ADVANCED MOBLIE PROGRAMMING JOURNAL

INDEX

Practical 1

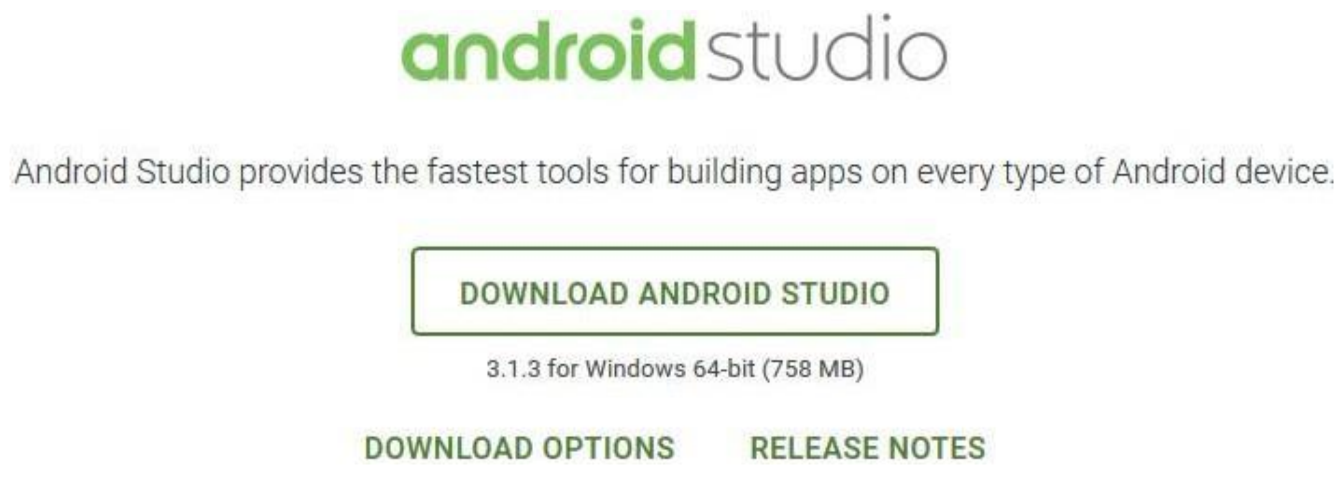
Building a simple Hello World Application

Steps to Install Android Studio

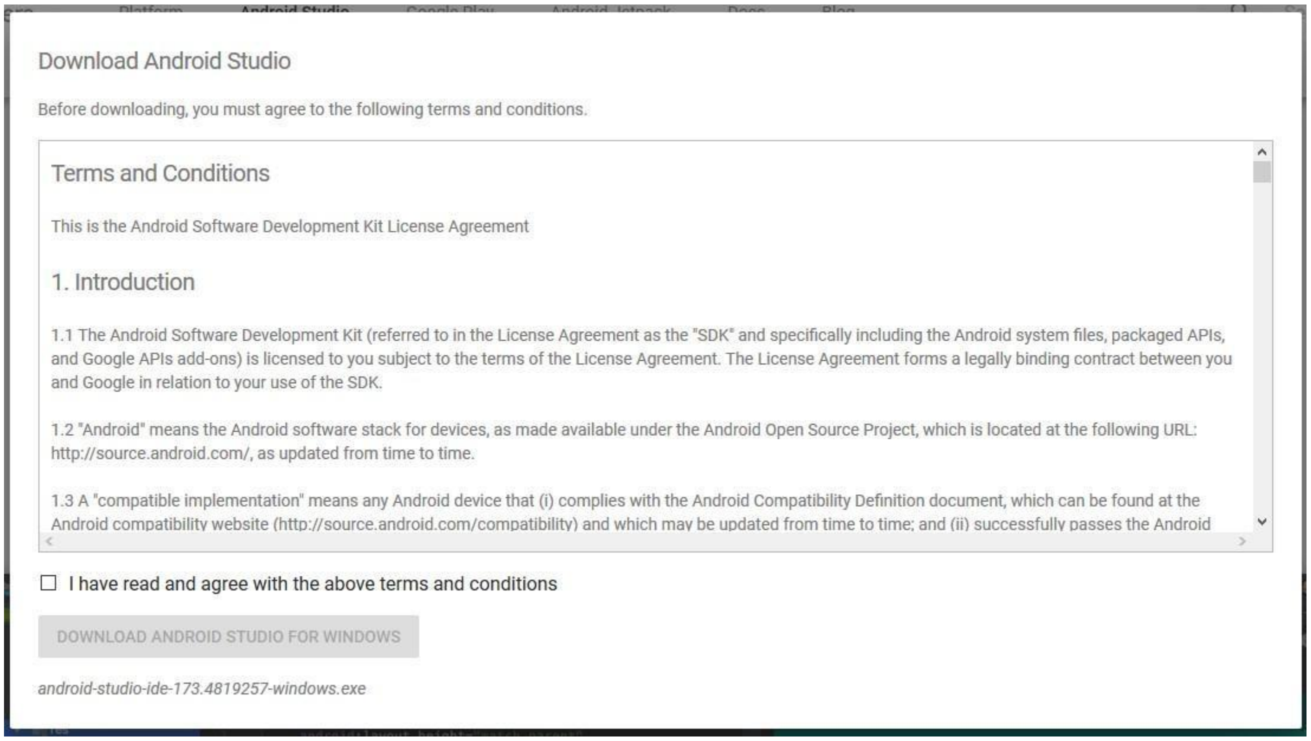
Step – 1:
Head over to bellow link to get the Android Studio executable or zip file .

https://developer.android.com/studio/#downloads

Step – 2:
Click on the download android studio button .


android studio
Android Studio provides the fastest tools for building apps on every type of Android device.
DOWNLOAD ANDROID STUDIO
3.1.3 for Windows 64-bit (758 MB)
DOWNLOAD OPTIONS
RELEASE NOTES

Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.



Download Android Studio

Before downloading, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

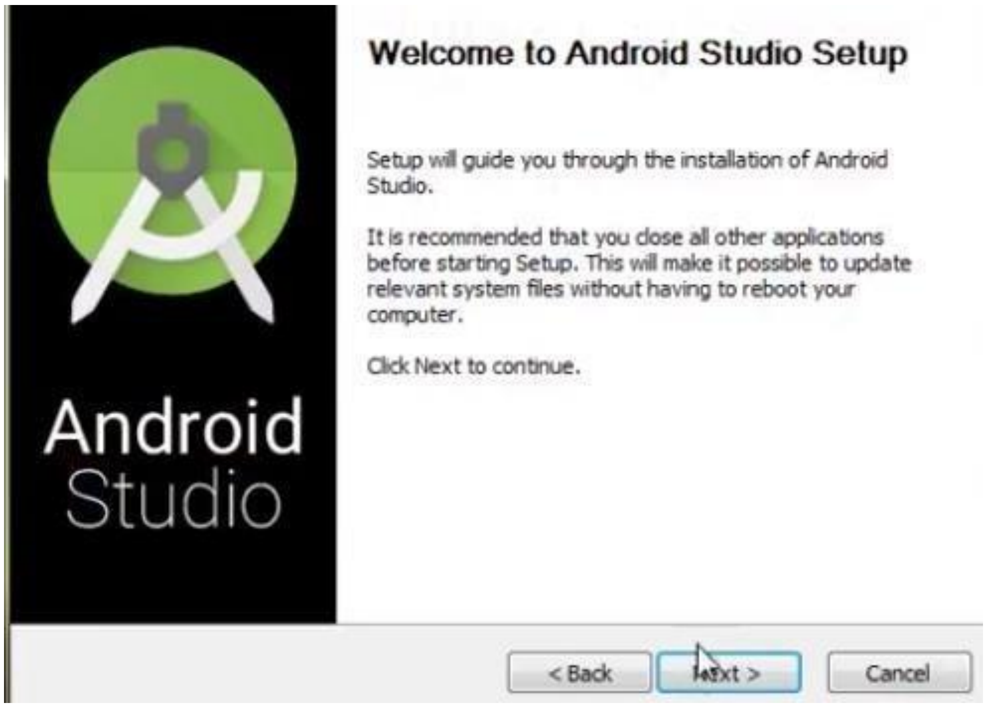
1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (http://source.android.com/compatibility) and which may be updated from time to time; and (ii) successfully passes the Android

☐ I have read and agree with the above terms and conditions

android-studio-ide-173.4819257-windows.exe

Click on save file button in the appeared prompt box and the file will start downloading.

Step – 3:
After the downloading has finished, open the file from downloads and run it .
It will prompt the following dialogue box



Welcome to Android Studio Setup

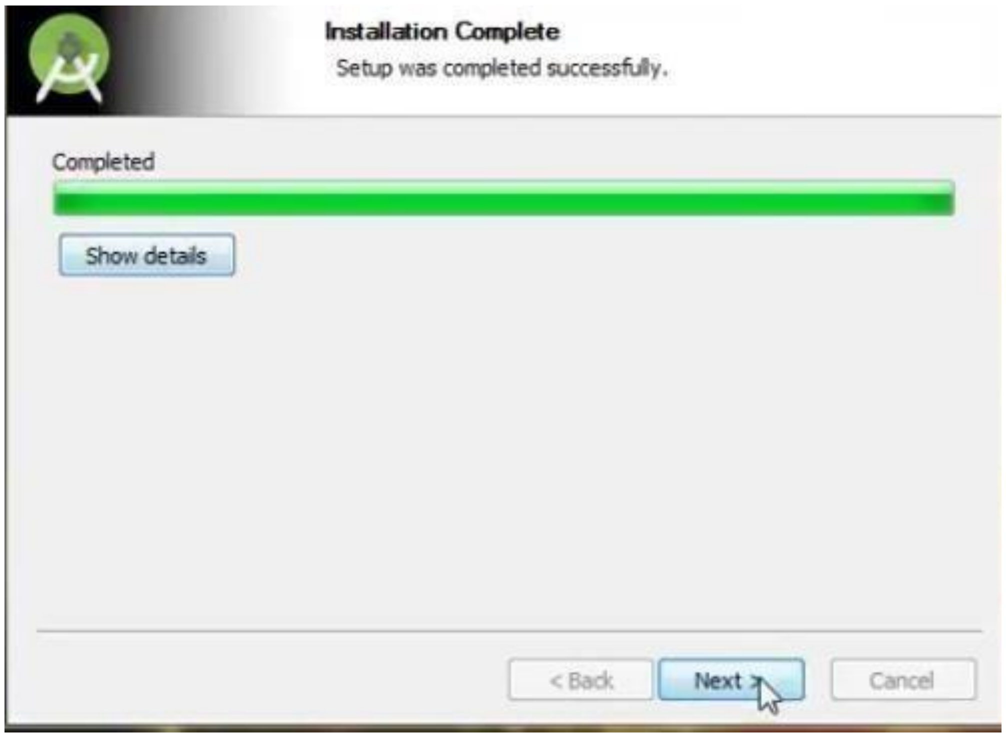
Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

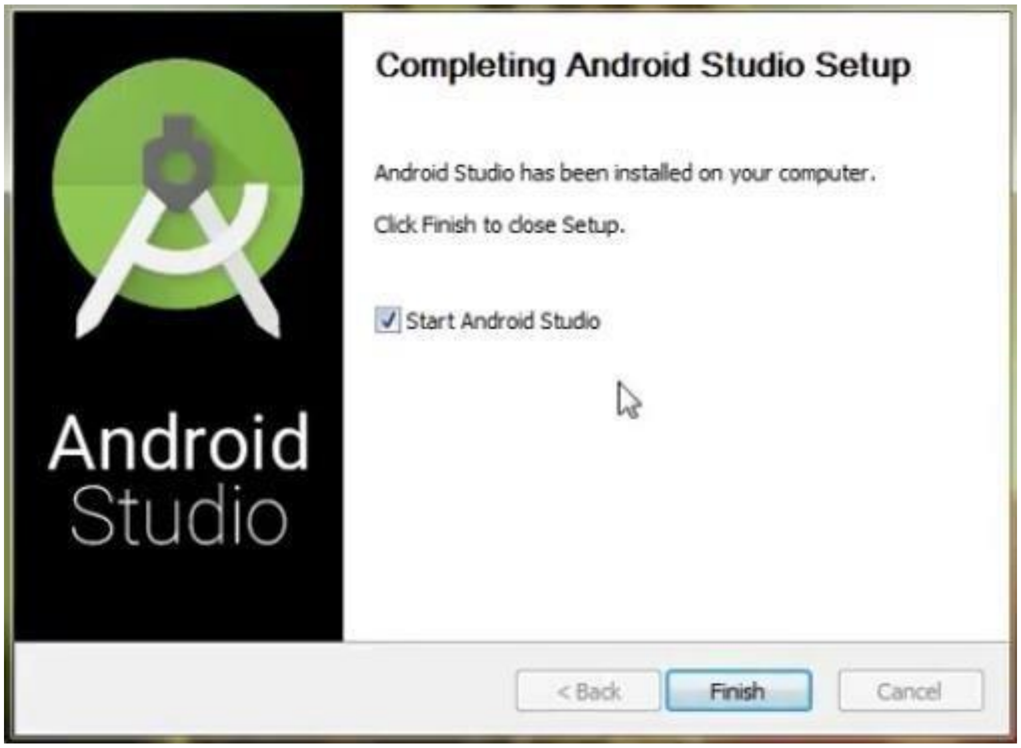
Click on next.
In the next prompt it'll ask for a path for installation. Choose a path and hit next.

Step – 4:
It will start the installation, and once it is completed, it will be like the image shown below


Installation Complete
Setup was completed successfully.

Completed

Click on next


Completing Android Studio Setup

Android Studio has been installed on your computer.

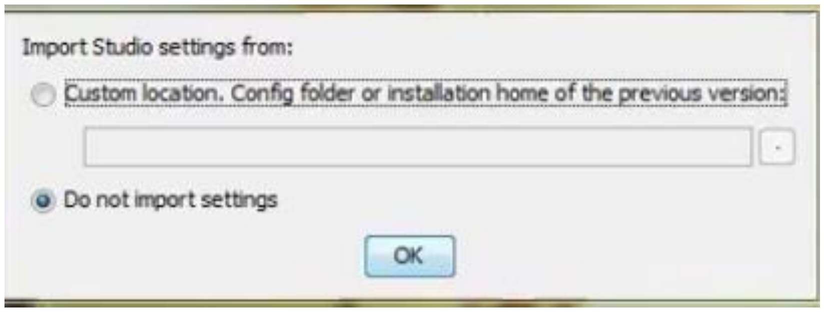
Click Finish to close Setup.

[✓] Start Android Studio

Step – 5 :

Once "Finish" is clicked, it will ask whether the previous settings needs to be imported [if android studio had been installed earlier], or not.

It is better to choose the 'Don't import Settings option' .

