UNIT II: Android Studio And User Interface Design

1.1   Introduction to Mobile Programming and Smartphones future

1.2 Overview of the Operating Systems used on different mobile devices

1.3 Android Operating System Features and Versions

1.4 Overview of the development languages available on different mobile devices

1.5 Explore mobile device features not available on PCs such as accelerometer and GPS etc.

1.6 Android Architecture

1.7 Installing Android Studio and Android Virtual Device

1.8 Creating First Android Project.

1.9 Android Project Structure.

1.1 Introduction to Mobile Programming and Smartphones future

Mobile Programming is the process of making software for smartphones and digital assistants, most commonly for Android and iOS.

The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser.

The programming and markup languages used for this kind of software development include Java, Kotlin, Swift, C# and HTML5, CSS and JavaScript etc.

Mobile Programming or Mobile App Development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, or mobile phones or tablets

Mobile app development is rapidly growing.

From retail, telecommunications and banking to insurance, healthcare and government,organizations across industries must meet user expectations for fast, convenient ways to conduct transactions and access information.

Today, mobile devices are the most popular way for people to connect to the internet. To stay relevant, responsive and successful, organizations need mobile apps.

So there is **the huge demand of mobile programmers and professional in future**.

There are different types of Mobile applications:
1. Native Applications
2. Hybrid Applications
3. Progressive Web Applications (PWAs)

1. Native Applications

These are applications developed to be used on a particular platform or operating system such as Android, iOS etc.

Native apps are usually written in languages that the platform accepts.

They are also built using the specific Integrated Development Environment (IDE) for the given operating systems, such as **Android Studio** for Android Apps and **XCode for iOS Apps**.

Examples of some popular Native Applications are Instagram for Android,VLC media player for Android, WordPress for iOS, and 2048 game for iOS

Advantages

•They are very fast and responsive because they are built for that specific platform.

They have the best performance.

They are more interactive, natural and run much smoother in terms of userinput and output.

Disadvantages: –

They are considerably **more expensive** to develop compared to cross-platform and web applications.
They require **more time** to develop as one application has to be written in different languages for different platforms.
They have a **higher cost** of maintenance and pushing out updates, due to multiple source code bases.

2. Hybrid Applications:

These are applications developed to be used across multiple platforms and can be deployed on both iOS and Android platforms.

•   Hybrid mobile applications are built in a similar manner as websites. Both use a combination of technologies like HTML, CSS, and JavaScript

•   Examples of some popular Hybrid Applications are MarketWatch, Untappd, FanReact, and TripCase

Advantages:

Adaptable to multiple platforms, as the same code can be re-used for Android, iOS, and Windows.

Unified and **less expensive** development, as the app only has to be developed once using one code base.
–   **Faster development time** when compared to native apps as only one

development process is involved.
- **Disadvantages:**
- **Slower app performance** when compared to Native Apps, because the hybrid framework acts as a bridge to communicate with the phone's native features.
- Apps with **heavy animations and sound effects** aren't as unbroken as theirnative counterparts

3. Progressive Web Applications (PWAs):

A Progressive Web App (PWA) is a web app that uses modern web capabilities to deliver an app-like experience to users without requiring them to install an app from the AppStore/PlayStore.

- They are usually accessible by a web URL which can always be pinned or saved on your phone's home screen.
- PWAs are usually built using HTML, CSS, and JavaScript also.
- Examples of some popular Progressive Web Applications are AliExpress's PWA, Financial Times PWA, NASA's PWA

Advantages

1. Easy to build.
2. Easy to maintain.
3. Works offline and performs well on low-quality networks.
4. Build one app for all platforms
5. iOS, Android, etc. as long as it can run a browser.

Disadvantages

Needs a browser to run. Users have to take another step to type in theURL of the app which amounts to a poor user experience.

Web apps are less interactive and natural than native apps

1.2 Overview of the Operating Systems used on different mobile devices

Android is a software package and Linux based operating system for mobile devices such as tablet computers and smartphones.

It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used.

Open Handset Alliance (OHA) is a consortium of 84 companies such as google, samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc.

It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Platform.

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 launched in the market.

The goal of android project is to create a successful real-world product that improves the mobile experience for end users.

Android is the best-selling **Operating System** 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 captured around **75%** of the global market share by the end of 2020, and this trend is growing bigger every other day.

Features

It's user-friendly.
Has huge community support.
Provides a greater extent of customization.
A large number of companies build Android-compatible smartphones.

Android becomes an absolute set of software for all devices like tablets, phone, wear os, smart TVs, etc.

1.3 Android Operating System Features and Versions

Android is an open source and Linux-based **Operating System** for mobile devices such as smartphones and tablet computers.

Android was developed by the *Open Handset Alliance*, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

Features of Android

Beautiful UI
Android OS basic screen provides a beautiful and intuitive user interface.

Connectivity

GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

Storage

SQLite, a lightweight relational database, is used for data storage purposes.

Media support

H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.

Messaging

SMS and MMS

Web browser

Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.

Multi-touch

Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.

Multi-tasking

User can jump from one task to another and same time various application can run simultaneously.

Multi-Language

Supports single direction and bi-directional text.

Android versions, name

Cupcake	**April 27, 2009**
Donut	**September 15, 2009**
Eclair	**October 26, 2009**
Froyo	**May 20, 2010**
Gingerbread	**December 6, 2010**
Honeycomb	**February 22, 2011**
Ice Cream Sandwich	**October 18, 2011**
Jelly Bean	**July 9, 2012**
KitKat	**October 31, 2013**
Lollipop	**November 12, 2014**
Marshmallow	**October 5, 2015**
Nougat	**August 22, 2016**
Nougat	**October 4, 2016**
Oreo	**August 21, 2017**
Oreo	**December 5, 2017**
Pie	**August 6, 2018**
Android 10(Quince Tart)	**September 3, 2019**
Android 11(Red Velvet Cake)	**September 8, 2020**
Android 12(Snow Cone)	**October 4, 2021**
Android 13(Tiramisu)	**August 15, 2022**
Android 14(Upside Down Cake)	**October 4, 2023**
Android 15(Vanilla Ice Cream)	**-**

1.4 OVERVIEW OF THE OPERATING SYSTEMS USED ON DIFFERENT MOBILE DEVICES:

1. A mobile operating system is an operating system that helps to run other application software on mobile devices.
2. It is the same kind of software as the famous computer operating systems like Linux and Windows, but now they are light and simple to some extent.
3. The operating systems found on smartphones include Symbian OS, iPhone OS, RIM's BlackBerry, Windows Mobile, Palm WebOS, Android, and Maemo. Android, WebOS, and Maemo are all derived from Linux.
4. The iPhone OS originated from BSD and NeXTSTEP, which are related to UNIX.
5. It combines the beauty of computer and hand use devices.
6. It typically contains a cellular built-in modem and SIM tray for telephony and internet connections.
7. Popular platforms of the Mobile OS are

1. Android OS: The Android operating system is the most popular operating system today.

It is a mobile OS based on the **Linux Kernel** and **open-source software**. The android operating system was developed by **Google**. The first Android device was launched in **2008**.

2. Bada (Samsung Electronics): Bada is a Samsung mobile operating system that was launched in 2010.

The Samsung wave was the first mobile to use the bada operating system.

The bada operating system offers many mobile features, such as 3-D graphics, application installation, and multipoint-touch.

3. BlackBerry OS: The BlackBerry operating system is a mobile operating system developed by **Research in Motion** (RIM).

This operating system was designed specifically for BlackBerry handheld devices.

This operating system is beneficial for the corporate users.

4. IPhone OS / iOS: The iOS was developed by the Apple inc for the use on its device.

The iOS operating system is the most popular operating system today. It is a very secure operating system. The iOS operating system is not available for any other mobiles.

5. Symbian OS: Symbian operating system is a mobile operating system that provides a high-level of integration with communication.

