang chạy. Nếu bạn thêm câu lệnh `Log` vào ứng dụng của mình, thông báo `Log` sẽ xuất hiện ở đây.
2. Menu mức `Log` được đặt thành **Verbose** (mặc định), hiển thị tất cả các thông báo `Log`. Các cài đặt khác bao gồm **Debug**, **Error**, **Info** và **Warn**.

## 6.2 Thêm câu lệnh nhật ký vào ứng dụng của bạn

Câu lệnh nhật ký trong mã ứng dụng của bạn sẽ hiển thị thông báo trong ngăn Logcat. Chẳng hạn:

```
Log.d("Hoạt động chính", "Xin chào thế giới");
```

Các phần của thông điệp là:

- Log: Lớp [Log](#) để gửi tin nhắn nhật ký đến ngăn Logcat.
- d: Cài đặt cấp độ **Debug** Log để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Các mức log khác là e cho **Error**, w cho **Warn** và i cho **Info**.
- "MainActivity": Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của Activity mà thông điệp bắt nguồn từ đó. Tuy nhiên, bạn có thể làm cho điều này bất cứ thứ gì hữu ích cho bạn để gỡ lỗi. Theo quy ước, các thẻ nhật ký được định nghĩa là hằng số cho khoảng cách A:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- "Xin chào thế giới": Đối số thứ hai là thông điệp thực tế.

Làm theo các bước sau:

1. Mở ứng dụng Hello World của bạn trong Android studio và mở MainActivity.
2. Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như `Android.util.Log` cần thiết để sử dụng `Log`), hãy chọn **Cài đặt** > Windows hoặc **Tùy chọn** > **ndroid Studio** trong macOS.
3. Chọn **Editor** > **General** > **Auto Import**. Chọn tất cả các hộp kiểm và đặt **Insert imports on paste** thành **II**.
4. Nhấp vào **Apply** và sau đó nhấp vào **OK**.
5. Trong phương thức `onCreate()` của MainActivity, thêm câu lệnh sau:

```
Log.d("MainActivity", "Hello World");
```