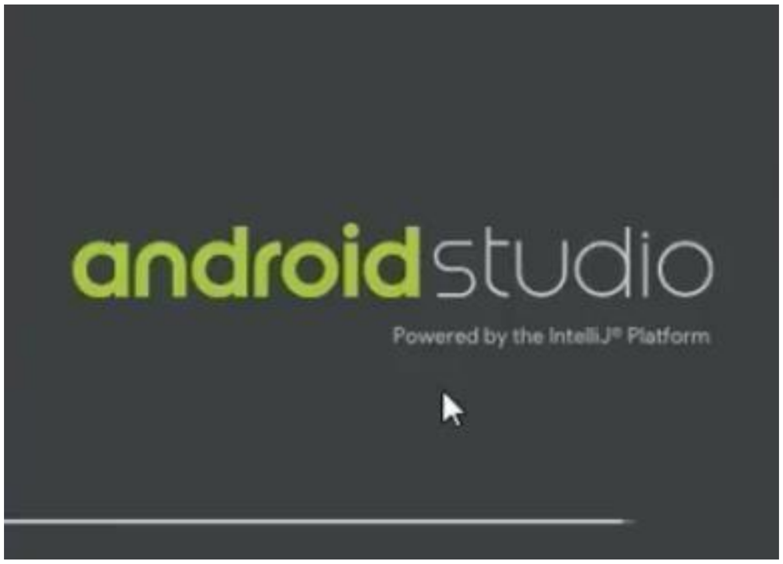

Import Studio settings from:

○ Custom location. Config folder or installation home of the previous version:

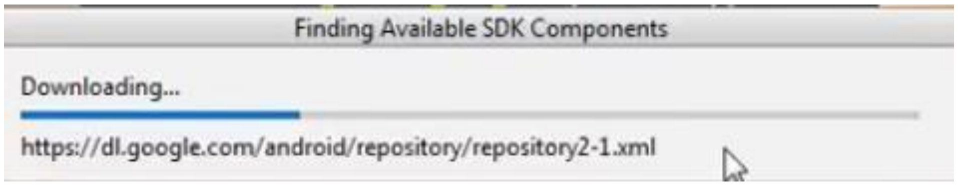
● Do not import settings

Step – 6 :

This will start the Android Studio.



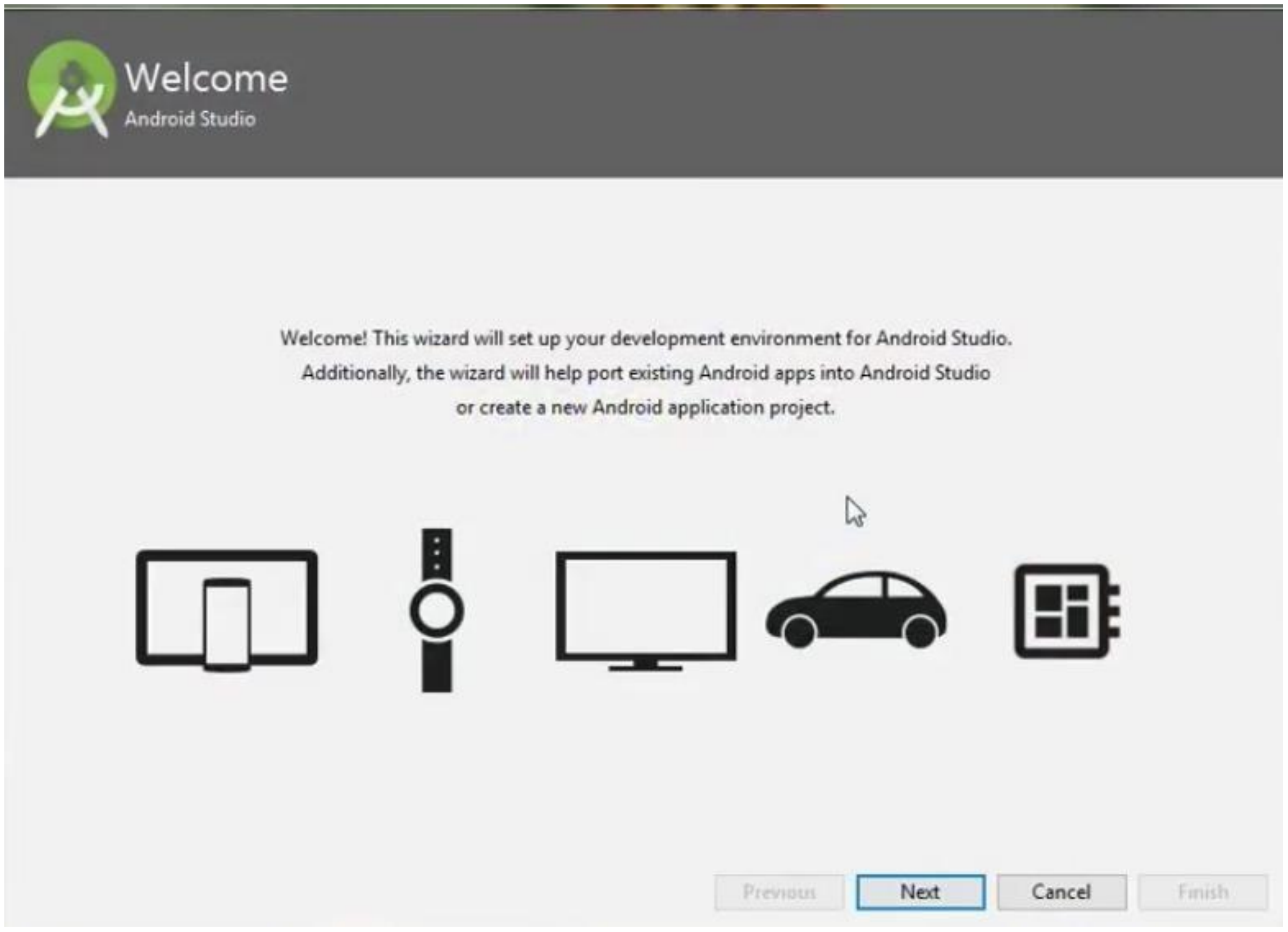
Meanwhile it will be finding the available SDK components .


Finding Available SDK Components

Downloading...

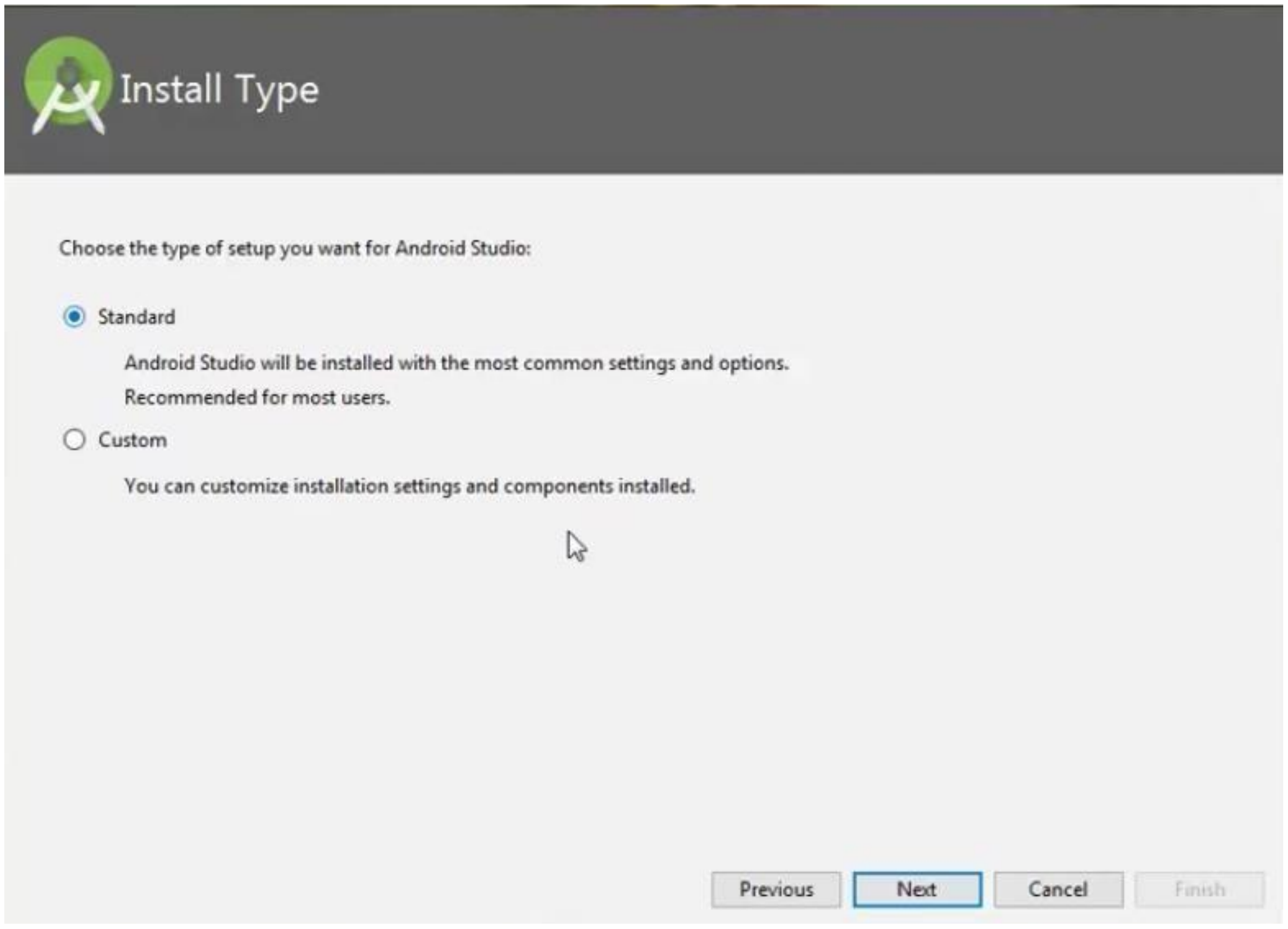
https://dl.google.com/android/repository/repository2-1.xml

Step – 7:
After it has found the SDK components, it will redirect to the Welcome dialog box .



Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.

Click on next .


Install Type

Choose the type of setup you want for Android Studio:

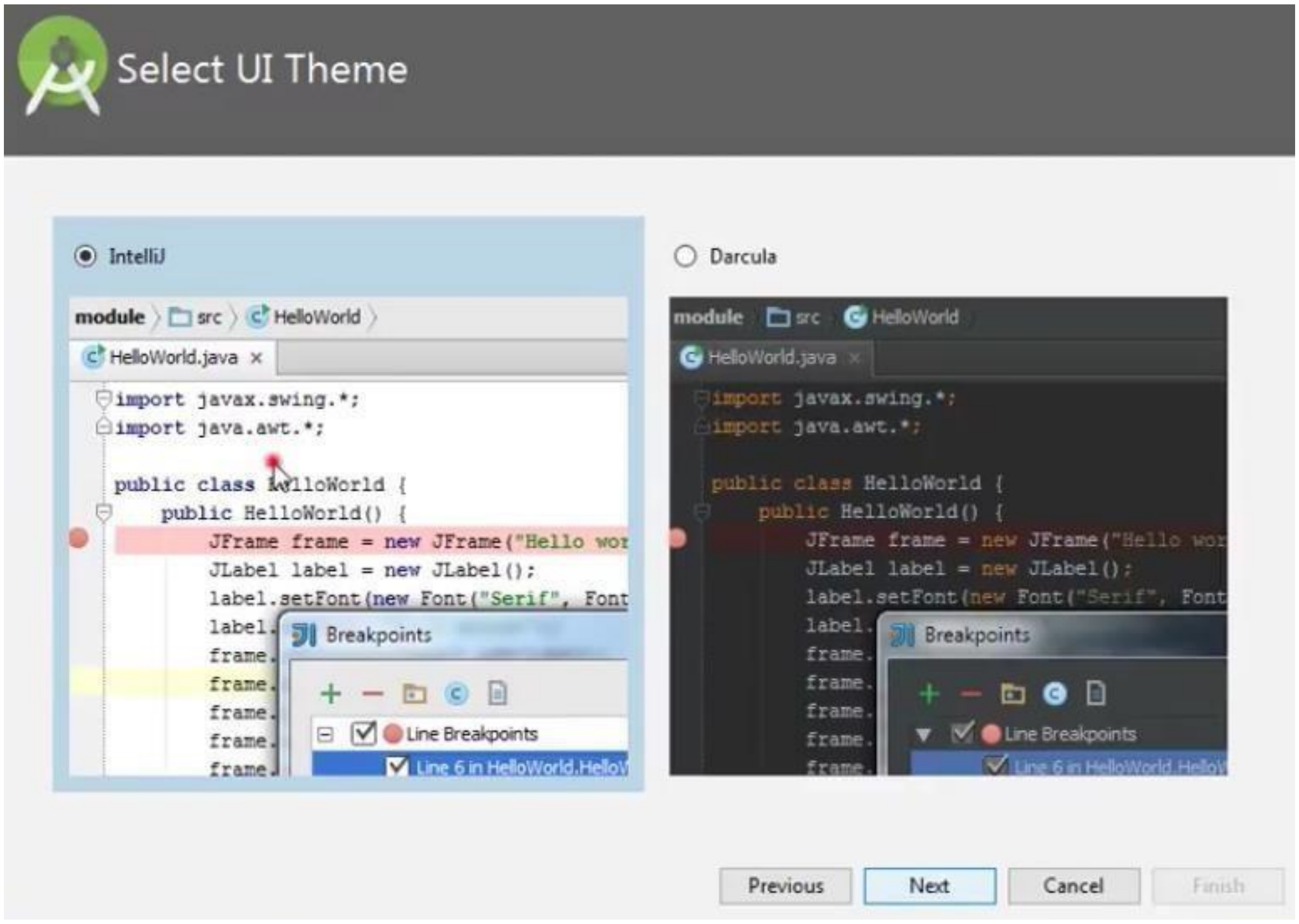
[✓ Standard]

Android Studio will be installed with the most common settings and options.
Recommended for most users.

Custom

You can customize installation settings and components installed.

