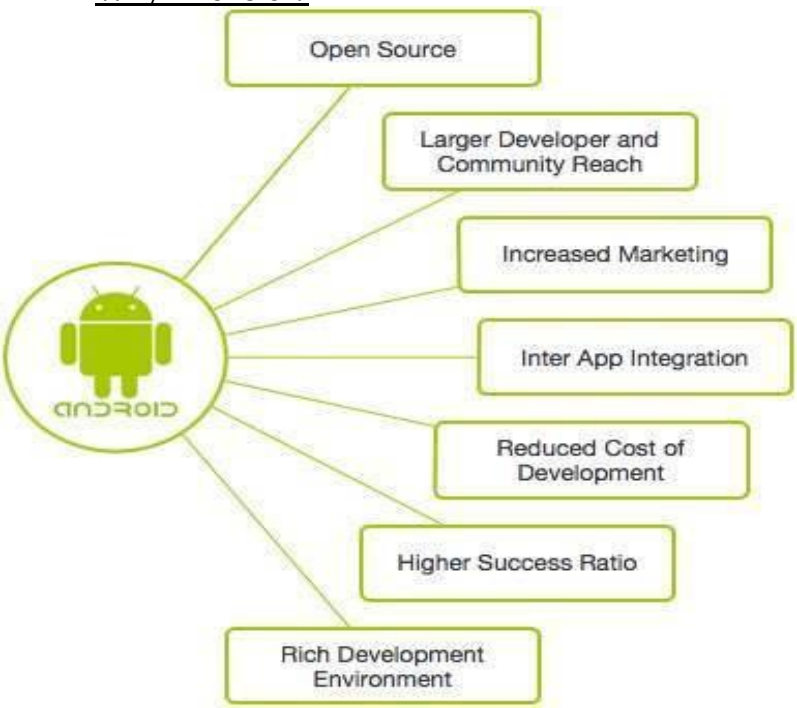
Stream: B.SC (computer science)

Subject : Fundamentals of Mobile Programming

Unit - 1 | Introduction to Android

- Android operating system is the largest installed base among various mobile platforms across the globe.
- Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world.
- It conquered around 71% of the global market share by the end of 2021, and this trend is growing bigger ever0
- y other day.
- The company named **Open Handset Alliance** developed Android for the first time that is based on the modified version of the Linux kernel and other open-source software.
- Google sponsored the project at initial stages and in the year 2005, it acquired the whole company.
- **In September 2008, the first Android-powered device was launched in the market.**
- Android dominates the mobile OS industry because of the long list of features it provides.
- It's user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android-compatible smartphones.
- As a result, the market observes a sharp increase in the demand for developing Android mobile applications, and with that companies need smart developers with the right skill set.

Why Android ?



History of Mobile Software Development :

- Initially, **Andy Rubin** founded Android Incorporation in Palo Alto, California, United States in October, 2003.
- In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.
- The key employees of Android Incorporation are **Andy Rubin**, **Rich Miner**, **Chris White** and **Nick Sears**.

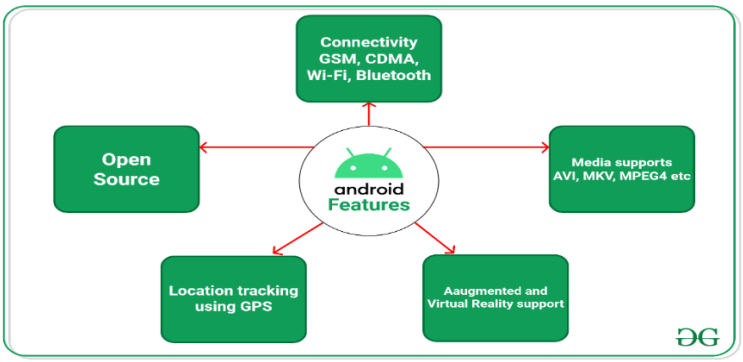
- Originally intended for camera but shifted to smart phones later because of low market for camera only.
- Android is the nick name of Andy Rubin given by coworkers because of his love to robots.
- In 2007, Google announces the development of android OS.
- In 2008, HTC launched the first android mobile.

What is Android API?

Application program interface (API) is a code for a programmer that they use in their applications.
API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.
The Android platform provides a framework API that applications can use to interact with the underlying Android system.
The framework API consists of: A core set of packages and classes.

Features of Android :

- Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.