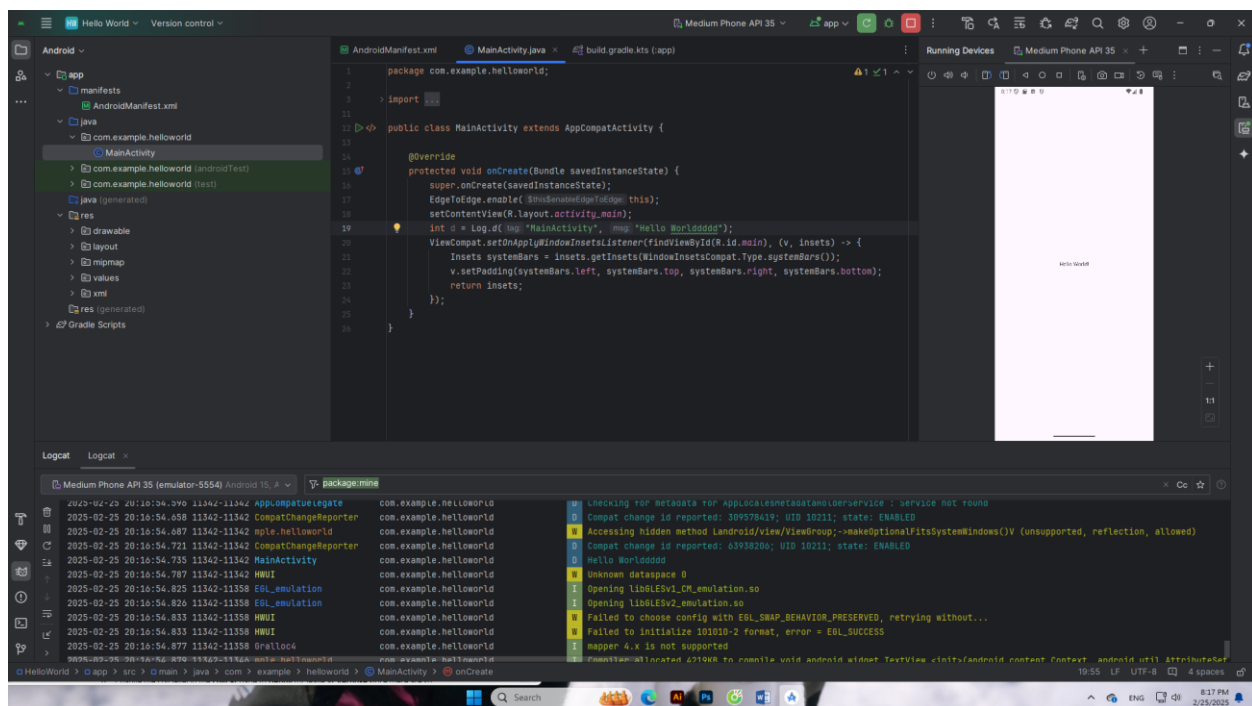
Phương thức `OnCreate()` bây giờ sẽ trông giống như mã sau:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("MainActivity", "Hello World");
}
```

6. Nếu ngăn Logcat chưa mở, hãy nhấp vào tab **Logcat** ở cuối Android Studio để mở.
7. Kiểm tra xem tên của mục tiêu và tên gói của ứng dụng có chính xác không.
8. Thay đổi mức Log trong ngăn **Logcat** thành **Debug** (hoặc để nguyên **Verbose** vì có rất ít thông báo nhật ký).
9. Chạy ứng dụng của bạn.

Thông báo sau sẽ xuất hiện trong ngăn Logcat:

11-24 14:06:59.001 4696-4696/? D/MainActivity: Hello World



Thử thách mã hóa

**Lưu ý:** Tất cả các thử thách mã hóa đều là tùy chọn và không phải là điều kiện tiên quyết cho các bài học sau này.

**Thử thách:** Nếu bạn đã thiết lập và quen thuộc với quy trình phát triển cơ bản, hãy làm như sau:

1. Tạo một dự án mới trong Android Studio.
2. Thay đổi lời chào "Hello World" thành "Chúc mừng sinh nhật " và tên của ai đó có sinh nhật gần đây.

3. (Tùy chọn) Chụp ảnh màn hình ứng dụng đã hoàn thành của bạn và gửi email cho người mà bạn quên ngày sinh.
4. Một cách sử dụng phổ biến của lớp Log là ghi nhật ký các ngoại lệ Java khi chúng xảy ra trong chương trình của bạn.

Có một số phương thức hữu ích, chẳng hạn như Log.e(), mà bạn có thể sử dụng cho mục đích này. Khám phá các phương thức bạn có thể sử dụng để bao gồm ngoại lệ với thông báo Log. Sau đó, viết mã trong ứng dụng của bạn để kích hoạt và ghi lại một ngoại lệ.

## Tóm tắt

- Để cài đặt Android Studio, hãy truy cập **Android Studio** và làm theo hướng dẫn để tải xuống và cài đặt nó.
- Khi tạo một ứng dụng mới, hãy đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được đặt làm SDK tối thiểu.
- Để xem hệ thống phân cấp Android của ứng dụng trong ngăn Dự án, hãy bấm vào **tab Project** trong cột tab dọc, sau đó chọn **ndroid** trong menu bật lên ở trên cùng.
- Chỉnh sửa tệp build.gradle(Module:app) khi bạn cần thêm thư viện mới vào dự án hoặc thay đổi phiên bản thư viện.
- Tất cả mã và tài nguyên cho ứng dụng nằm trong thư mục app và res. Thư mục java bao gồm các hoạt động, kiểm tra và các thành phần khác trong mã nguồn Java. Thư mục res chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh.
- Chỉnh sửa tệp AndroidManifest.xml để thêm các tính năng, thành phần và quyền vào ứng dụng Android của bạn. Tất cả các thành phần cho một ứng dụng, chẳng hạn như nhiều hoạt động, phải được khai báo trong tệp XML này.
- Sử dụng trình quản lý Thiết bị ảo (AVD) ndroid để tạo thiết bị ảo (còn được gọi là trình mô phỏng) để chạy ứng dụng của bạn.