The Symbian operating system is based on the java language.

It combines middleware of wireless communications and personal information management (PIM) functionality.

The Symbian operating system was developed by **Symbian Ltd** in **1998** for the use of mobile phones.

Nokia was the first company to release Symbian OS on its mobile phone at that time.

6. Windows Mobile OS: The window mobile OS is a mobile operating system that was developed by **Microsoft**. It was designed for the pocket PCs and smart mobiles.

7. Harmony OS: The harmony operating system is the latest mobile operating system that was developed by Huawei for the use of its devices. It is designed primarily for IoT devices.

1.5 Explore mobile device features not available on PCs such as accelerometer and GPS etc.

The features of mobile phones are the set of capabilities, services and applications that they offer to their users.

The common components found on all phones are:

A number of metal–oxide–semiconductor (MOS) integrated circuit (IC) chips.

A battery, providing the power source for the phone functions.

An input mechanism to allow the user to interact with the phone. The most common input mechanism is a keypad, but touch screens are also found in smartphones.

Basic mobile phone services to allow users to make calls and send text messages.

All phones use a SIM card for communication

Phone devices are uniquely identified by an International Mobile

Equipment Identity (IMEI) number.

All mobile phones are designed to work on cellular networks and contain a standard set of services.

Roaming which permits the same phone to be used in multiple countries, providing that the operators of both countries have a roaming agreement.

Send and receive data and faxes (if a computer is attached),

WAP services, and provide full Internet access using technologies such as GPRS (General Packet Radio Services).

Proximity Sensor

Detects when an object is near to the phone. Most commonly used to sense when a phone is held up to the users ear to turn off the display. This saves both battery life and prevents accidental screen touches.

Digital compass

The digital compass that's usually based on a sensor called the magnetometer and provides mobile phones with a simple orientation in relation to the Earth's magnetic field.

As a result, your phone always knows which way is North (North on Maps when in portrait and landscape orientation) so it can auto rotate your digital maps depending on your physical orientation.

Biometrics

Biometric related sensors provide levels of enhanced security by capturing and validating human related metrics. Including Finger

Print recognition, IRIS (eye) scanning and full facial recognition.

Biometric sensors provide a more secure but more convenient way to unlock phones and pay for purchases.

Additionally, biometric sensors can be used to collect a user's heart rate and SpO2 (the estimate of arterial oxygen saturation) for use within a vendors 'health' application.

Augmented & Virtual Reality

The highly accurate sensors combined with the powerful CPU &

GPU's of modern smart phones, allow very realist and responsive Virtual Reality applications to be created. When the sensors are combined with a smartphones camera they facilitate Augmented Reality applications.

GPS

GPS stands for Global Positioning System. It's a technology developed by the U.S. Navy and currently owned by the U.S. government. It's free for everyone to use

GPS is a radio navigation system. It uses radio waves between satellites and a receiver inside your phone to provide location and time information to any software that needs to use it.

1.6 Android Architecture

Android architecture **is a software stack of components to support mobile device needs. Android software stack contains a Linux Kernel, collection of c/c++ libraries which are exposed through an application framework services, runtime, and application.**

Following are main components of android architecture those are

1. Applications
2. Android Framework
3. Android Runtime
4. Platform Libraries
5. Linux Kernel

In these components, the Linux Kernel **is the main component in android to provide its operating system functions to mobile and** Dalvik Virtual Machine **(**DVM**) which is responsible for running a mobile application.**
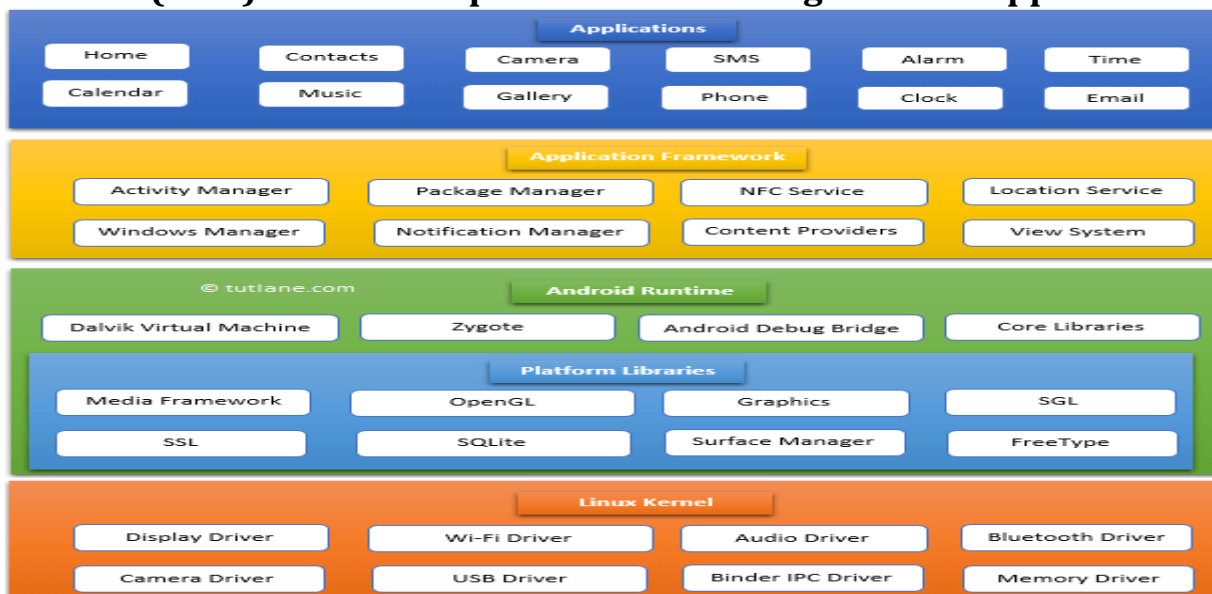
Applications

The top layer of the android architecture is **Applications**. The native and third-party applications like contacts, email, music, gallery, clock, games, etc. whatever we will build those will be installed on this layer only.

Application Framework

The **Application Framework** provides the classes used to create Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources. It basically provides the services through which we can create a particular class and make that class helpful for the Application creation.

Android Runtime

Android Runtime environment is an important part of Android rather than an internal part and it contains components like **core libraries** and the **Dalvik virtual machine**. The Android run time is the engine that powers our applications along with the libraries and it forms the basis for the application framework.

Dalvik Virtual Machine **(**DVM**) is a register-based virtual machine like Java Virtual Machine (JVM). It is specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It relies on the Linux kernel for threading and low-level memory management.**

Platform Libraries

The **Platform Libraries** includes various C/C++ core libraries and Java-based libraries such as SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL, etc. to provide support for Android development.

Linux Kernel

Linux Kernel is a bottom layer and heart of the android architecture. It manages all the drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc.

1.7 Installing Android Studio

Step 1
To download the Android Studio, visit the official Android Studio website in your web browser.