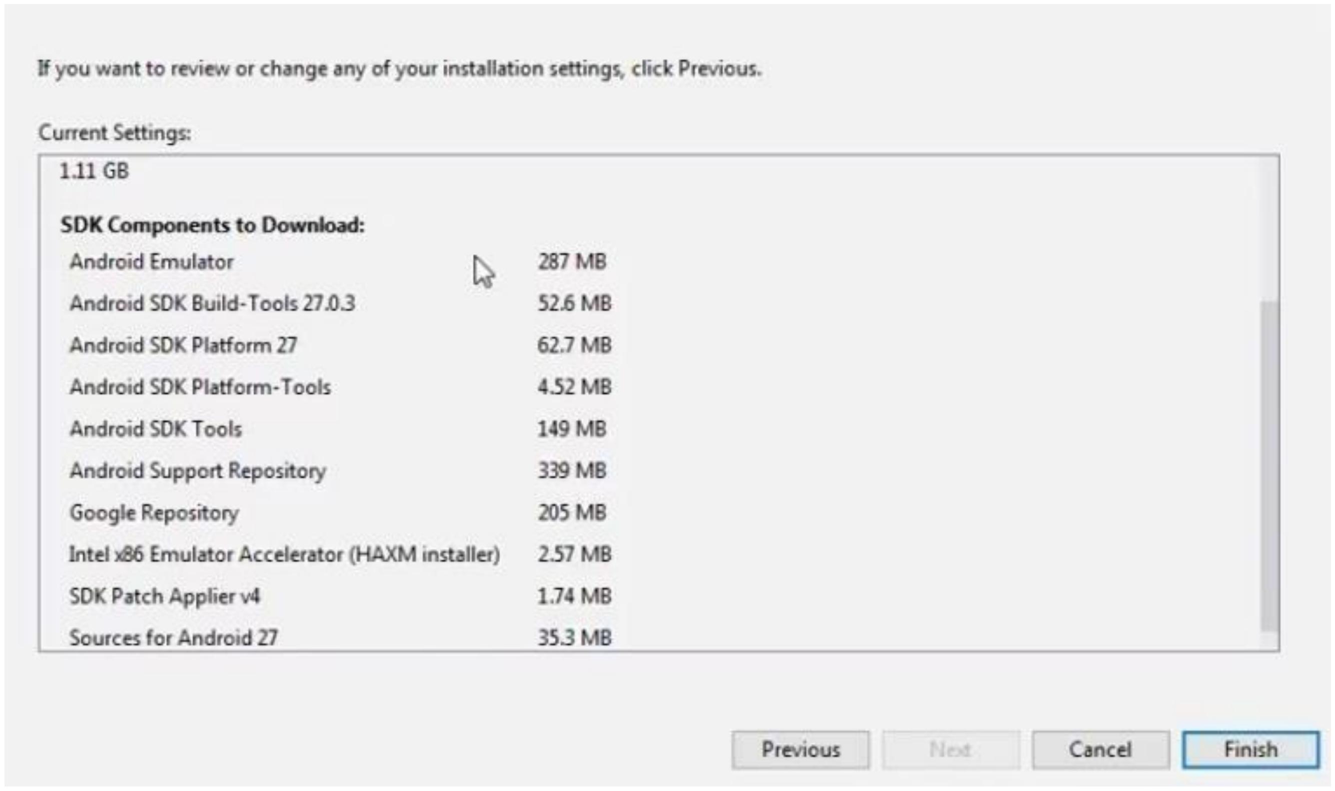
Choose Standard and click on Next.
Now choose the theme, whether Light theme or the Dark one .
The light one is called the IntelliJ theme whereas the dark theme is called Darcula . Choose as required.



```
import javax.swing.*;
import java.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame("Hello wor
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
```

```
import javax.swing.*;
import java.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame("Hello wor
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
```

- Click on the Next button
- Step – 8 :
  Now it is time to download the SDK components .



If you want to review or change any of your installation settings, click Previous.

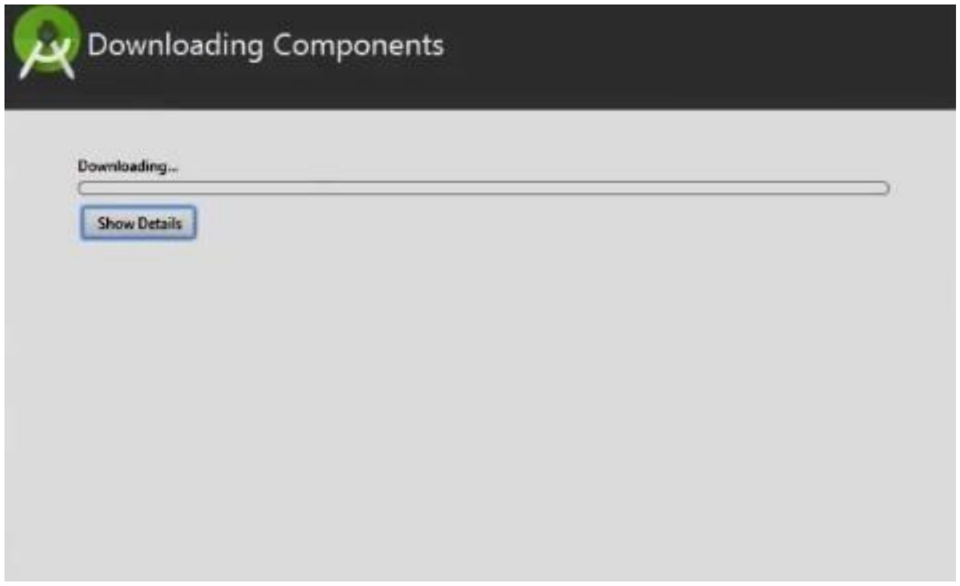
Current Settings:

1.11 GB

SDK Components to Download:

Android Emulator	287 MB
Android SDK Build-Tools 27.0.3	52.6 MB
Android SDK Platform 27	62.7 MB
Android SDK Platform-Tools	4.52 MB
Android SDK Tools	149 MB
Android Support Repository	339 MB
Google Repository	205 MB
Intel x86 Emulator Accelerator (HAXM installer)	2.57 MB
SDK Patch Applier v4	1.74 MB
Sources for Android 27	35.3 MB

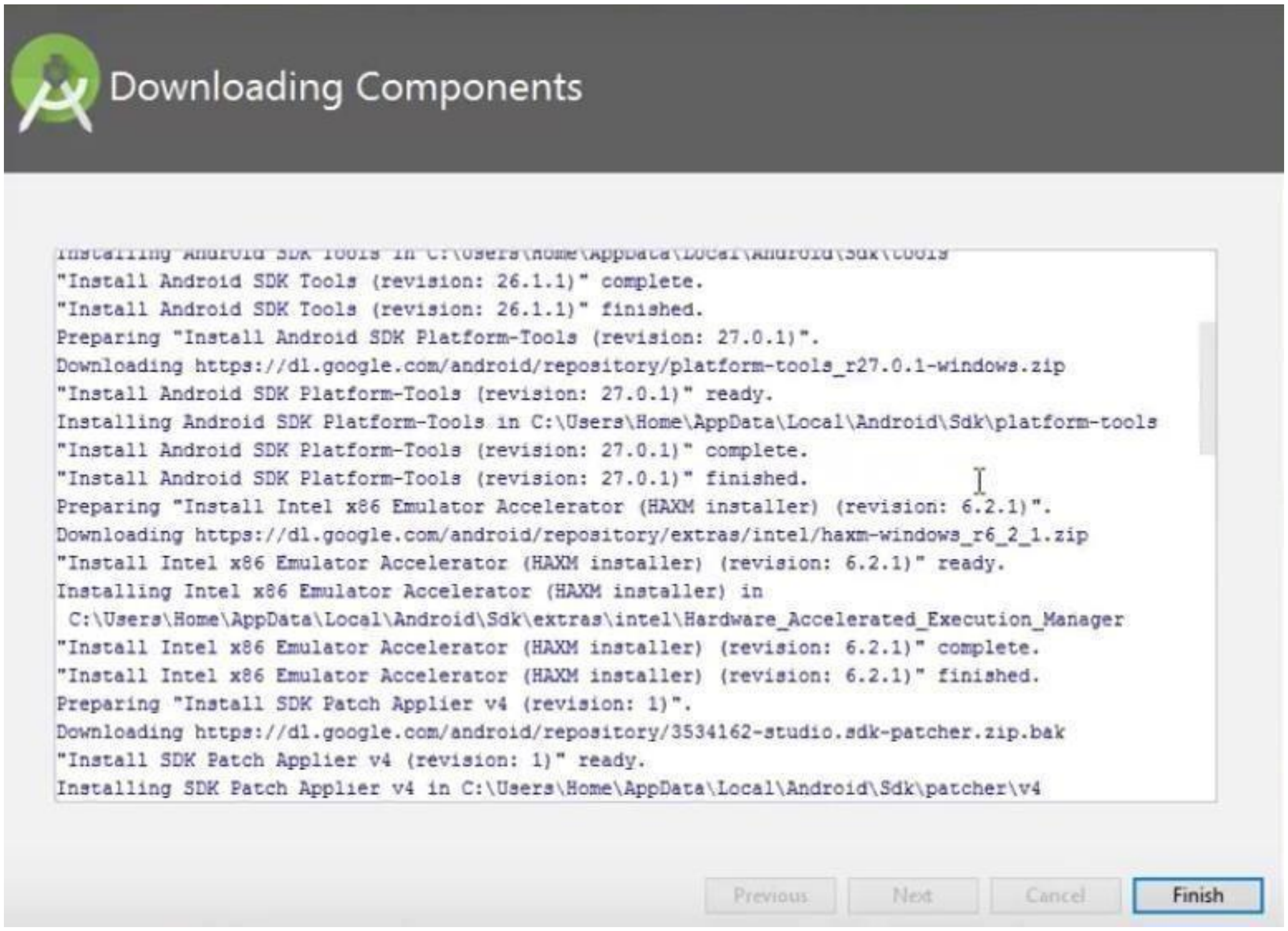
Click on Finish .



Downloading Components

Downloading...

It has started downloading the components



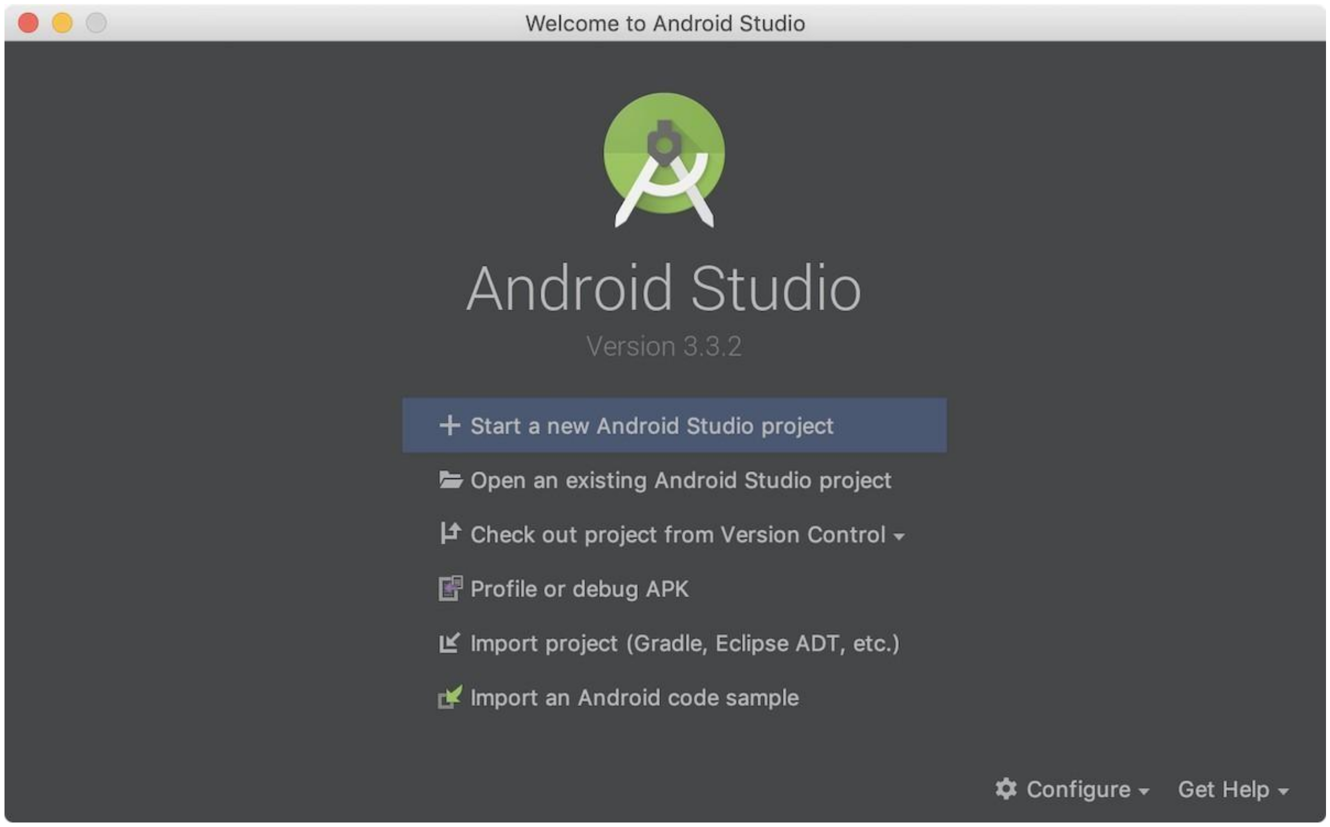
```
Installing Android SDK Tools in C:\Users\Home\AppData\Local\Android\Sdk\tools
"Install Android SDK Tools (revision: 26.1.1)" complete.
"Install Android SDK Tools (revision: 26.1.1)" finished.
Preparing "Install Android SDK Platform-Tools (revision: 27.0.1)".
Downloading https://dl.google.com/android/repository/platform-tools_r27.0.1-windows.zip
"Install Android SDK Platform-Tools (revision: 27.0.1)" ready.
Installing Android SDK Platform-Tools in C:\Users\Home\AppData\Local\Android\Sdk\platform-tools
"Install Android SDK Platform-Tools (revision: 27.0.1)" complete.
"Install Android SDK Platform-Tools (revision: 27.0.1)" finished.
Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)".
Downloading https://dl.google.com/android/repository/extras/intel/haxm-windows_r6_2_1.zip
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" ready.
Installing Intel x86 Emulator Accelerator (HAXM installer) in
 C:\Users\Home\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" complete.
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" finished.
Preparing "Install SDK Patch Applier v4 (revision: 1)".
Downloading https://dl.google.com/android/repository/3534162-studio.sdk-patcher.zip.bak
"Install SDK Patch Applier v4 (revision: 1)" ready.
Installing SDK Patch Applier v4 in C:\Users\Home\AppData\Local\Android\Sdk\patcher\v4
```

The Android Studio has been successfully configured.
Now it's time to launch and build apps.