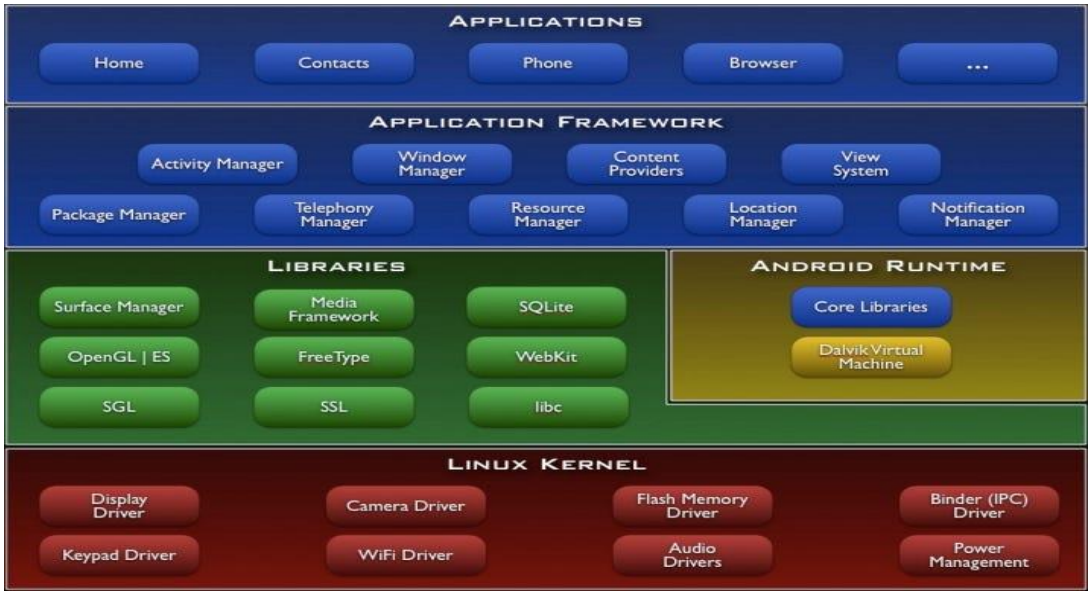
- Android Open Source Project so we can customize the OS based on our requirements.
- Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
- Using wifi technology we can pair with other devices while playing games or using other applications.
- It contains multiple APIs to support location-tracking services such as GPS.
- We can manage all data storage-related activities by using the file manager.
- It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.
- It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
- It supports multimedia hardware control to perform playback or recording using a camera and microphone.
- Android has an integrated open-source WebKit layout-based web browser to support User Interfaces like HTML5, and CSS3.
- Android supports multi-tasking means we can run multiple applications at a time and can switch between them.
- It provides support for virtual reality or 2D/3D Graphics.

The Open Handset Alliance :

- The open handset alliance (oha) is a business alliance.
- That was created for the purpose of developing open Mobile device standards. The oha has approximately 80 Member companies, including htc, dell, intel, motorola, Qualcomm and google.
- The oha's main product is the Android platform - the world's most popular smartphone platform.

Android Architecture :

- Android architecture contains a different number of components to support any Android device's needs.
- **Components of Android Architecture :**
  The main components of Android architecture are the following:-
  1. Applications
  2. Application Framework
  3. Android Runtime
  4. Platform Libraries
  5. Linux Kernel



1) Linux Kernel :

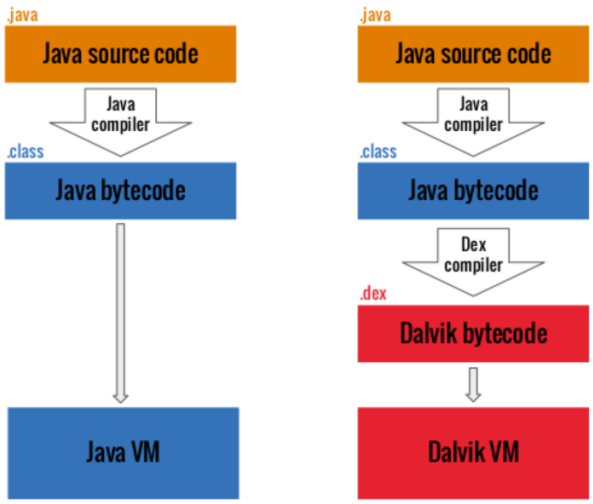
- Linux Kernel is heart of the android architecture.
- It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.
- The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture.
- It is responsible for management of memory, power, devices etc.