Step 2
Click on the "Download Android Studio" option.
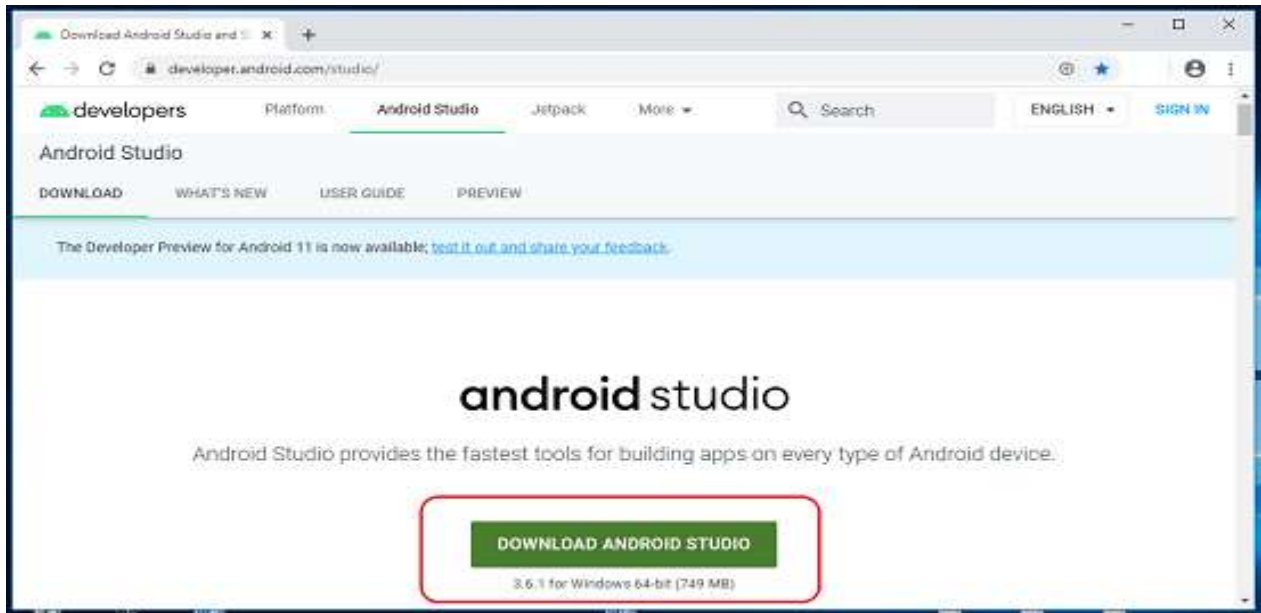
Step 3
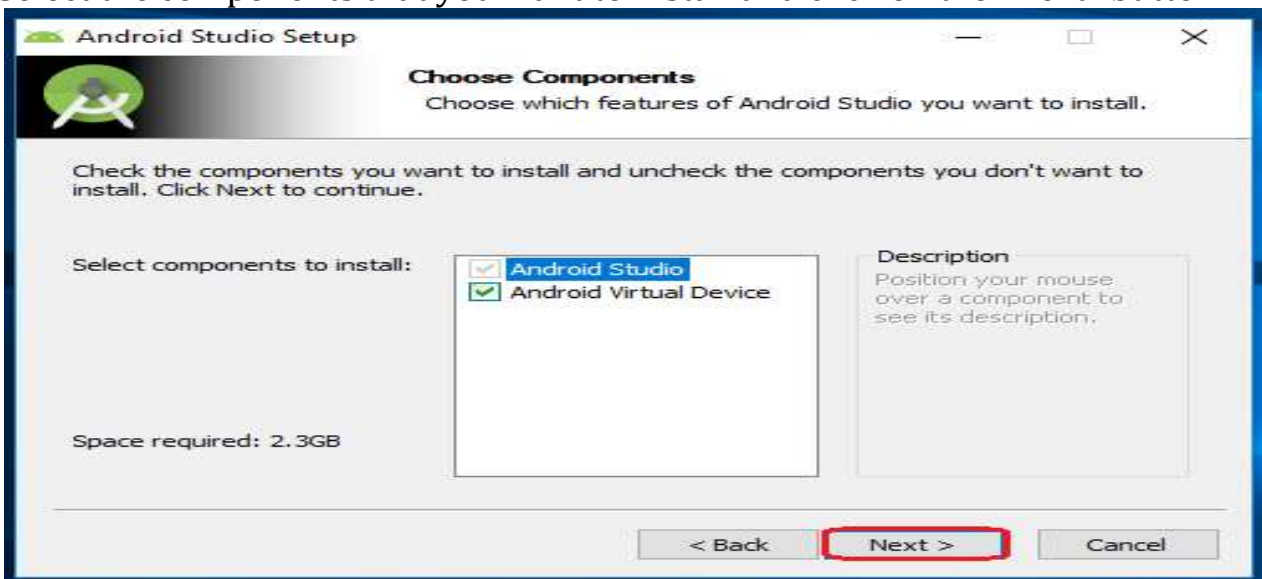Double click on the downloaded "Android Studio-ide.exe" file.
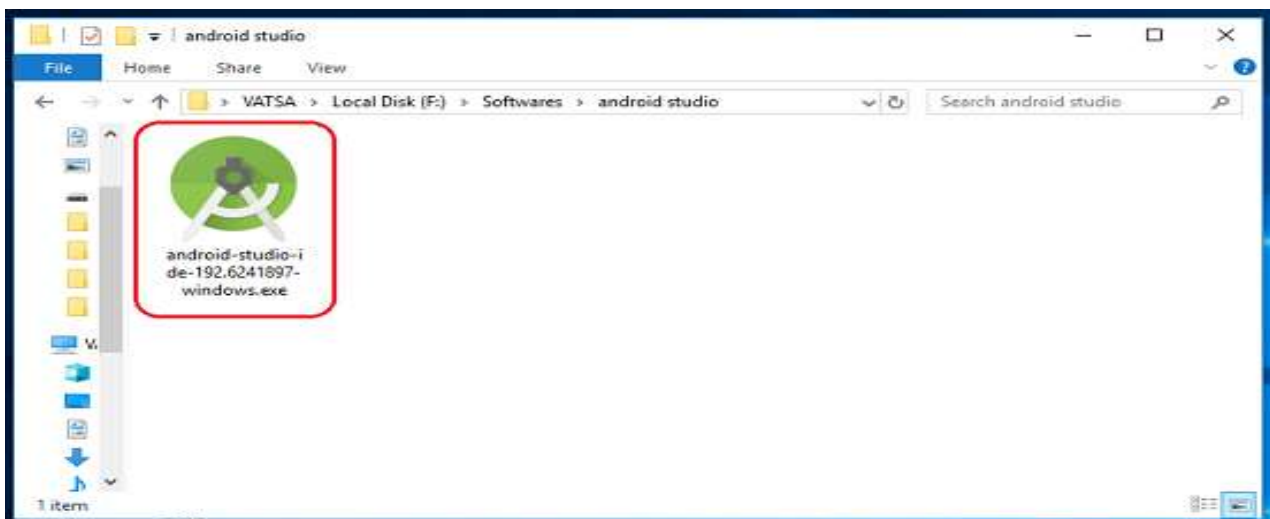
Step 4
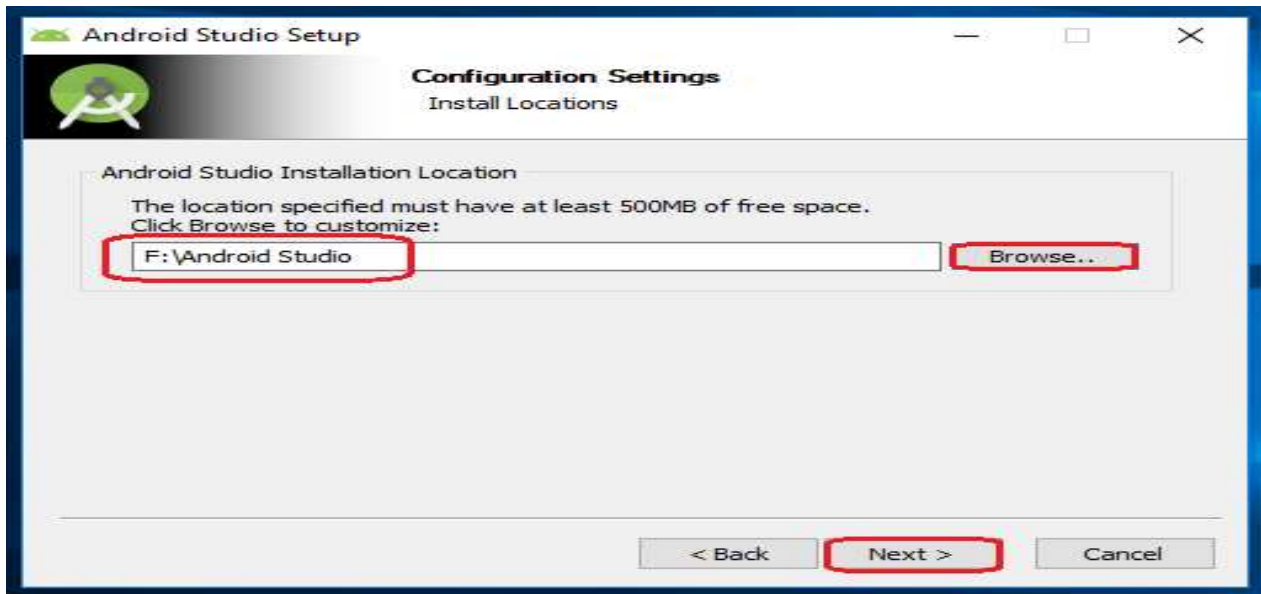"Android Studio Setup" will appear on the screen and click "Next" to proceed.