Steps to create a project in android

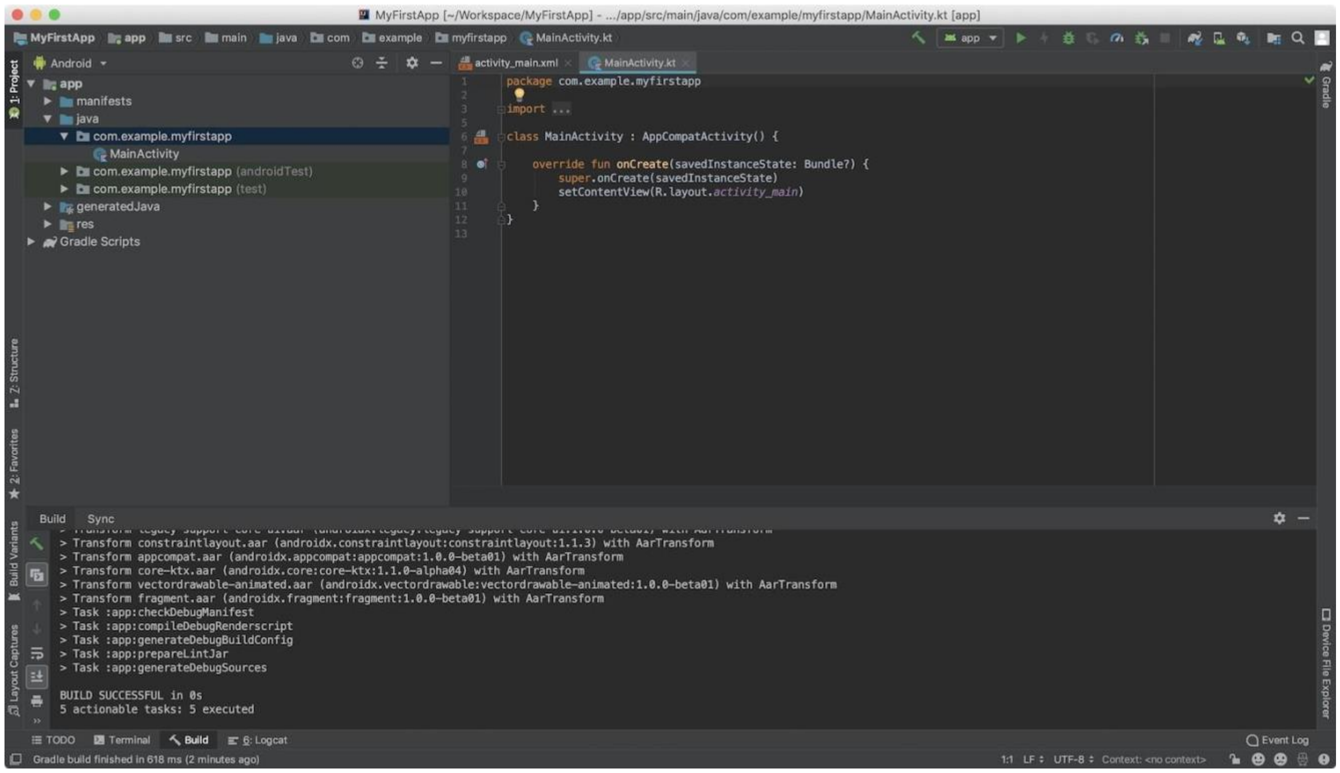
To create your new Android project, follow these steps:

- Install the latest version of Android Studio.
- In the Welcome to Android Studio window, click Start a new Android Studio project.


Welcome to Android Studio
Android Studio
Version 3.3.2
+ Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

- If you have a project already opened, select File > New > New Project.
- In the Choose your project window, select Empty Activity and click Next.
- In the Configure your project window, complete the following:
    1. Enter "My First App" in the Name field.
    2. Enter "com.example.myfirstapp" in the Package name field.
    3. If you'd like to place the project in a different folder, change its Save location.
    4. Select either Java or Kotlin from the Language drop-down menu.
- Select the checkbox next to Use androidx.* artifacts.
- Leave the other options as they are.
- Click Finish.

After some processing time, the Android Studio main window appears


MyFirstApp [~/Workspace/MyFirstApp] - .../app/src/main/java/com/example/myfirstapp/MainActivity.kt [app]
MyFirstApp app src main java com example myfirstapp MainActivity.kt
Android
app
manifests
java
com.example.myfirstapp
MainActivity
com.example.myfirstapp (androidTest)
com.example.myfirstapp (test)
generatedJava
res
Gradle Scripts
activity_main.xml
MainActivity.kt
Build
Sync
TODO
Terminal
Build
6: Logcat
Event Log
Gradle build finished in 618 ms (2 minutes ago)

```kotlin
package com.example.myfirstapp

import ...

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

```
> Transform constraintlayout.aar (androidx.constraintlayout:constraintlayout:1.1.3) with AarTransform
> Transform appcompat.aar (androidx.appcompat:appcompat:1.0.0-beta01) with AarTransform
> Transform core-ktx.aar (androidx.core:core-ktx:1.1.0-alpha04) with AarTransform
> Transform vectordrawable-animated.aar (androidx.vectordrawable:vectordrawable-animated:1.0.0-beta01) with AarTransform
> Transform fragment.aar (androidx.fragment:fragment:1.0.0-beta01) with AarTransform
> Task :app:checkDebugManifest
> Task :app:compileDebugRenderscript
> Task :app:generateDebugBuildConfig
> Task :app:prepareLintJar
> Task :app:generateDebugSources

BUILD SUCCESSFUL in 0s
5 actionable tasks: 5 executed
```

Note

To Open Project Window

select View > Tool Windows > Project

To Open MainActivity.java file

app > java > PackageName > MainActivity.java

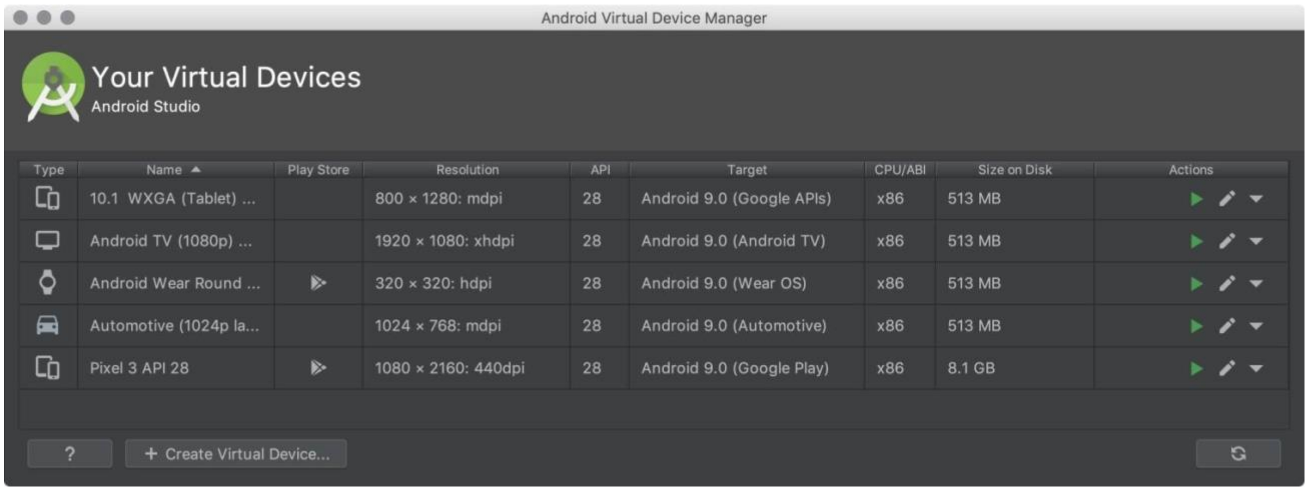
To Open Layout activity_main.xml file

app > res > layout > activity_main.xml

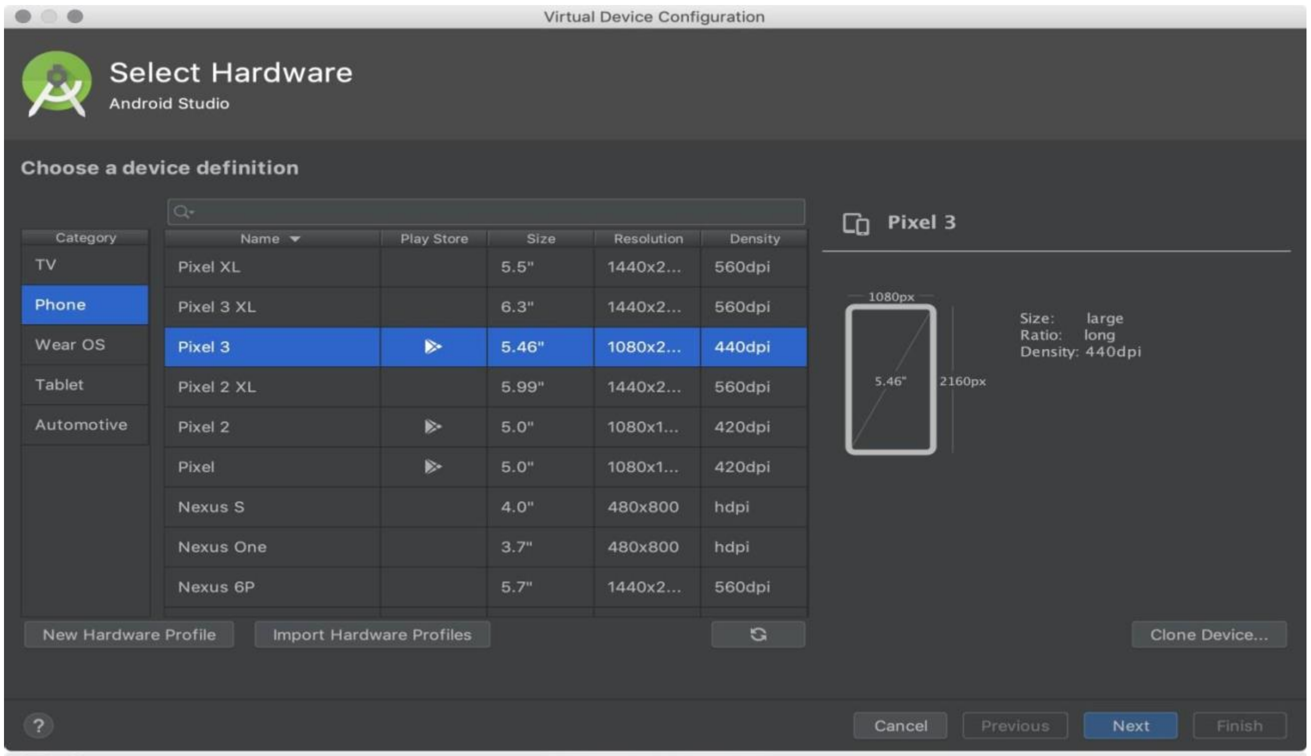
1. After the project is created, there are 2 files, **MainActivity.java and activity_main.xml**
2. Go to activity_main.xml and select **Design View**
3. In Design View, change the layout to **LinearLayout(Vertical)** select **Add TextView,** and change the text to “Hello World!”
4. Click on **Run** and select the AVD already created(if not created, first create the AVD)
5. Output screen should show “Hello World”

To create a new AVD:

1. Open the **AVD** Manager by clicking Tools > **AVD** Manager.
2. Click **Create Virtual Device**, at the bottom of the **AVD** Manager dialog. ...
3. Select a hardware profile, and then click Next.
4. Select the system image for a particular API level, and then click Next.
5. Change **AVD** properties as needed, and then click Finish.



Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	10.1 WXGA (Tablet) ...		800 × 1280: mdpi	28	Android 9.0 (Google APIs)	x86	513 MB	▶
	Android TV (1080p) ...		1920 × 1080: xhdpi	28	Android 9.0 (Android TV)	x86	513 MB	▶
	Android Wear Round ...	▶	320 × 320: hdpi	28	Android 9.0 (Wear OS)	x86	513 MB	▶
	Automotive (1024p la...		1024 × 768: mdpi	28	Android 9.0 (Automotive)	x86	513 MB	▶
	Pixel 3 API 28	▶	1080 × 2160: 440dpi	28	Android 9.0 (Google Play)	x86	8.1 GB	▶



Virtual Device Configuration

Select Hardware

Android Studio

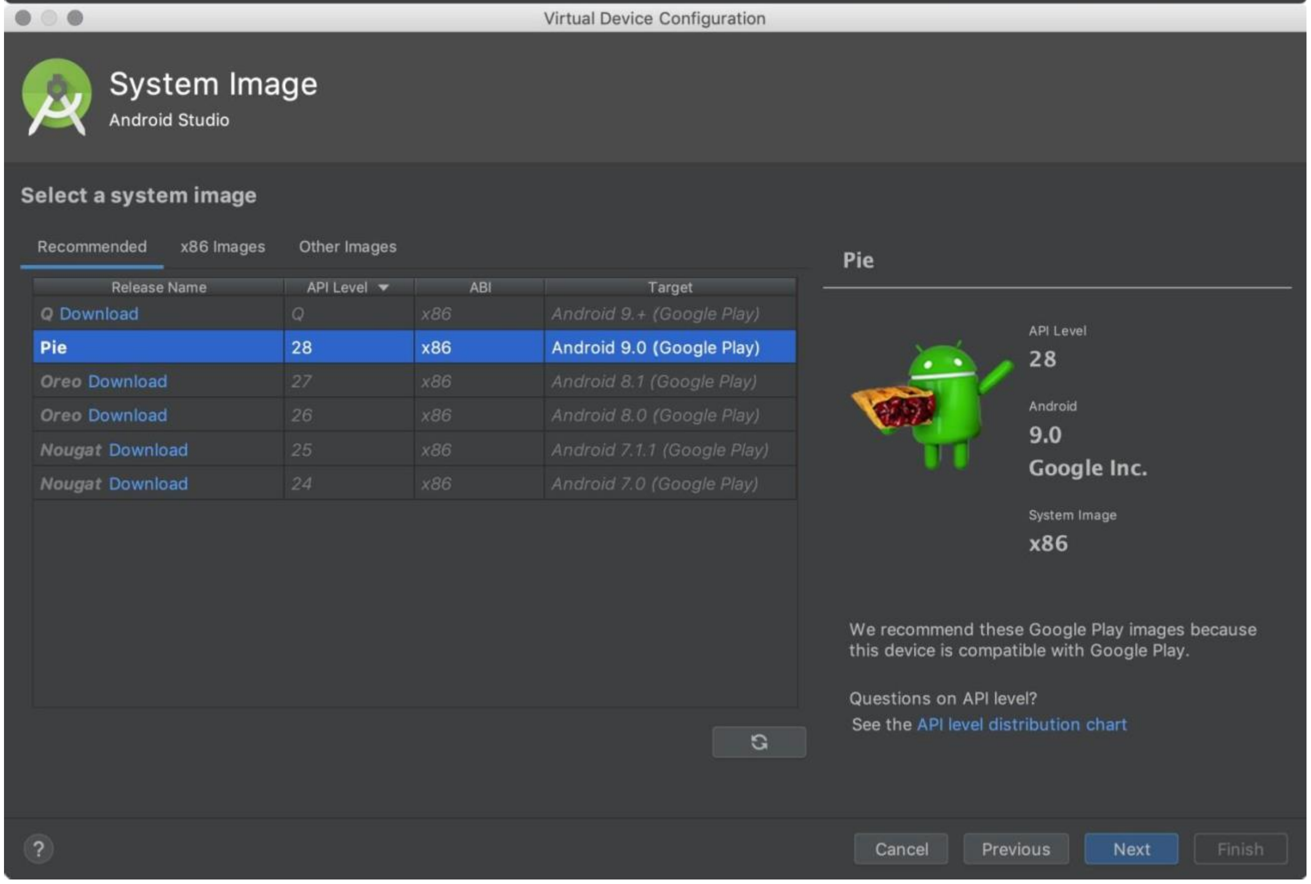
Choose a device definition

Category
TV
Phone
Wear OS
Tablet
Automotive

Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2...	560dpi
Pixel 3 XL		6.3"	1440x2...	560dpi
Pixel 3	▶	5.46"	1080x2...	440dpi
Pixel 2 XL		5.99"	1440x2...	560dpi
Pixel 2	▶	5.0"	1080x1...	420dpi
Pixel	▶	5.0"	1080x1...	420dpi
Nexus S		4.0"	480x800	hdpi
Nexus One		3.7"	480x800	hdpi
Nexus 6P		5.7"	1440x2...	560dpi

Pixel 3

Size: large
Ratio: long
Density: 440dpi



Virtual Device Configuration

System Image

Android Studio

Select a system image

Release Name	API Level	ABI	Target
Q Download	Q	x86	Android 9.+ (Google Play)
Pie	28	x86	Android 9.0 (Google Play)
Oreo Download	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

Pie

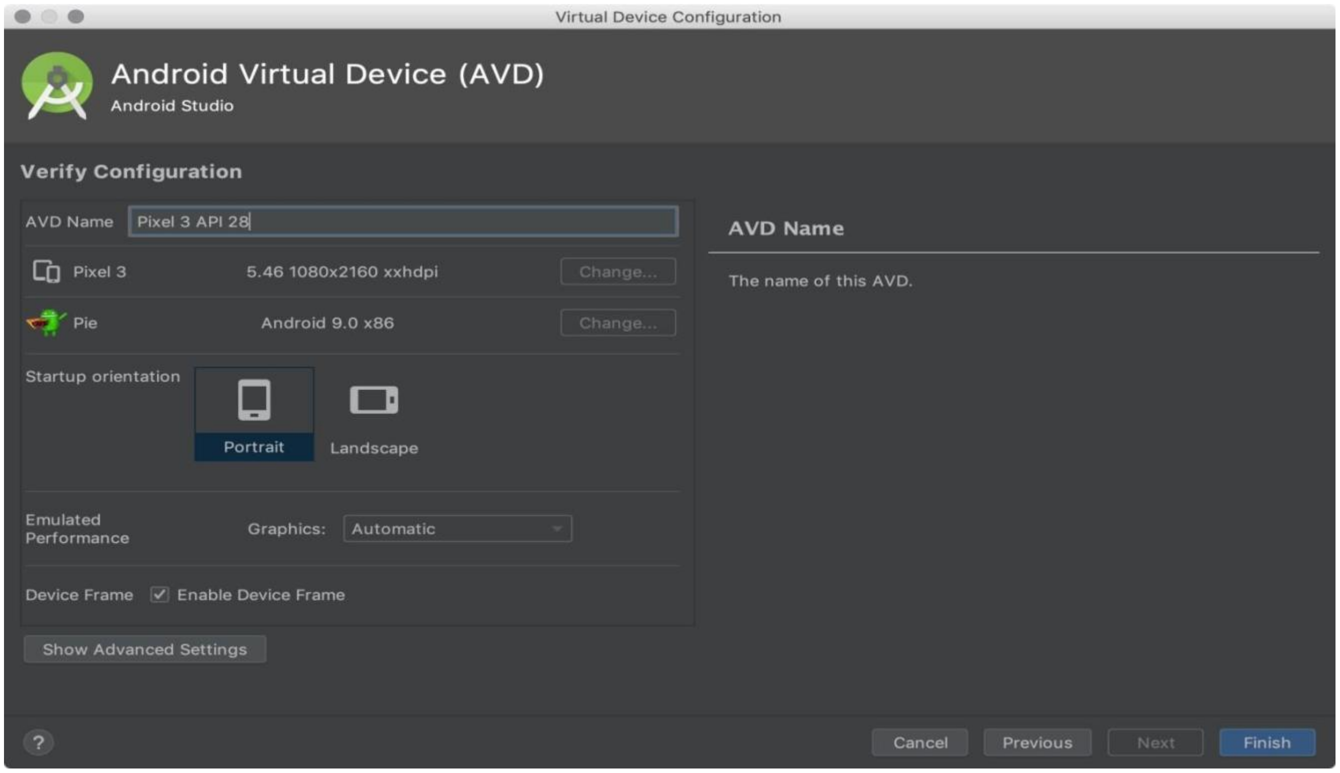
API Level
28

Android
9.0
Google Inc.

System Image
x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
See the API level distribution chart


Virtual Device Configuration

Android Virtual Device (AVD)
Android Studio

Verify Configuration

AVD Name: Pixel 3 API 28

Pixel 3 — 5.46 1080x2160 xxhdpi — Change...

Pie — Android 9.0 x86 — Change...

Startup orientation: Portrait (selected), Landscape

Emulated Performance — Graphics: Automatic

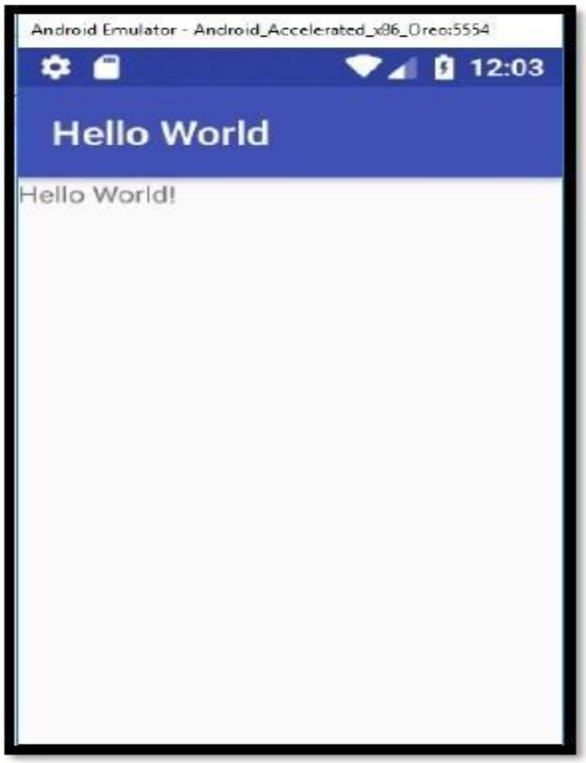
Device Frame — ✓ Enable Device Frame

Show Advanced Settings

AVD Name

The name of this AVD.

Output


Android Emulator - Android_Accelerated_x86_Oreo:5554
12:03
Hello World

Hello World!

Practical 2

Programming Resources
Android Resources: (Color, Theme, String, Drawable, Dimension, Image)

a) <u>Defining Color Property.</u>

1. Create a new project and go to:
   ProjectName>App>src>main>res>values>**colors.xml**
2. Defining new color properties in **colors.xml**

Go to
ProjectName->app->res->values->colors.xml

Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
<color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>

   // Define new colors here using hexadecimal color codes.

   <color name="Red">#FF0000</color>
   <color name="Blue">#0000FF</color>
   <color name="Pink">#FF00FF</color>
   <color name="Cyan">#00FFFF</color>
   <color name="Grey">#AABBFF</color>

</resources>
```