2) Platform libraries :

- The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.
  - **I.** **Media** library provides support to play and record an audio and video formats.
  - **II.** **Surface manager** responsible for managing access to the display subsystem.
  - **III.** **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) are used for 2D and 3D computer graphics.
  - **IV.** **SQLite** provides database support and FreeType provides font support.
  - **V.** **Web-Kit** This open source web browser engine provides all the functionality to display web content and to simplify page loading.
  - **VI.** **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser.

3) Application runtime :

- Android Runtime environment is one of the most important part of Android.
- It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries. Like Java Virtual Machine (JVM), Dalvik Virtual Machine (DVM) is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently.
- It depends on the layer Linux kernel for threading and low-level memory management.



- The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

4) Application framework :

- Application Framework provides several important classes which are used to create an Android application.
- It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources.
- Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.
- It includes different types of services activity manager, notification manager, view system, package manager etc.

5) Applications :

- Applications is the top layer of android architecture.
- The pre-installed applications like home, contacts, camera, gallery etc and third party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only.
- It runs within the Android run time with the help of the classes and services provided by the application framework.

Android SDK :

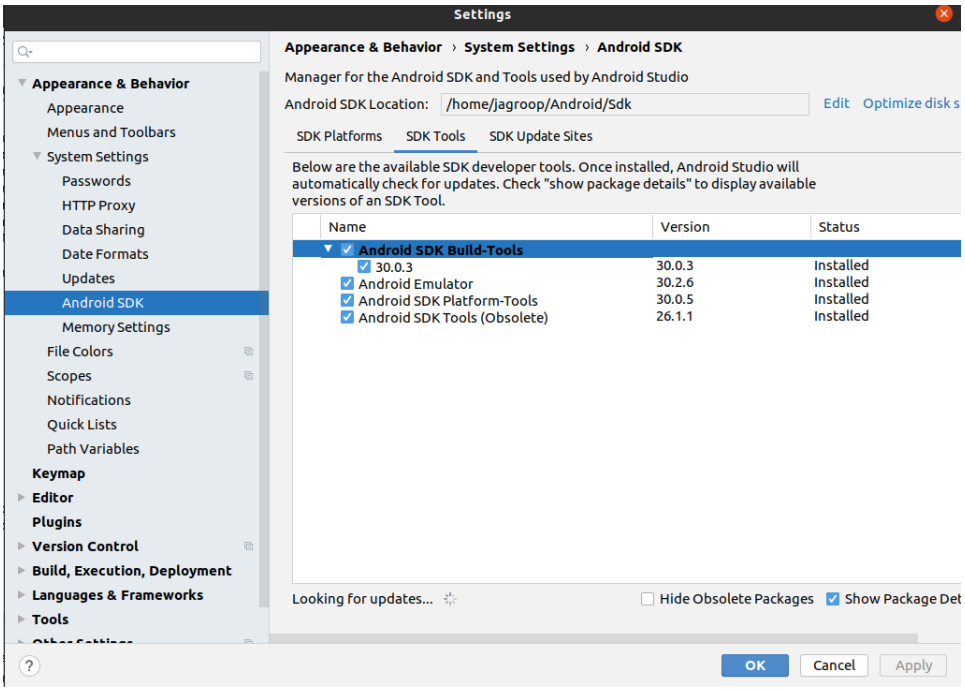
- **Android SDK** stands for **Android Software Development Kit** which is developed by Google for Android Platform. With the help of Android SDK, we can create android Apps easily.
- Android SDK is a collection of libraries and Software Development tools that are essential for Developing Android Applications.
- Whenever Google releases a new version or update of Android Software, a corresponding SDK also releases with it.
- In the updated or new version of SDK, some more features are included which are not present in the previous version.

- Android SDK consists of some tools which are very essential for the development of Android Application.
- These tools provide a smooth flow of the development process from developing and debugging. Android SDK is compatible with all operating systems such as Windows, Linux, macOS, etc.