Step 5

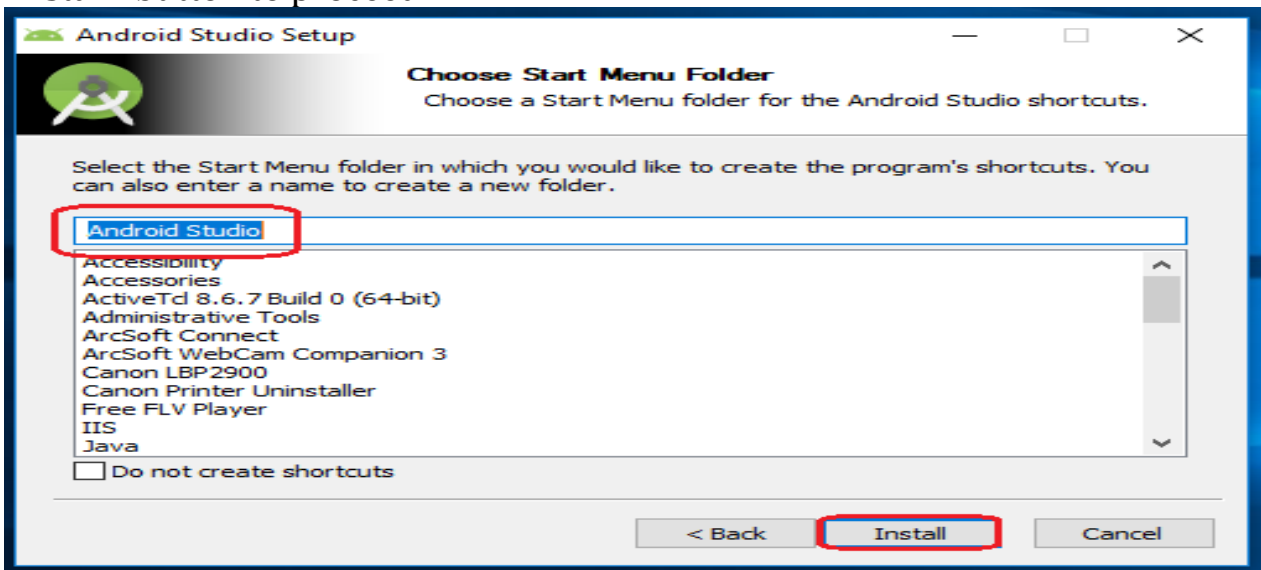Select the components that you want to install and click on the "Next" button.

Step 6
Now, browse the location where you want to install the Android Studio and click "Next" to proceed.

Step 7

Choose a start menu folder for the "Android Studio" shortcut and click the "Install" button to proceed.

Step 8

After the successful completion of the installation, click on the "Next" button.
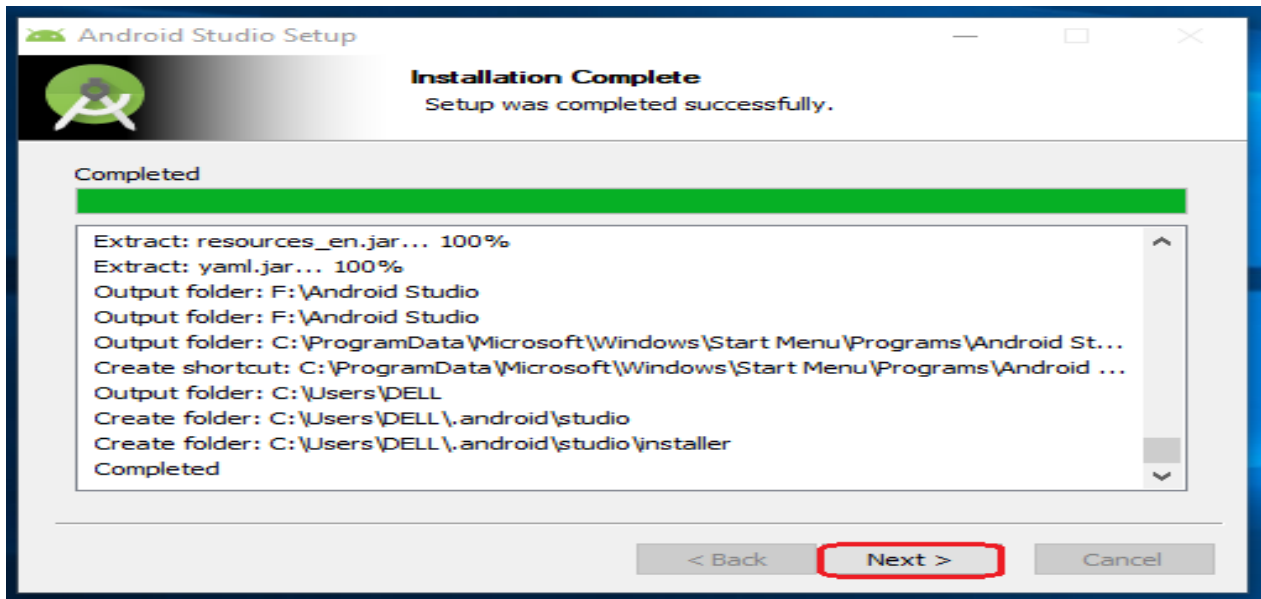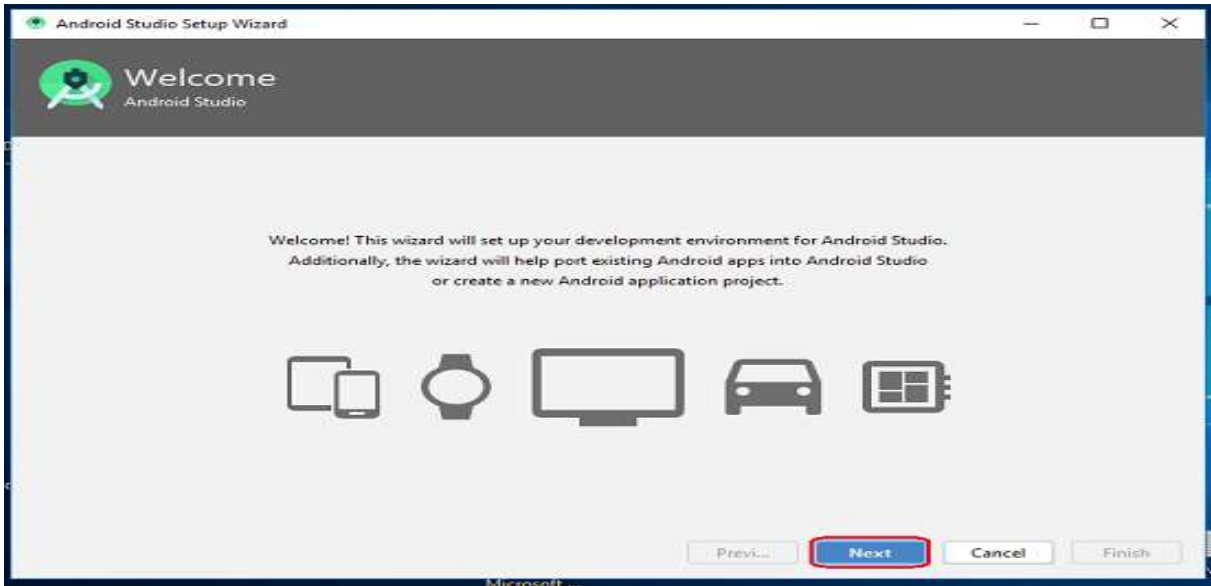
Step 9