Default Content

Add this code in the colors.xml

Now go to **activity_main.xml** and type the following code:

- Drag and drop LinearLayout(Vertical) From Layout tab in Palette window
- Drag and drop four TextView from Text tab in palette window
- The default Code of one TextView is as follows

```
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="56dp"
android:text="TextView" />
```

Now call the colors from the colors.xml as

```
android:background="@color/Red"
```

Now the TextView Code looks like

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="56dp"
    android:text="TextView"

    android:background="@color/Red"
```

Now the final Code of **activity_main.xml is**

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<LinearLayout
    android:layout_width="409dp"
    android:layout_height="729dp"
    android:orientation="vertical"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:text="TextView"

        android:background="@color/Red"
```

Add this line only

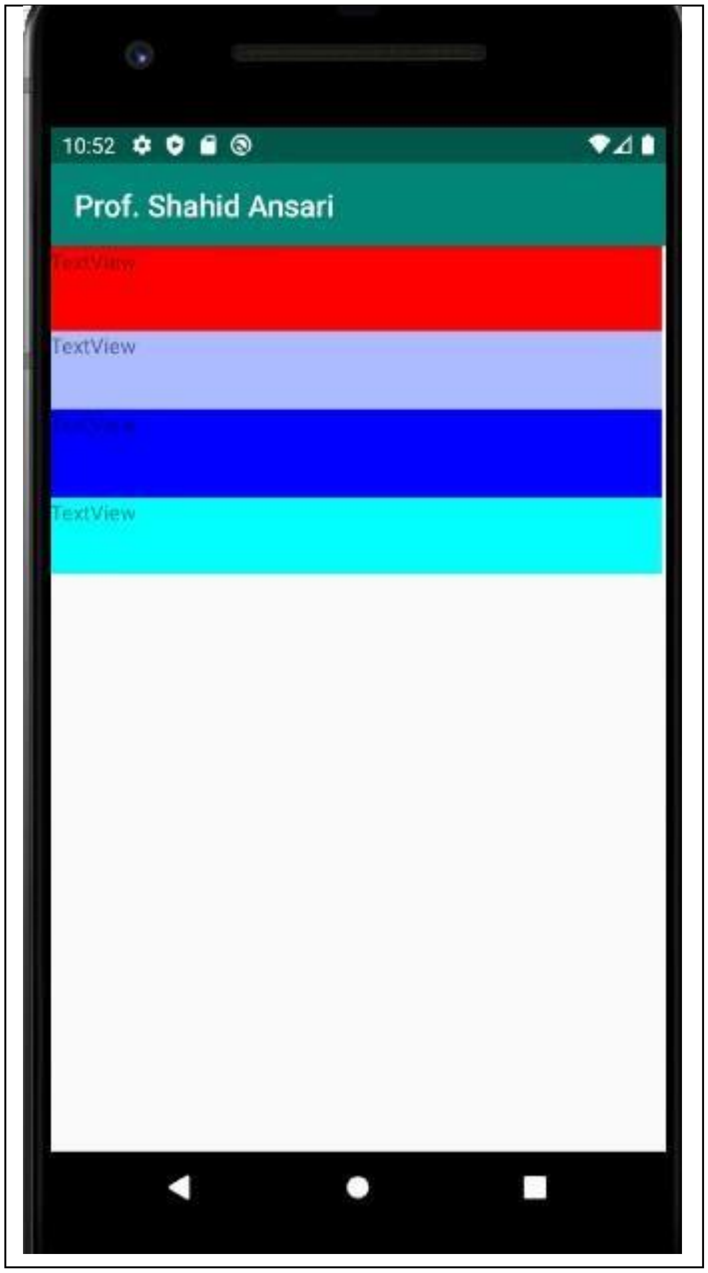
```
        />

      <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="52dp"
        android:text="TextView"

        android:background="@color/Grey"
        />

      <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="59dp"
        android:text="TextView"

        android:background="@color/Blue"
        />

      <TextView
        android:id="@+id/textView4"
        android:layout_width="match_parent"
        android:layout_height="51dp"
        android:text="TextView"

        android:background="@color/Cyan"
        />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Add this line only

Add this line only

Add this lineonly

Output



We are done with Colors

b) Defining Theme Property.

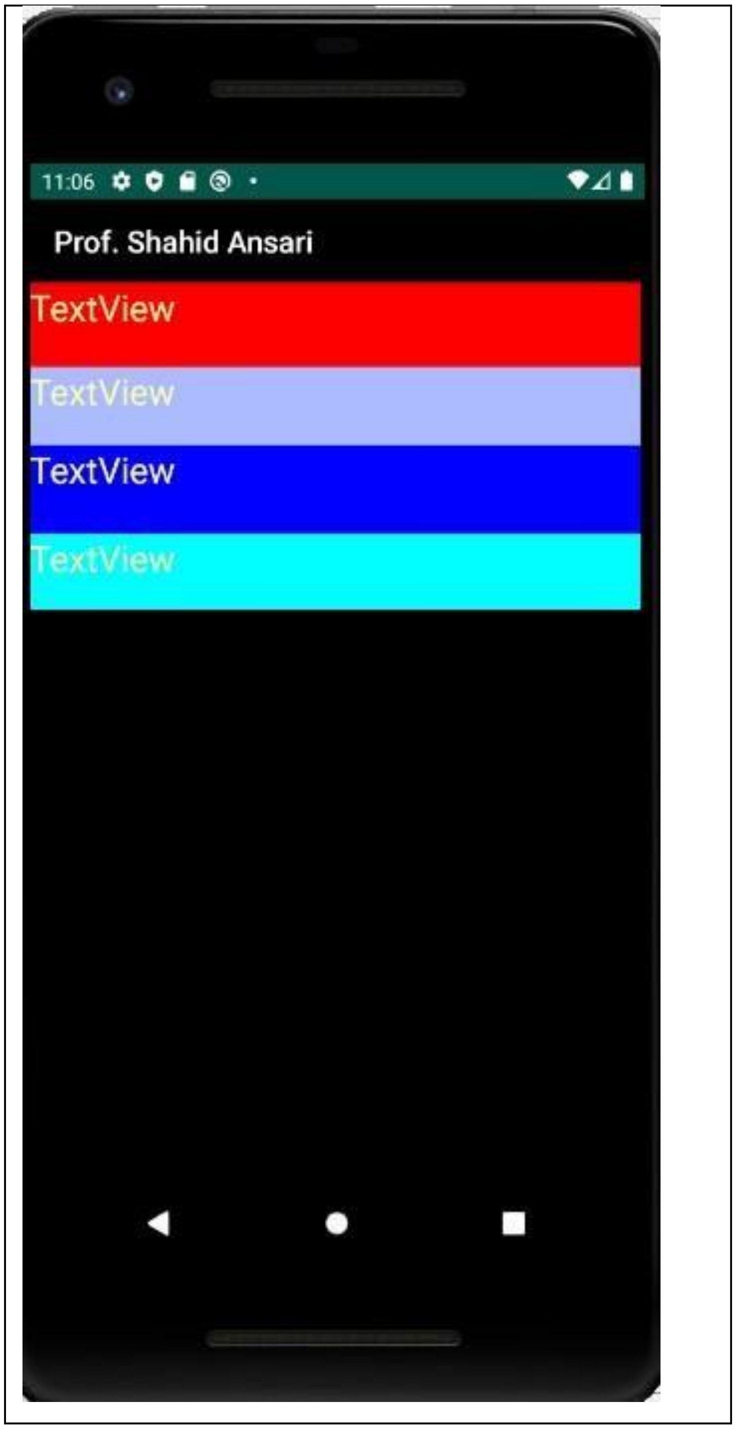
- Defining new theme properties in **styles.xml**
- Create a new project and go to:
  ProjectName>App>src>main>res>values>**styles.xml**

styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>


        <item
name="android:background">#000000</item>
        <item
name="android:textColor">#FFFFAA</item>

    </style>

</resources>
```

Add these lines in styles.xml

Output



We are done with the styles

c) Defining string property

- Defining paragraph and header property in **strings.xml**
- Create a new project and go to:
  ProjectName>App>src>main>res>values>**strings.xml**