➢ **Components of Android SDK :**

1. Android SDK Tools :

- Android SDK tool is an important component of Android SDK. It consists of a complete set of development and debugging tools. Below are the SDK developer tools:
  - Android SDK Build tool.
  - Android Emulator.
  - Android SDK Platform-tools.
  - Android SDK Tools.
- These are shown below :



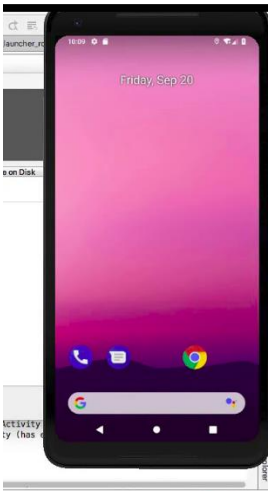
2. Android SDK Build-Tools :

- Android SDK build tools are used for building actual binaries of Android App.
- The main functions of Android SDK Build tools are built, debug, run and test Android applications.
- The latest version of the Android SDK Build tool is 30.0.3. While downloading or updating Android in our System, one must ensure that its latest version is download in SDK Components.

3. Android Emulator :

- An Android Emulator is a device that simulates an Android device on your system. Suppose we want to run our android application that we code.

- One option is that we will run this on our Android Mobile by Enabling USB Debugging on our mobile.
- Another option is using Android Emulator. In Android Emulator the virtual android device is shown on our system on which we run the Android application that we code.
- Thus, it simply means that without needing any physical device Android SDK component "Android Emulator" provides a virtual device on the System where we run our Application.
- The emulator's come with the configuration for Various android phones, tablets, Wear OS, and Android TV devices.
- In Android Virtual Emulator all functions that are feasible on real Android mobile is works on virtual Device like:
  - phone calls, text messages.
  - stimulate different network speeds.
  - specify the location of a device
  - access on google play store and lot's more.
- But there is one disadvantage of this emulator is that. It is very slow when System's PC has less RAM. It works fine when a maximum GB of RAM is present on our device.



4. Android SDK Platform-tools :

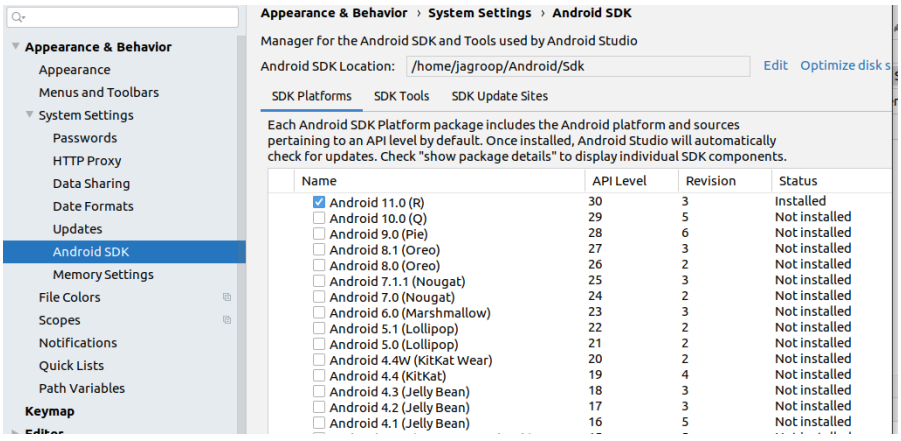
- Android SDK Platform-tools is helpful when we are working on Project and they will show the error messages at the same time. It is specifically used for testing. It includes:
  - Android Debug Bridge (ADB), is a command-line tool that helps to communicate with the device. It allows us to perform an action such as Installing App and Debugging App etc.
  - Fastboot allows you to flash a device with a new system image.
  - Systrace tools help to collect and inspect timing information. It is very crucial for App Debugging.

5. Android SDK Tools :

- Android SDK tool is a component of SDK tool.
- It consists of a set of tools which and other Utilities which are crucial for the development of Android Application.
- It contains the complete set of Debugging and Development tools for android.

6. SDK Platforms :

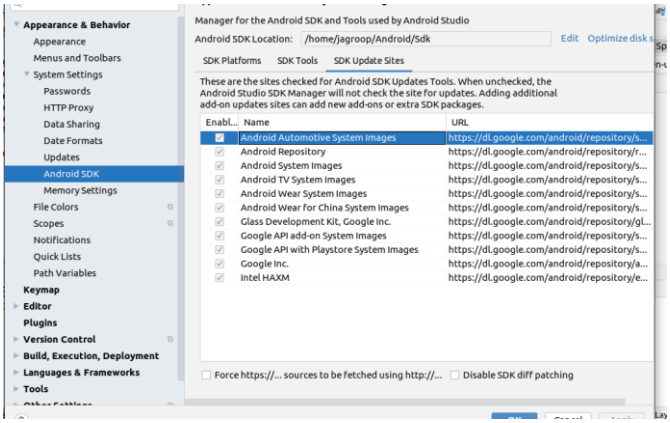
- For Each Android Software, one SDK platform is available as shown below:
- Like in this Android 11.0(R) is installed.
- These are numbered according to the android version. The new version of the



SDK platform has more features and more compatible but the old version is less compatible with fewer features.