Click on the "Finish" button to proceed.
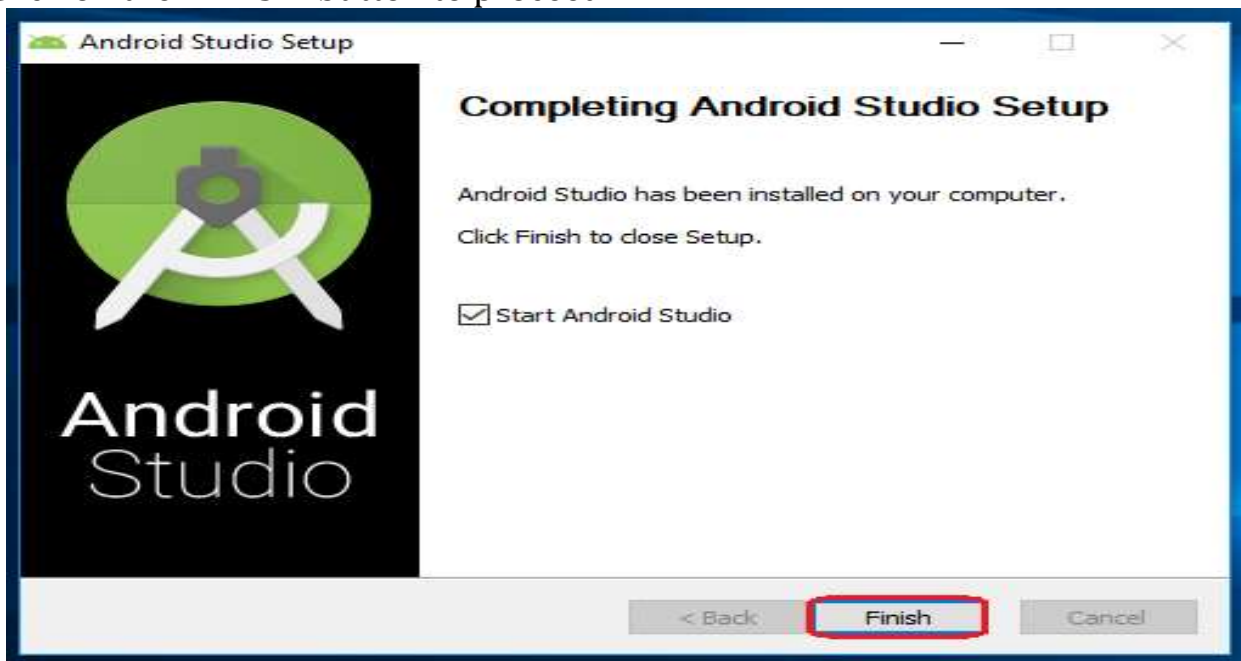
Now, your Android studio welcome screen will appear on the screen.

Step 10
"Android Studio Setup Wizard" will appear on the screen with the welcome
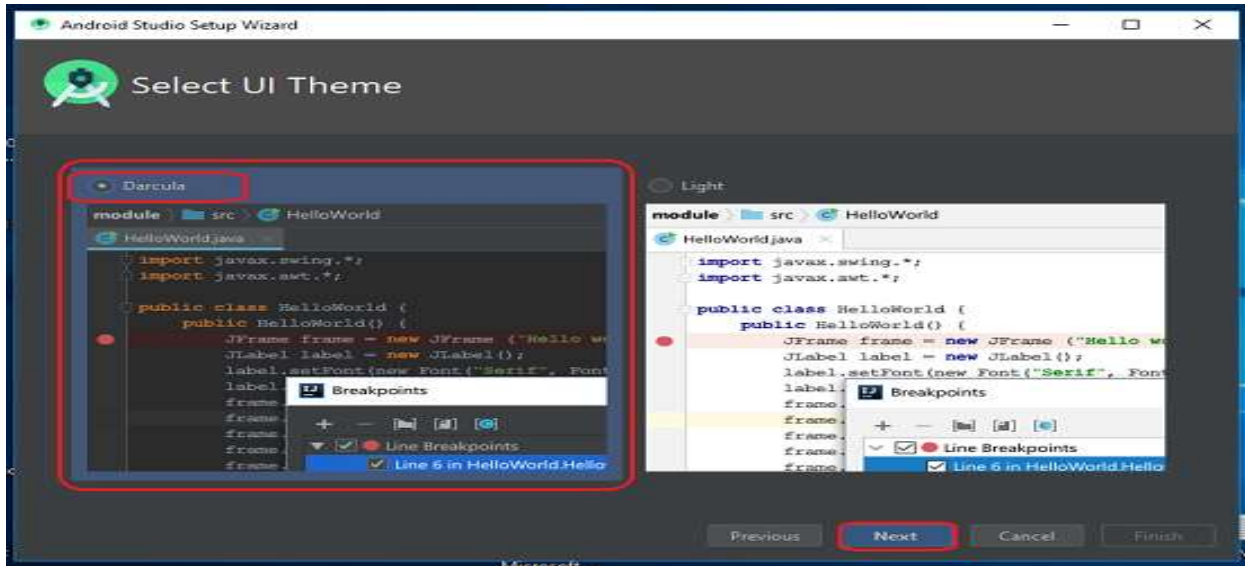wizard. Click on the "Next" button.