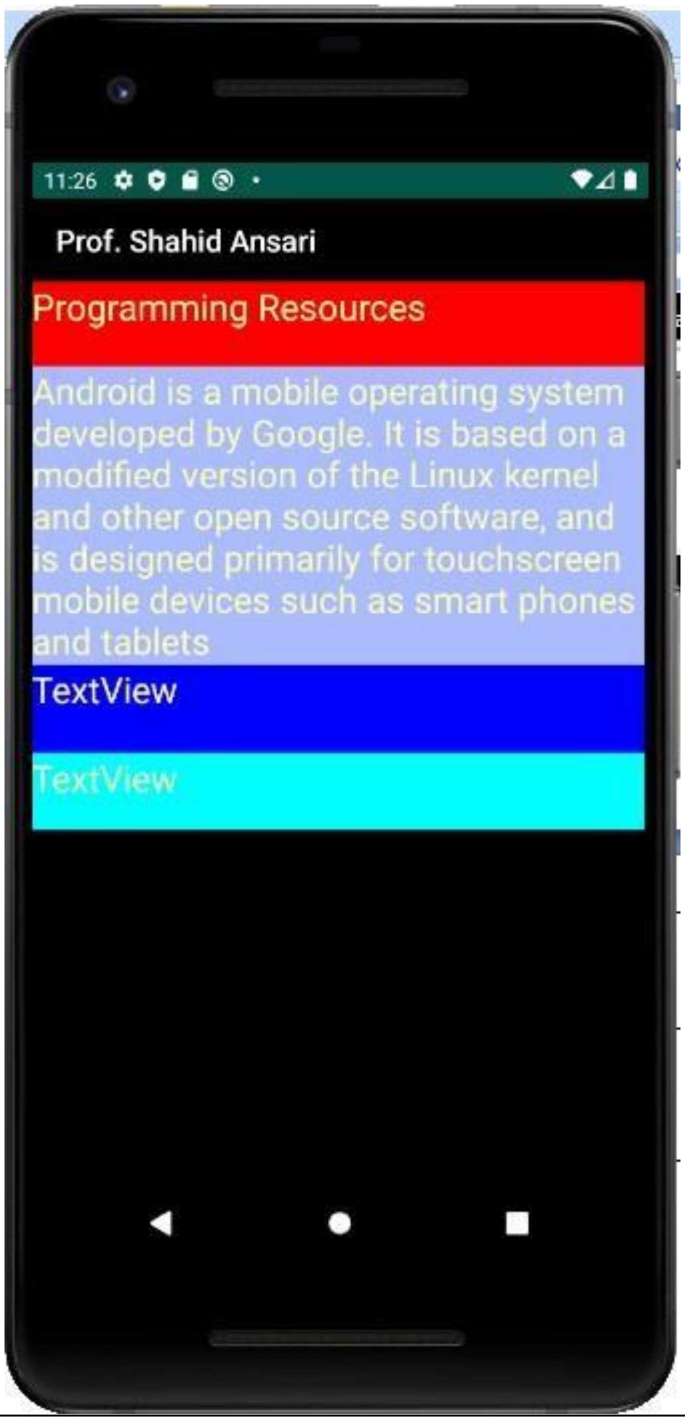
strings.xml

```
<resources>
<string name="app_name">Prof. Shahid Ansari</string>
<string name="Heading">Programming Resources</string>
<string name="Description">

            Android is a mobile operating system developed by Google.
            It is based on a modified version of the Linux
            kernel and other open source software, and is
            designed primarily for touch screen mobile devices
            such as smart phones and tablets
</string>
</resources>
```

Now go to **activity_main.xml** and type the following code

```
<TextView
   android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="56dp"
android:background="@color/Red"

                                          Change this attribute

android:text="@string/Heading"
android:textSize="24sp" />

<TextView
   android:id="@+id/textView2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/Grey"
                                              Change this attribute
   android:text="@string/Description"
android:textSize="24sp" />
```

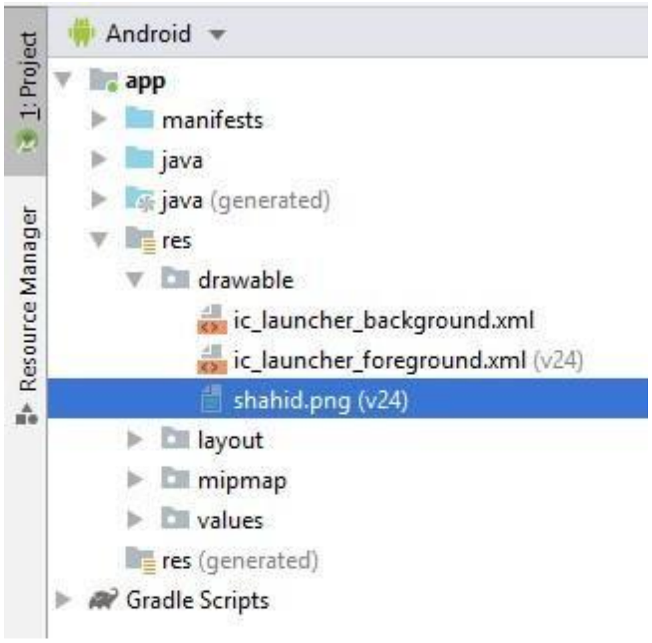
Output



```
11:26
Prof. Shahid Ansari
Programming Resources
Android is a mobile operating system
developed by Google. It is based on a
modified version of the Linux kernel
and other open source software, and
is designed primarily for touchscreen
mobile devices such as smart phones
and tablets
TextView
TextView
```

d) Adding images and dimensions

- Adding Images to Application created
- For adding a new image files, do the following:
- ProjectName>App>src>main>res>drawable>Right-click and paste the images that are copied



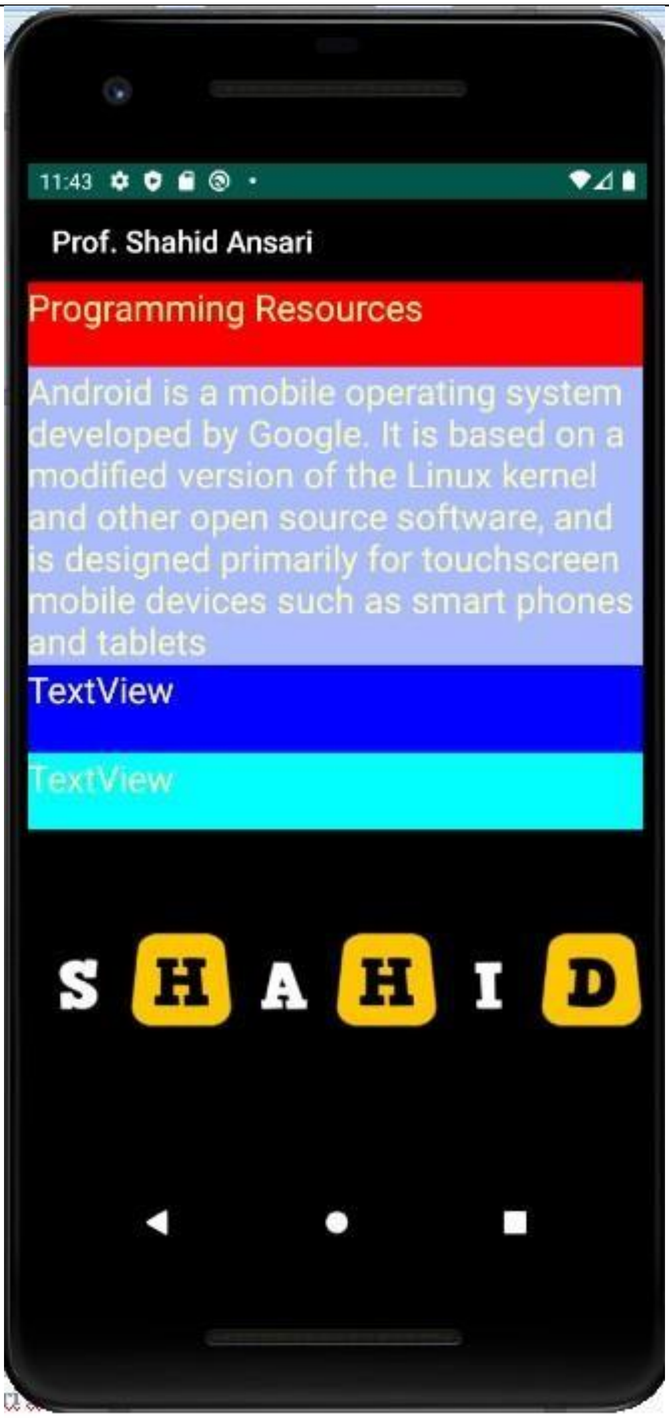
- Add a new **dim.xml** file and write the following code in dimens.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
   <dimen name="textview_height">35dp</dimen>
   <dimen name="textview_width">150dp</dimen>
   <dimen name="font_size">26sp</dimen>
</resources>
```

Now go to **activity_main.xml** and type the following code:

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

 <LinearLayout
android:layout_width="409dp"
android:layout_height="729dp"
android:orientation="vertical"
tools:layout_editor_absoluteX="1dp"
tools:layout_editor_absoluteY="1dp">



 <ImageView
android:id="@+id/imageView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:srcCompat="@drawable/shahid" />


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

Output



Programming Resources

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smart phones and tablets

TextView

TextView

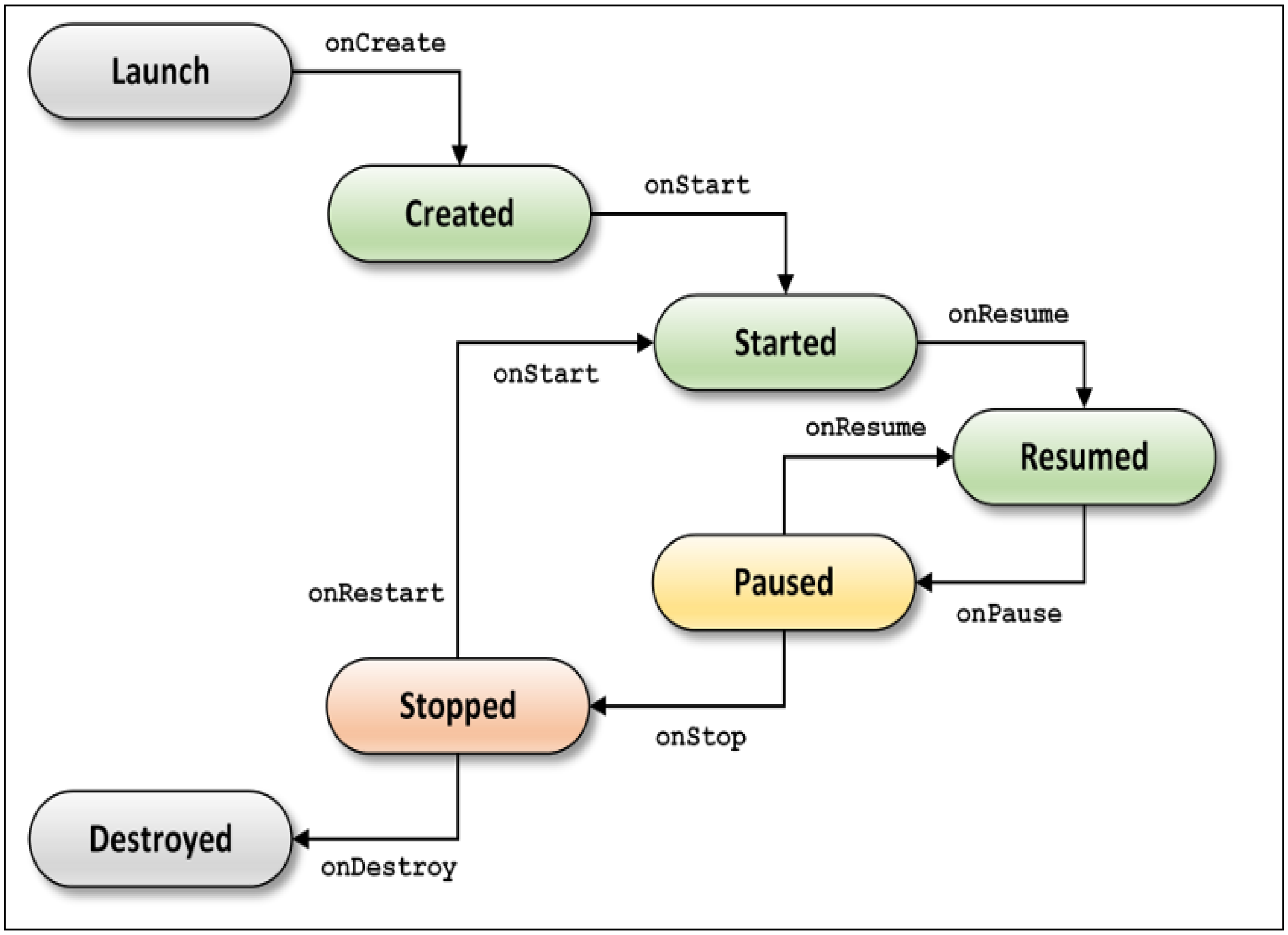
Practical 3

Programming Activities and fragments

Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments.

a) Activity Life Cycle

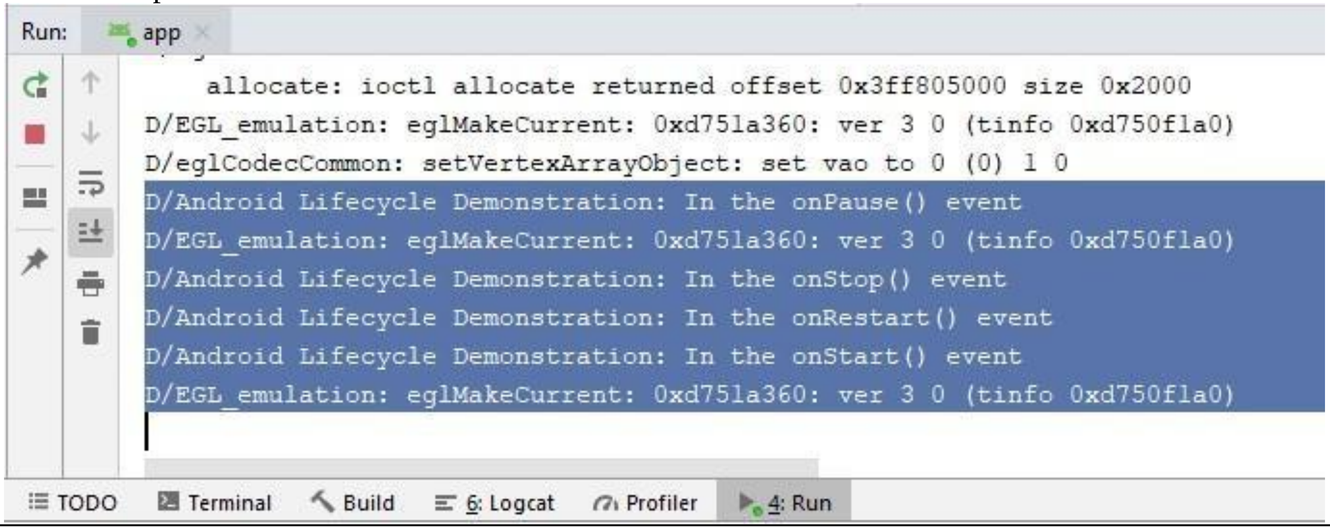
The various methods to be created are onStart(), onRestart(), onStop(), onResume(),onDestroy() and onPause() as shown in the figure.



Create a new project and go to, **MainActivity.java**