- Like in Android 11.0(R) have more compatible and have more feature but the below versions like Android 10.0(Q), Android4.4(KitKat) have less feature and is less compatible.

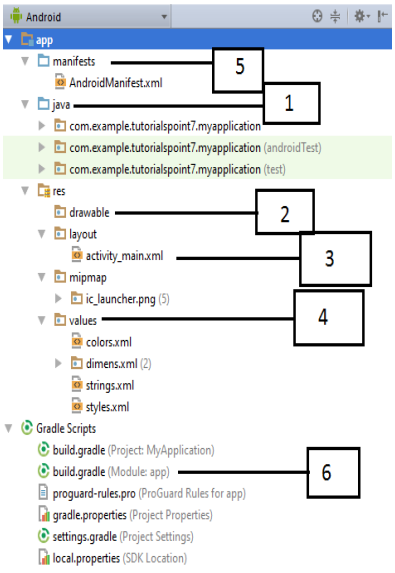
7. SDK Update Sites :



- In SDK Update Sites, some sites are embedded in it which will check for Android SDK Updates Tools.
- In this, one must ensure we don't unclick the button below because these are checked by default which will check for updates if we will unclick it then it doesn't check updates for those.

Building a sample Android application :

Sr.No.	Folder, File & Description
1	Java This contains the .java source files for your project. By default, it includes an *MainActivity.java* source file having an activity class that runs when your app is launched using the app icon.
2	res/drawable-hdpi This is a directory for drawable objects that are designed for high-density screens.
3	res/layout This is a directory for files that define your app's user interface.
4	res/values This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.
5	AndroidManifest.xml This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.
6	Build.gradle This is an auto generated file which contains compileSdkVersion, buildToolsVersion, applicationId, minSdkVersion, targetSdkVersion, versionCode and versionName

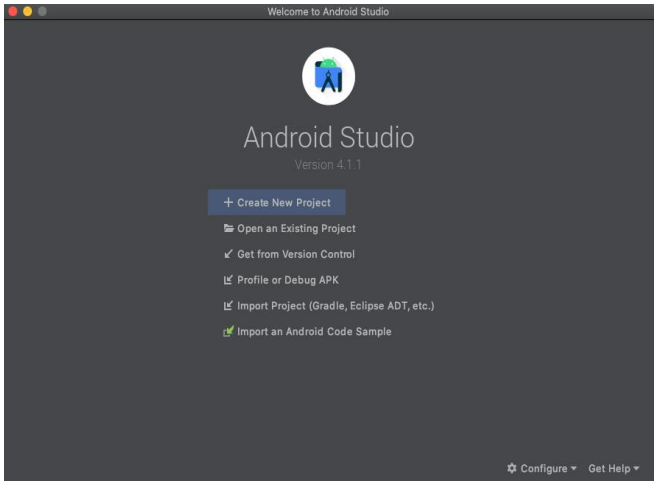


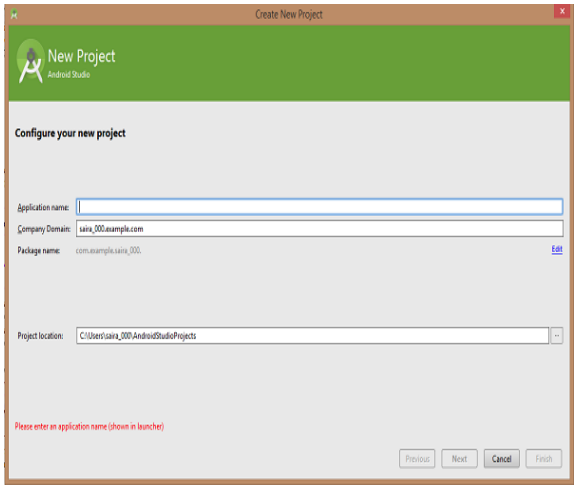
Step 1: Create a New Project

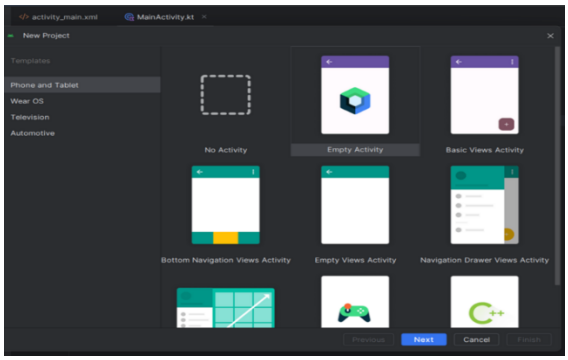
To create your new Android project, follow these steps:

Install the latest version of [Android Studio](#).