Step 11
Select (check) the "Standard" option if you are a beginner and do not have any
idea about Android Studio. It will install the most common settings and options
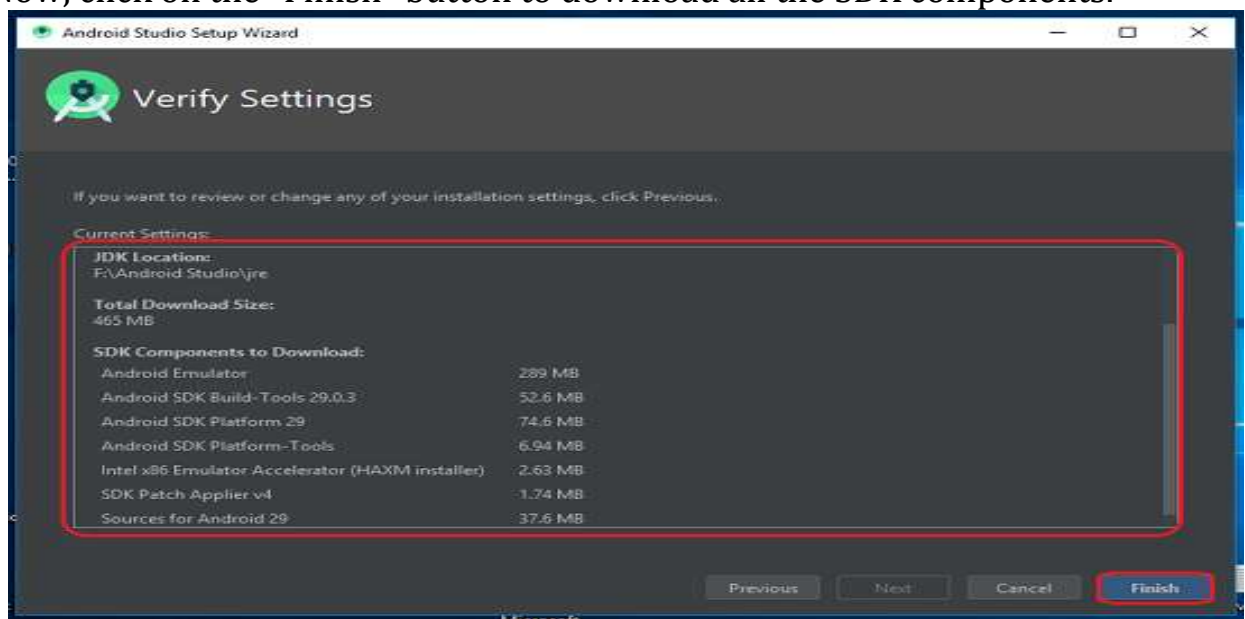for you. Click "Next" to proceed.
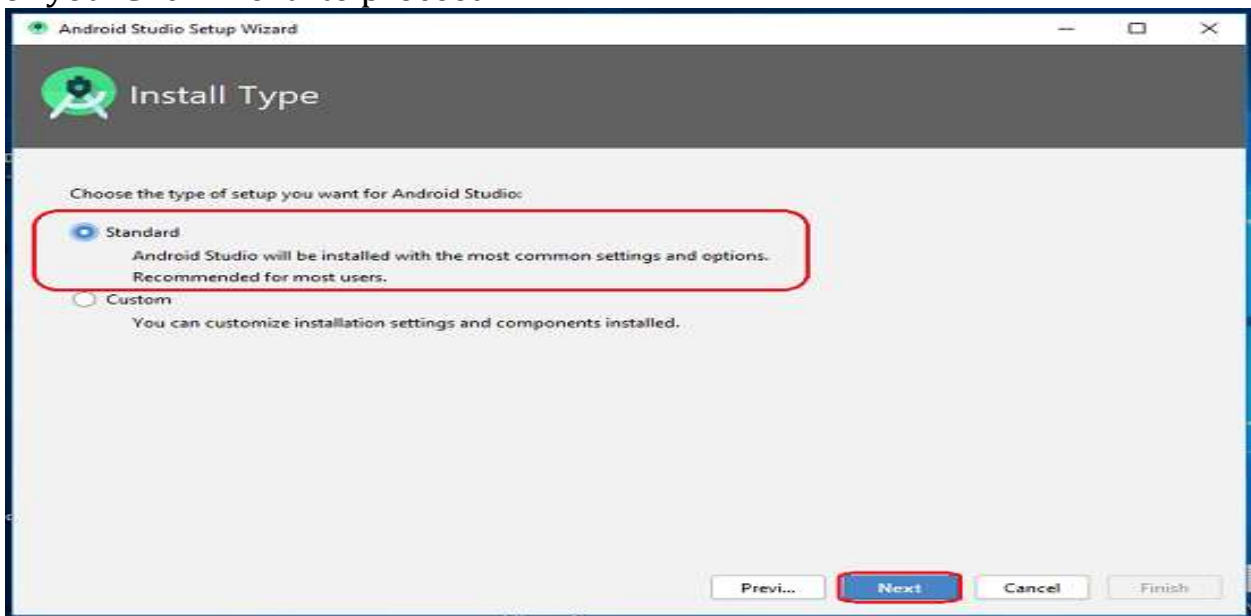
Step 12

Now, select the user interface theme as you want. (I prefer Dark theme (Dracula) that is most liked by the coders). Then, click on the "Next" button.
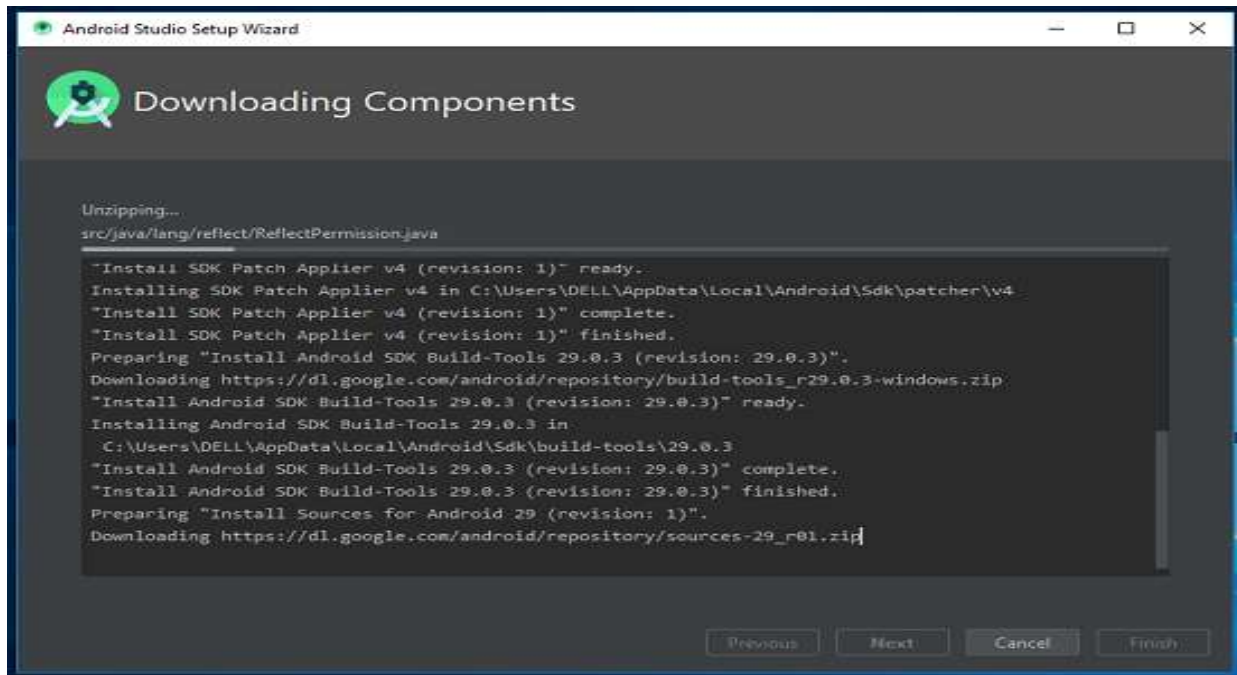