- Thêm câu lệnh Log vào ứng dụng của bạn, câu lệnh này hiển thị thông báo trong ngăn Logcat như một công cụ cơ bản để gỡ lỗi.
- Để chạy ứng dụng của bạn trên thiết bị Android vật lý bằng Android Studio, hãy bật tính năng Gỡ lỗi USB trên thiết bị. Mở **Settings > About phone** và nhấn **Build** số bảy lần. Quay lại màn hình trước đó (**Settings**) và nhấn vào **Developer options**. Chọn **gỡ lỗi USB**.

## Các khái niệm liên quan

Tài liệu khái niệm liên quan có trong 1.0: Giới thiệu về Android và 1.1 Ứng dụng Android đầu tiên của bạn.

Tìm hiểu thêm

Tài liệu Android Studio:

- Trang tải xuống Android Studio
- Ghi chú phát hành Android Studio
- Làm quen với Android Studio
- Công cụ dòng lệnh Logcat
- Trình quản lý thiết bị ảo Android (AVD)
- Tổng quan về tệp kê khai ứng dụng
- Định cấu hình bản dựng của bạn
- Lốp nhật ký
- Tạo và quản lý thiết bị ảo Khác:
- Làm cách nào để cài đặt Java?
- Cài đặt phần mềm JDK và JAVA\_HOME cài đặt
- Trang web Gradle
- Cú pháp Apache Groovy
- Trang Wikipedia của Gradle

## Homework

### Xây dựng và chạy ứng dụng

- Tạo một dự án Android mới từ Empty Template.
- Thêm câu lệnh ghi nhật ký cho các cấp độ nhật ký khác nhau trong onCreate() trong hoạt động chính.
- Tạo trình mô phỏng cho thiết bị, nhắm mục tiêu bất kỳ phiên bản Android nào bạn thích và chạy ứng dụng.
- Sử dụng tính năng lọc trong **Logcat** để tìm các câu lệnh nhật ký của bạn và điều chỉnh các cấp độ để chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi nhật ký lỗi.

Trả lời những câu hỏi này

*Câu hỏi 1*

Tên của tệp bố cục cho hoạt động chính là gì?

- MainActivity.java

- AndroidManifest.xml
- activity\_main.xml
- xây dựng.gradle

### *Câu hỏi 2*

Tên của tài nguyên chuỗi chỉ định tên của ứng dụng là gì?

- app\_name
- xmlns:ứng dụng
- android:tên
- applicationId

### *Câu hỏi 3*

Bạn sử dụng công cụ nào để tạo trình giả lập mới?

- Màn hình thiết bị Android
- Trình quản lý AVD
- Trình quản lý SDK • Trình chỉnh sửa chủ đề

### *Câu hỏi 4*

Giả sử rằng ứng dụng của bạn bao gồm câu lệnh ghi nhật ký sau:

```
Log.i("MainActivity", "Bố cục MainActivity đã hoàn tất");
```

Bạn sẽ thấy câu lệnh "Bố cục MainActivity đã hoàn tất" trong ngăn **Logcat** nếu menu Mức nhật ký được đặt thành tùy chọn nào sau đây? (Gợi ý: nhiều câu trả lời là được.)

- Dài dòng
- Gỡ lỗi
- Thông tin
- Cảnh báo
- Lỗi

- Khẳng định

Gửi ứng dụng của bạn để chấm điểm

Kiểm tra để đảm bảo ứng dụng có những điều sau:

- Một ctivity hiển thị "Hello World" trên màn hình.
- Ghi nhật ký các câu lệnh trong onCreate() trong hoạt động chính.
- Cập nhật ký trong **ngăn Logcat** chỉ hiển thị các câu lệnh gỡ lỗi hoặc ghi nhật ký lỗi.