In the Welcome to Android Studio **window, click** Create New Project**.**







If you have a project already opened, select **File > New > New Project**.

In the **Select a Project Template** window, select **Empty Activity** and click **Next**.

In the **Configure your project** window, complete the following:

- Enter "My First App" in the **Name** field.
- Enter "com.example.myfirstapp" in the **Package name** field.
- If you'd like to place the project in a different folder, change its **Save** location.
- Select either **Java** or **Kotlin** from the **Language** drop-down menu.
- Select the lowest version of Android you want your app to support in the **Minimum SDK** field.
- If your app will require legacy library support, mark the **Use legacy android.support libraries** checkbox.
- Leave the other options as they are.
- Click **Finish**.

Step 2: Design First Screen Layout

In the activity_main.xml file let us design the layout for the first screen.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textSize="32sp"
        app:layout_constraintBottom_toTopOf="@+id/btnSecondActivity"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.469"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3"
        app:layout_constraintVertical_bias="0.206" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

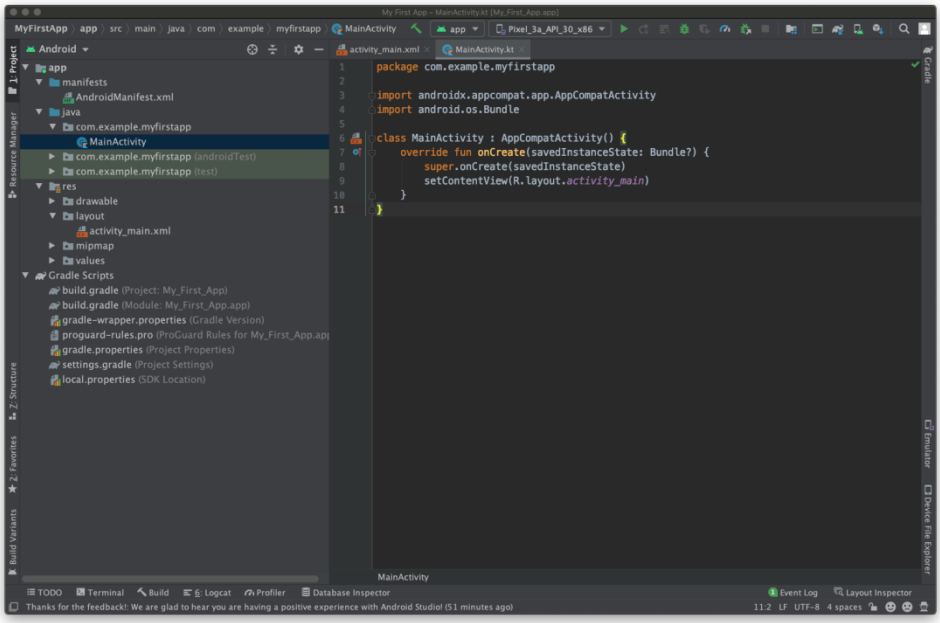
Note :
This **XML** code defines two **TextView** **and a** **Button** for the **first screen layout in an Android app** with the following specifications:

- **XML code** for an Android layout is provided.
- Includes a large **TextView displaying "First Screen"** with specific text attributes.r.
- Contains another **TextView displaying "Hello World,"** positioned below the first TextView and above the button.
- **Layout constraints** ensure proper alignment of elements relative to the parent and each other.

Step 3: Implement First Activity :

In **MainActivity.kt,** we implement the **functionality** for the first screen. In the first screen we are going to create the functionality that is **when the user clicks the button he is taken to the second screen**. Let us see the code for it in the file named **MainActivity.kt**.

After some processing time, the Android Studio main window appears.



Step : 4 Running the Application :

Let's try to run our Hello World! application we just created.

➔ I assume you had created your AVD while doing environment set-up.

➔ To run the app from Android studio, open one of your project's activity files and click Run icon from the tool bar.

➔ Android studio installs the app on your AVD and starts it and if everything is fine with your set-up and application, it will display following Emulator window.