Step 13

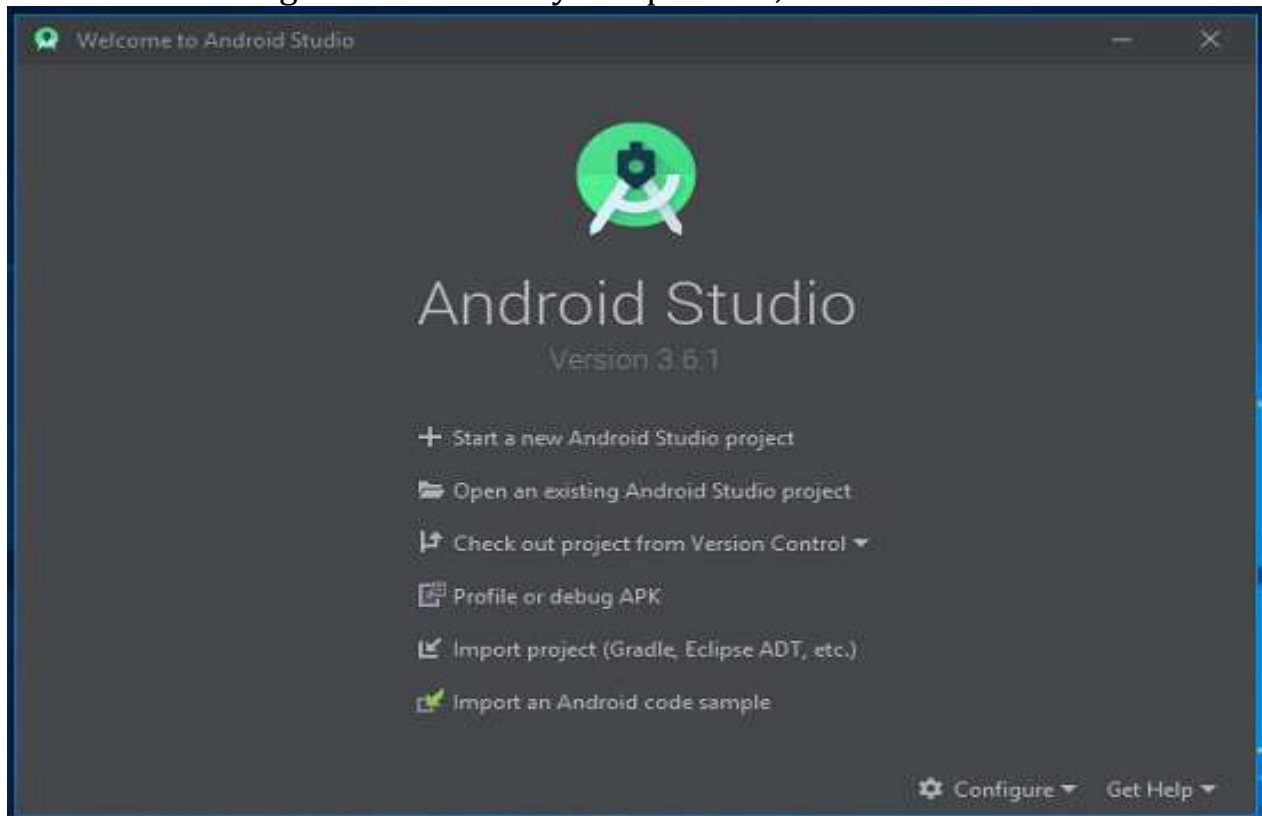Now, click on the "Finish" button to download all the SDK components.

And, the downloading and installation process of components gets started.

Step 14
After downloading all the necessary components, click on the "Finish" button.
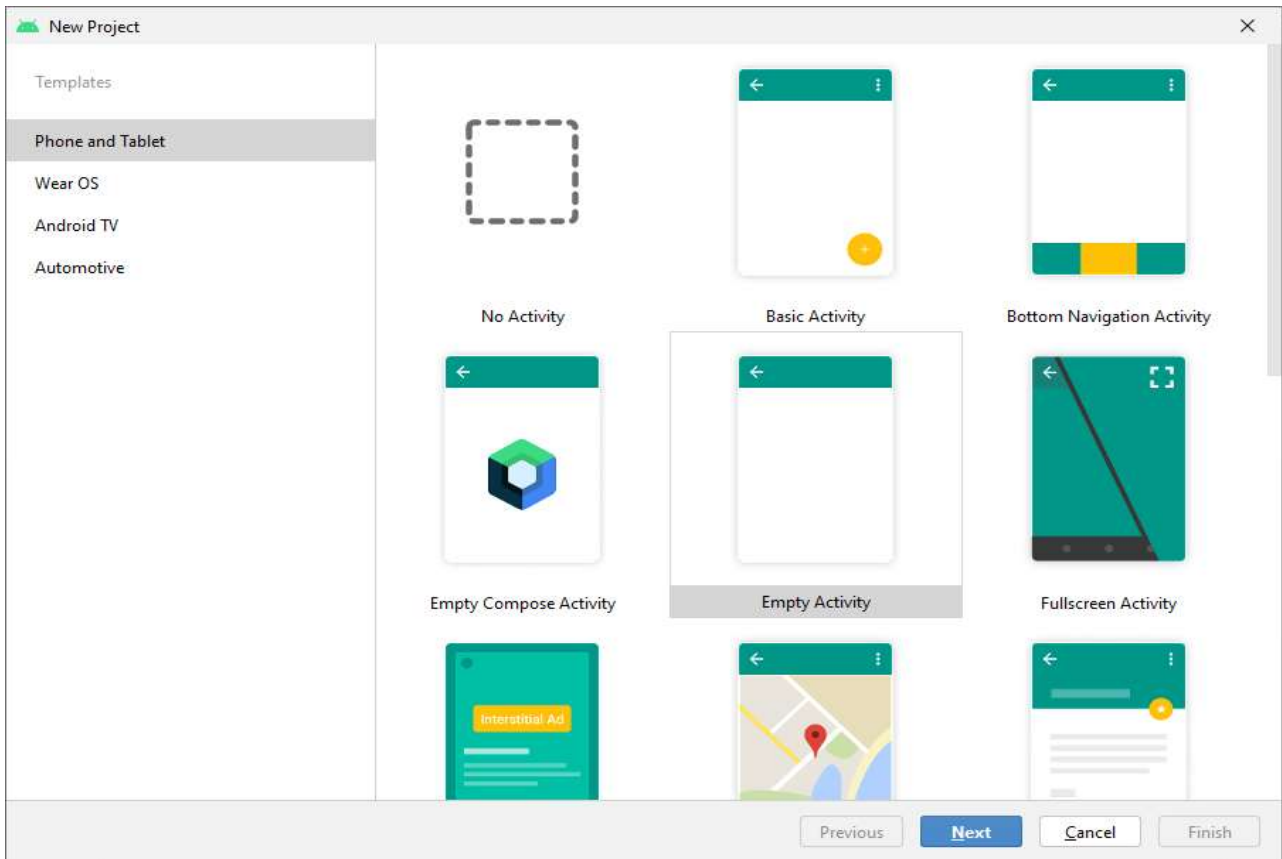
1.8 Creating First Android Project.

To create your new Android project, follow these steps:

1. Open the latest version of Android Studio.

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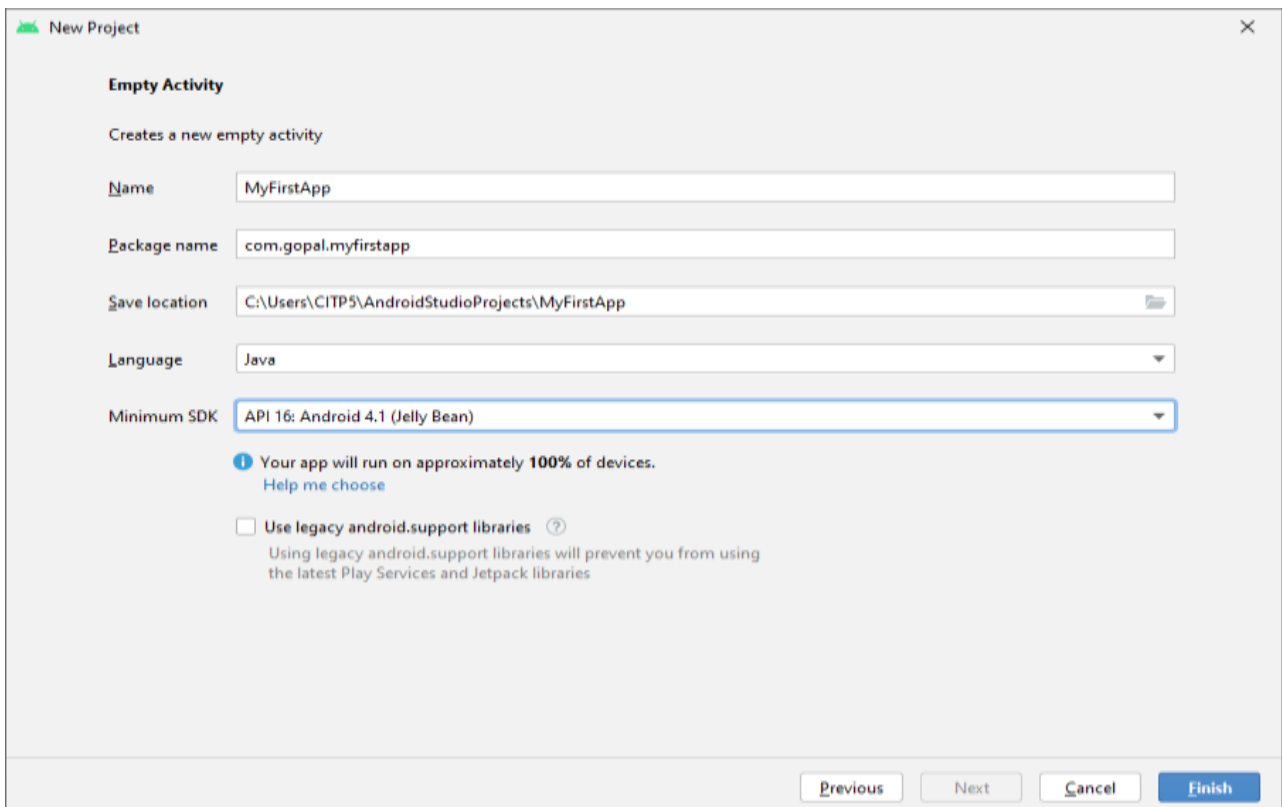