```
package MaharashtraCollege.example.profshahidansari;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
public class MainActivity extends AppCompatActivity {

   String tag= "Android Lifecycle Demonstration";
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
      Log.d(tag,"In the onCreate() event");
   }
   public void onStart()
   {
      super.onStart();
      Log.d(tag,"In the onStart() event");
   }
   public void onRestart()
   { super.onRestart();
      Log.d(tag,"In the onRestart() event");
   }
   public void onPause()
   {
      super.onPause();
      Log.d(tag,"In the onPause() event");
   }
   public void onStop()
   {
      super.onStop();
      Log.d(tag,"In the onStop() event");
   }
   public void onDestroy()
   {
      super.onDestroy();
      Log.d(tag,"In the onDestroy() event");
   }


}
```

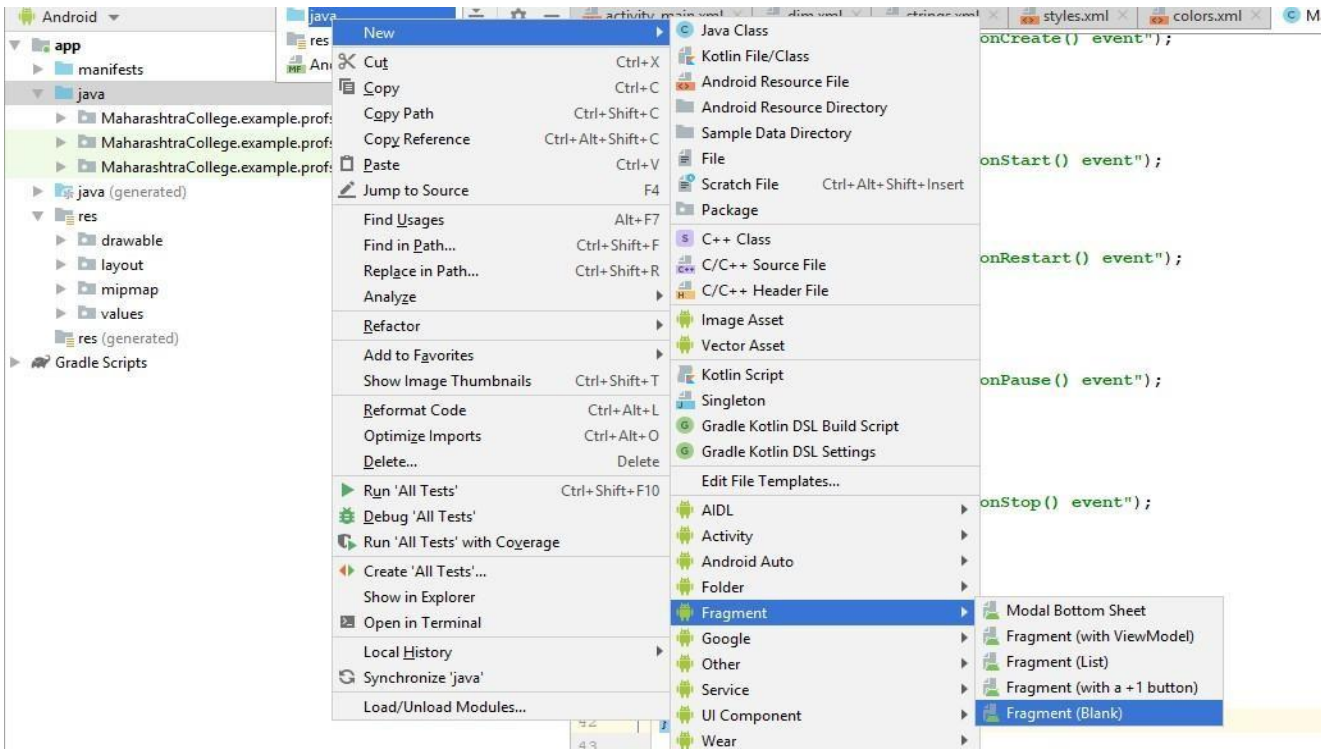
Output
See the Output in Run tab



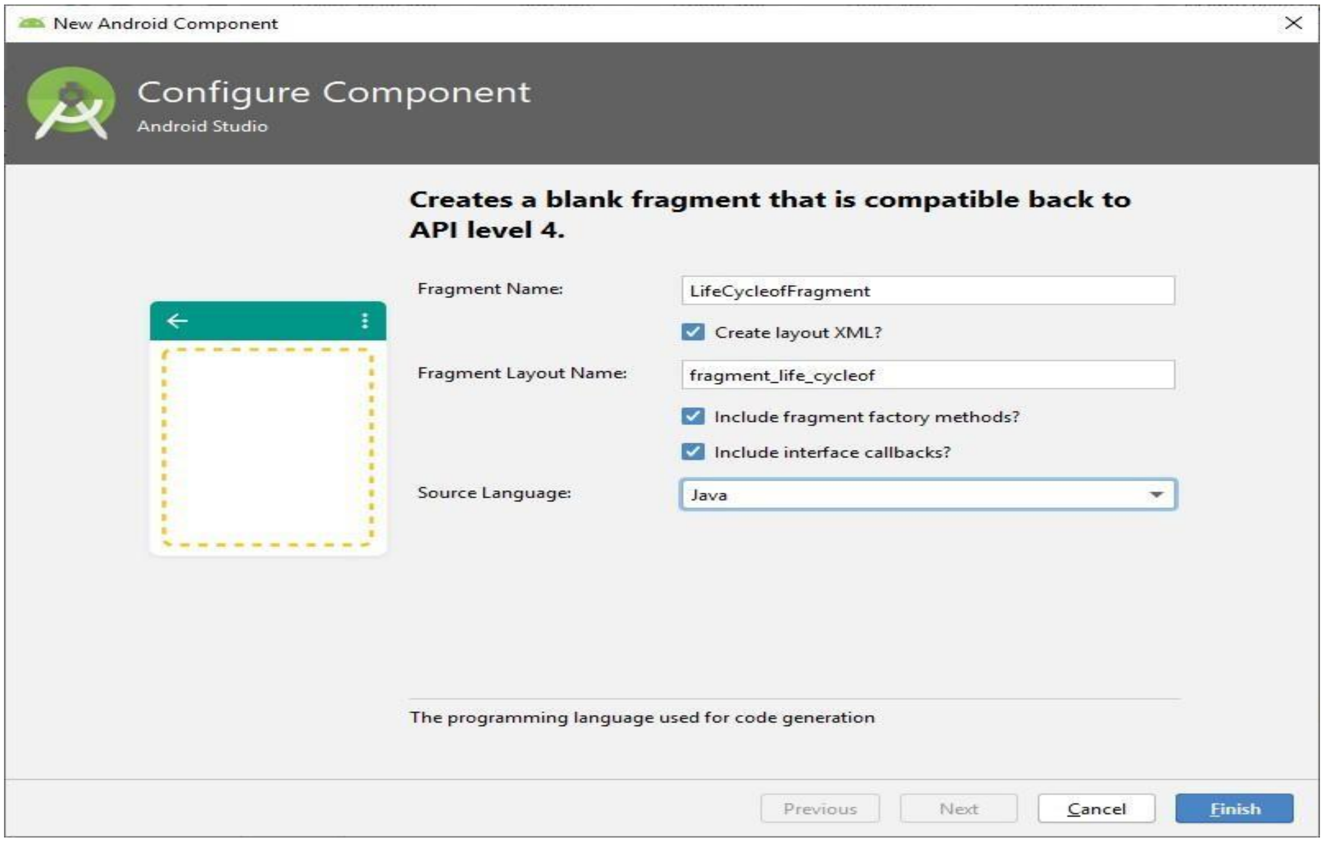
```
    allocate: ioctl allocate returned offset 0x3ff805000 size 0x2000
D/EGL_emulation: eglMakeCurrent: 0xd751a360: ver 3 0 (tinfo 0xd750f1a0)
D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 1 0
D/Android Lifecycle Demonstration: In the onPause() event
D/EGL_emulation: eglMakeCurrent: 0xd751a360: ver 3 0 (tinfo 0xd750f1a0)
D/Android Lifecycle Demonstration: In the onStop() event
D/Android Lifecycle Demonstration: In the onRestart() event
D/Android Lifecycle Demonstration: In the onStart() event
D/EGL_emulation: eglMakeCurrent: 0xd751a360: ver 3 0 (tinfo 0xd750f1a0)
```

b) Life Cycle of fragments

Create a new project
Add a new Fragment
ProjectName>App>src>main>Java
3. Right click on Java and select **‘Add Fragment’>Add Fragment(Blank)**
Give a name to the fragment and click Finish



```
onCreate() event");
onStart() event");
onRestart() event");
onPause() event");
onStop() event");
```



Creates a blank fragment that is compatible back to API level 4.

Fragment Name: LifeCycleofFragment

[✓] Create layout XML?

Fragment Layout Name: fragment_life_cycleof

[✓] Include fragment factory methods?

[✓] Include interface callbacks?

Source Language: Java

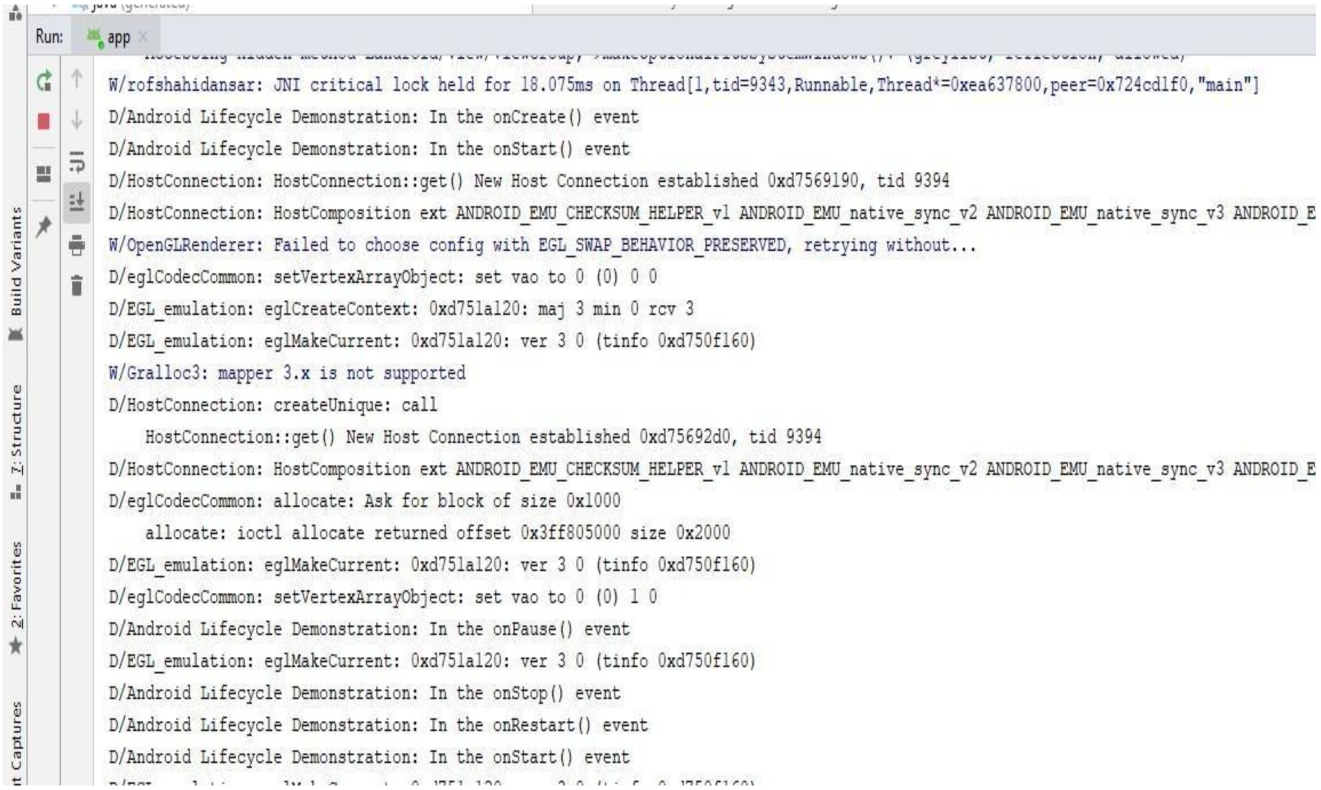
The programming language used for code generation

```
package MaharashtraCollege.example.profshahidansari;

import android.content.Context; import
android.net.Uri;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater; import
android.view.View;
import android.view.ViewGroup; import
android.util.Log;

public class LifeCycleofFragment extends Fragment
{
String tag = "Life Cycle of Fragment";
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";

    private String mParam1;
    private String mParam2;

    private OnFragmentInteractionListener mListener;

    public LifeCycleofFragment() {

    public static LifeCycleofFragment newInstance(String param1, String param2)
{
LifeCycleofFragment fragment = new LifeCycleofFragment();             Bundle args = new
Bundle();
args.putString(ARG_PARAM1, param1);        args.putString(ARG_PARAM2, param2);
fragment.setArguments(args);
return fragment;
    }

@Override
public void onCreate(Bundle savedInstanceState)  {
super.onCreate(savedInstanceState);
if (getArguments() != null)
{
mParam1 = getArguments().getString(ARG_PARAM1);
mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
// Inflate the layout for this fragment
            return  inflater.inflate(R.layout.fragment_life_cycleof,container,  false);
    }
```

```
   public void onButtonPressed(Uri uri) {                                   if(mListener != null) {
mListener.onFragmentInteraction(uri);
        }
    }

@Override
  public void onAttach(Context context) {
  super.onAttach(context);
 if (context instanceof OnFragmentInteractionListener) {                mListener = (OnFragmentInteractionListener)
context;
  } else {
      throw new RuntimeException(context.toString()
           + " must implement OnFragmentInteractionListener");
        }
    }

@Override
public           void           onDetach()           {
super.onDetach();         mListener = null;
    }

 public interface OnFragmentInteractionListener {         // TODO: Update
argument type and name
void onFragmentInteraction(Uri uri);
    }
    public void onStart()
    {
        super.onStart();
        Log.d(tag,"In the onStart() event");
    }
    public void onRestart()
    {

        Log.d(tag,"In the onRestart() event");
    }
    public void onPause()
    {
        super.onPause();
        Log.d(tag,"In the onPause() event");
    }
    public void onStop()
    {
        super.onStop();
        Log.d(tag,"In the onStop() event");      }
```