Figure 1. Android Studio welcome screen.

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

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

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

- Enter "MyFirstApp" 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 **Java** from the **Language** drop-down menu.

- Select the lowest version of Android you want your app to support in the **Minimum SDK** field.

5. Click **Finish**.

1.9 Android Project Structure.

Android Studio is the official IDE (Integrated Development Environment) developed by Jet Brains community which is freely provided by Google for android app development.
After completing the setup of Android Architecture we can create android application in the studio.
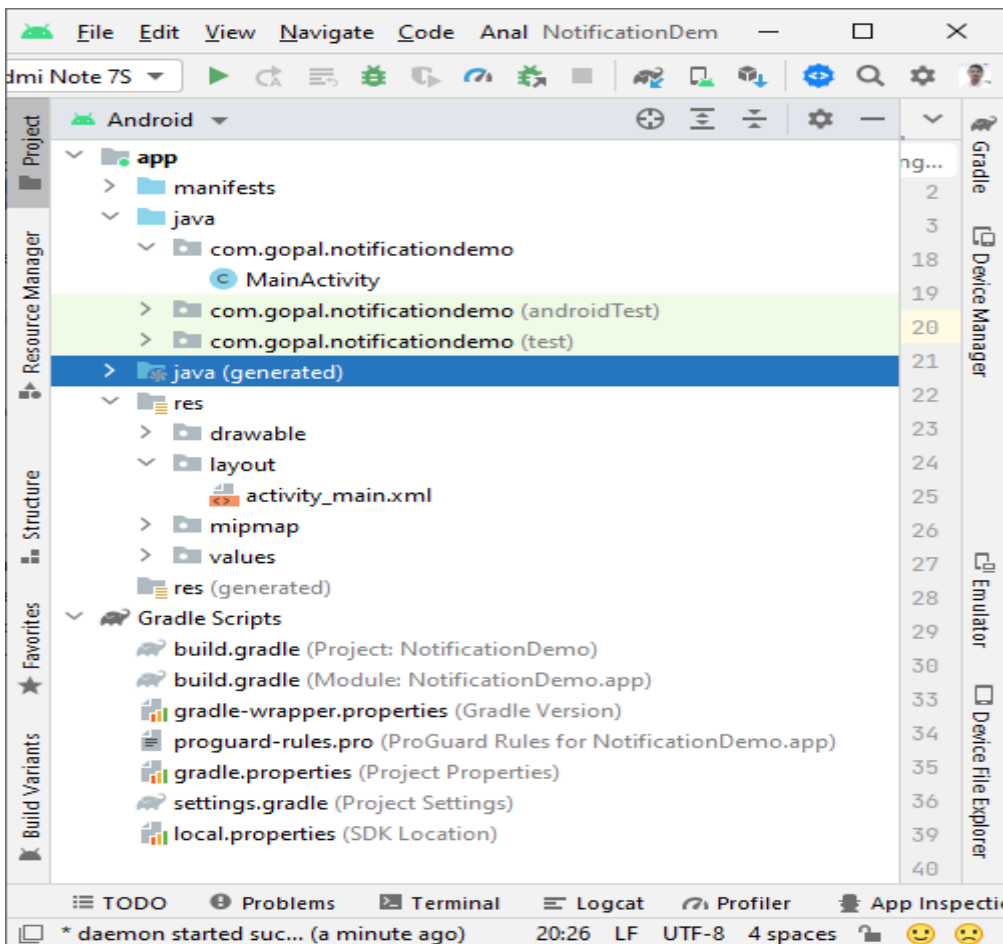
The android project contains different types of app modules, source code files, and resource files. We will explore all the folders and files in the android app.

1. Manifests Folder
2. Java Folder
3. res (Resources) Folder
   - Drawable Folder
   - Layout Folder
   - Mipmap Folder
   - Values Folder
4. Gradle Scripts

Manifests Folder

Manifests folder contains **AndroidManifest.xml** for our creating the android application. This file contains information about our application such as application components. It acts as an intermediate between android OS and our application.

Following is the manifests folder structure in android application.

```xml
activity_main.xml    MainActivity.java    AndroidManifest.xml
1    <?xml version="1.0" encoding="utf-8"?>
2    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3        package="com.gopal.notificationdemo">
4
5        <application
6            android:allowBackup="true"
7            android:icon="@mipmap/ic_launcher"
8            android:label="NotificationDemo"
9            android:roundIcon="@mipmap/ic_launcher_round"
10           android:supportsRtl="true"
11           android:theme="@style/Theme.NotificationDemo">
12           <activity
13               android:name=".MainActivity"
14               android:exported="true">
15               <intent-filter>
16                   <action android:name="android.intent.action.MAIN" />
17
18                   <category android:name="android.intent.category.LAUNCHER" />
19               </intent-filter>
20           </activity>
21       </application>
22
23   </manifest>
```

Java folder

Java folder contains all the java source code (.java) files that we create during the app development, including other Test files. If we create any new project using Java, by default the class file MainActivity.java file will create automatically under the package name "com.gopal.notificationdemo" .

```
∨  📁 java
   ∨  📁 com.gopal.notificationdemo
         Ⓒ MainActivity
   >  📁 com.gopal.notificationdemo (androidTest)
   >  📁 com.gopal.notificationdemo (test)
```
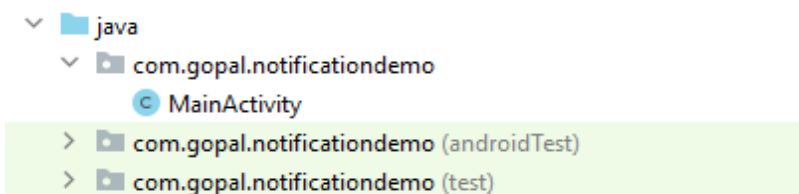
Resource (res) folder

The resource folder is the most important folder because it contains all the non-code sources like images, XML layouts, UI strings for our android application.

res/drawable folder

It contains the different types of images used for the development of the application. We need to add all the images in a drawable folder for the application development.

res/layout folder

The layout folder contains all XML layout files which we used to define the user interface of our application. It contains the **activity_main.xml** file.

res/midmap folder

This folder contains launcher.xml files to define icons that are used to show on the home screen. It contains different density types of icons depending upon the size of the device such as hdpi, mdpi, xhdpi.

res/values folder

Values folder contains a number of XML files like strings, dimensions, colors and styles definitions. One of the most important file is **strings.xml** file which contains the resources.

Gradle Scripts folder

Gradle means automated build system and it contains number of files which are used to define a build configuration which can be apply to all modules in our application. In build.gradle (Project) there are buildscripts and in build.gradle (Module) plugins and implementations are used to build configurations that can be applied to all our application modules.

Thank You