```
public void onDestroy()
{
    super.onDestroy();
    Log.d(tag,"In the onDestroy() event");
} }
```

Output



```
W/rofshahidansar: JNI critical lock held for 18.075ms on Thread[1,tid=9343,Runnable,Thread*=0xea637800,peer=0x724cd1f0,"main"]
D/Android Lifecycle Demonstration: In the onCreate() event
D/Android Lifecycle Demonstration: In the onStart() event
D/HostConnection: HostConnection::get() New Host Connection established 0xd7569190, tid 9394
D/HostConnection: HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3 ANDROID_E
W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 0 0
D/EGL_emulation: eglCreateContext: 0xd751a120: maj 3 min 0 rcv 3
D/EGL_emulation: eglMakeCurrent: 0xd751a120: ver 3 0 (tinfo 0xd750f160)
W/Gralloc3: mapper 3.x is not supported
D/HostConnection: createUnique: call
    HostConnection::get() New Host Connection established 0xd75692d0, tid 9394
D/HostConnection: HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3 ANDROID_E
D/eglCodecCommon: allocate: Ask for block of size 0x1000
    allocate: ioctl allocate returned offset 0x3ff805000 size 0x2000
D/EGL_emulation: eglMakeCurrent: 0xd751a120: ver 3 0 (tinfo 0xd750f160)
D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 1 0
D/Android Lifecycle Demonstration: In the onPause() event
D/EGL_emulation: eglMakeCurrent: 0xd751a120: ver 3 0 (tinfo 0xd750f160)
D/Android Lifecycle Demonstration: In the onStop() event
D/Android Lifecycle Demonstration: In the onRestart() event
D/Android Lifecycle Demonstration: In the onStart() event
```

Practical4

Programs related to different Layouts Linear, Relative, Table.

Linear Layout in Android

LinearLayout is a ViewGroup that is responsible for holding views in it. It is a layout that arranges its children i.e the various views and layouts linearly (one after another) in a single column(vertically) or a single row(horizontally).

Horizontal LinearLayout

In a horizontal LinearLayout, as the name suggests, the Views defined inside the Linear Layout will be arranged horizontally one after another, like in a row. By default, the orientation is set to horizontal. But its a good practice to explicitly specify the orientation of the linear layout by setting the attribute android:orientation with value horizontal in the LinearLayout tag.

Vertical Linear Layout

In a vertical LinearLayout, as the name suggests, the Views defined inside the Linear Layout are arranged verically one after another, like in a column. And for this we need to mention the android:orientation attribute with value vertical within the LinearLayout tag.

Activity_main.xml

Remember

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">


<EditText
android:id="@+id/editText"
android:layout_width="wrap_content"
android:layout_height="86dp"
android:layout_weight="1"
android:ems="10"
android:hint="Enter The Text"
android:inputType="textPersonName"
```

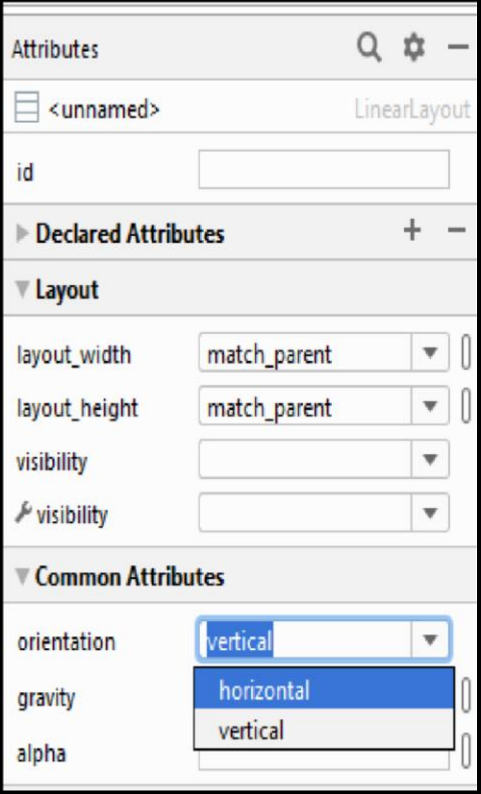
If you want horizontal layout just change orientation to horizontal

android:orientation="horizontal"

```
android:text="Type Here"        android:textColor="#FFEB3B"
/>
```

```
  <Button
     android:id="@+id/button"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_weight="1"
     android:text="Button" />

  <TextView
     android:id="@+id/textView2"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_weight="1"
     android:text="TextView" />

  <TextView
     android:id="@+id/textView3"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_weight="1"
     android:text="TextView" />

  <TextView
     android:id="@+id/textView4"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_weight="1"
     android:text="TextView" />
</LinearLayout>
```

You can also change the orientatnio using attribute window as

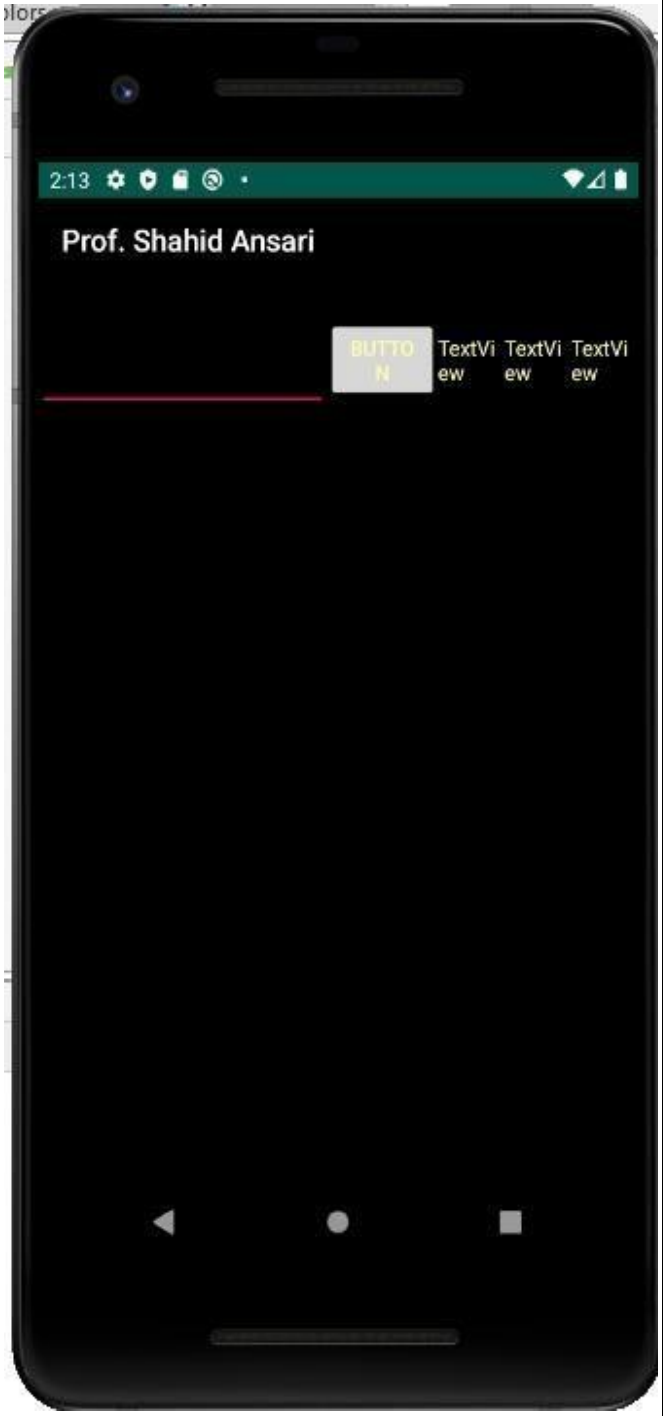
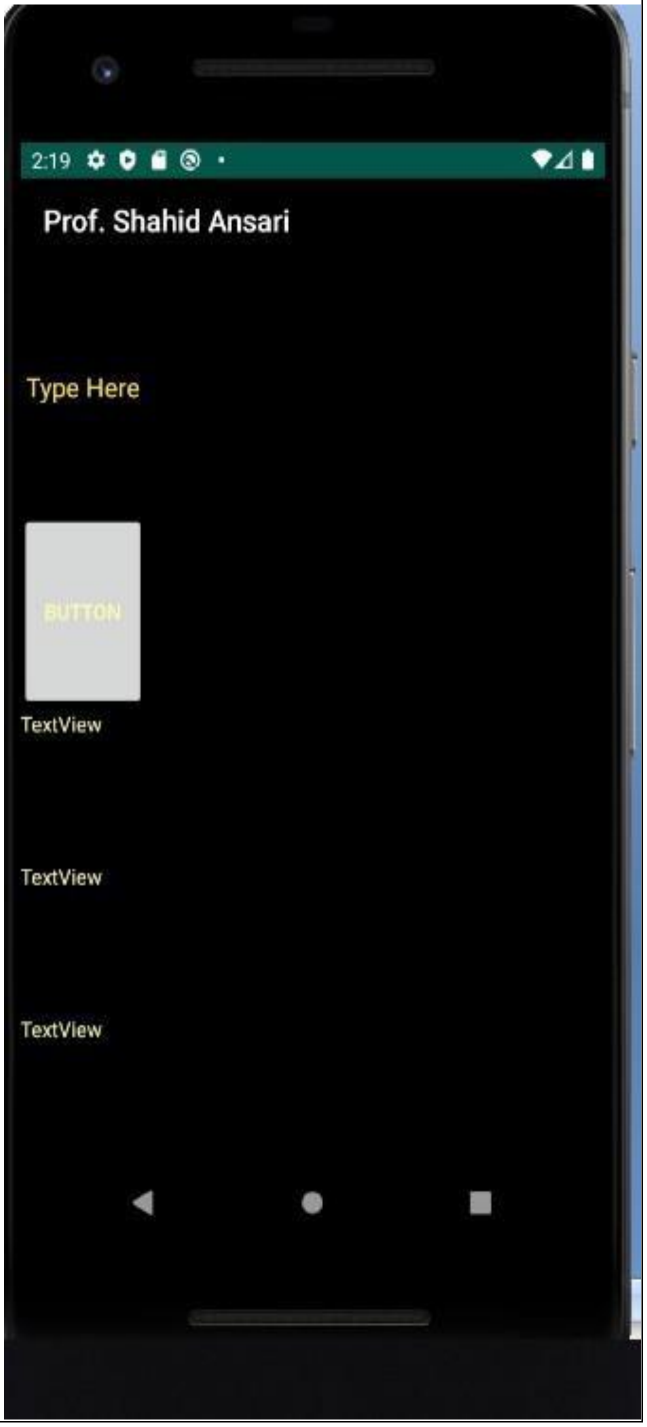


layout_width: match_parent
layout_height: match_parent
orientation: vertical

Note : Here We are getting background as black because we set the background color in style.xml in previous practical

```
<item name="android:background">#000000</item>
<item name="android:textColor">#FFFFAA</item>
```

Output:

Horizontal LinearLayout	Vertical LinearLayout
	

Relative Layout

- RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).

- A RelativeLayout is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested LinearLayout groups, you may be able to replace them with a single RelativeLayout.

Activity_main.xml

Remember

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
 <EditText
 android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter Text Here" />
 <Spinner
android:id="@+id/dates"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_below="@id/name"
android:layout_alignParentLeft="true"
android:layout_toLeftOf="@+id/times" />
  <Spinner
android:id="@id/times"
android:layout_width="96dp"
android:layout_height="wrap_content"
android:layout_below="@id/name"
android:layout_alignParentRight="true" />
   <Button
android:layout_width="96dp"
android:layout_height="wrap_content"
android:layout_below="@id/times"
```

```
android:layout_alignParentRight="true"
android:text="Click Here" />
</RelativeLayout>
```

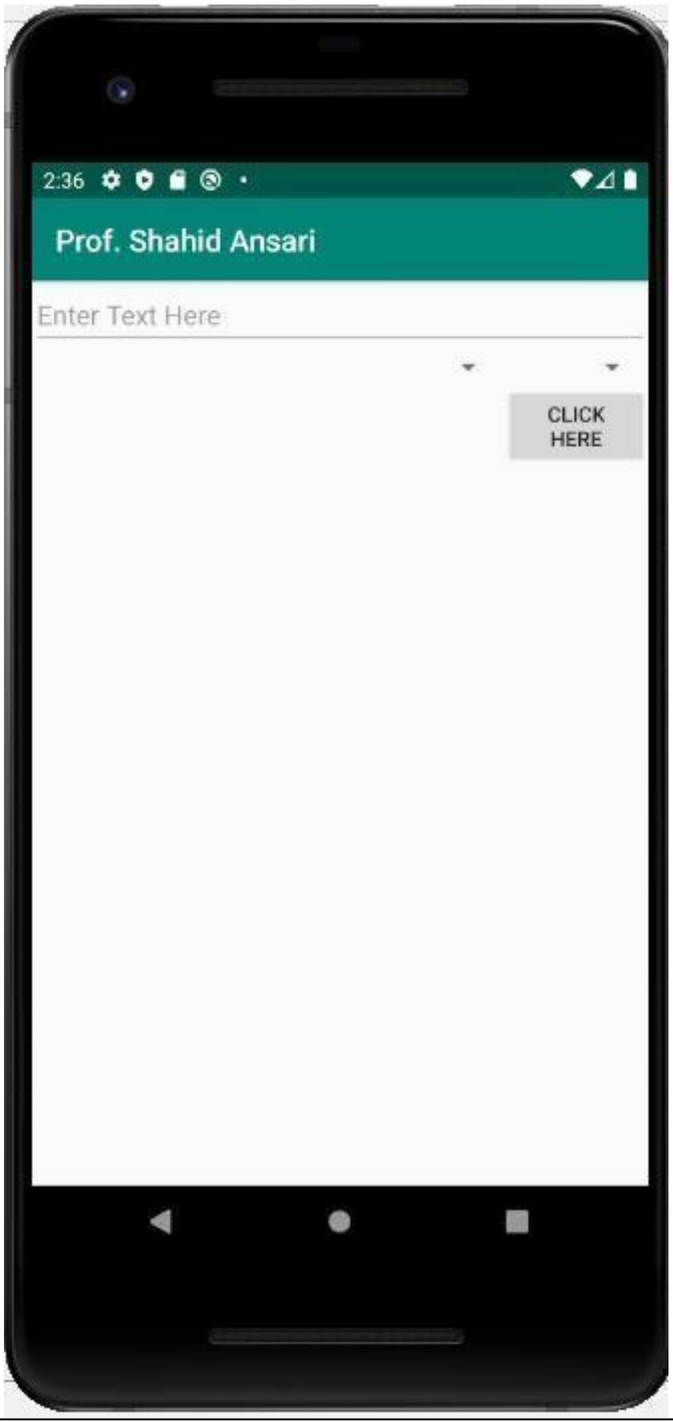


Table Layout

Android TableLayout going to be arranged groups of views into rows and columns. You will use the <TableRow> element to build a row in the table. Each row has zero or more cells; each cell can hold one View object.

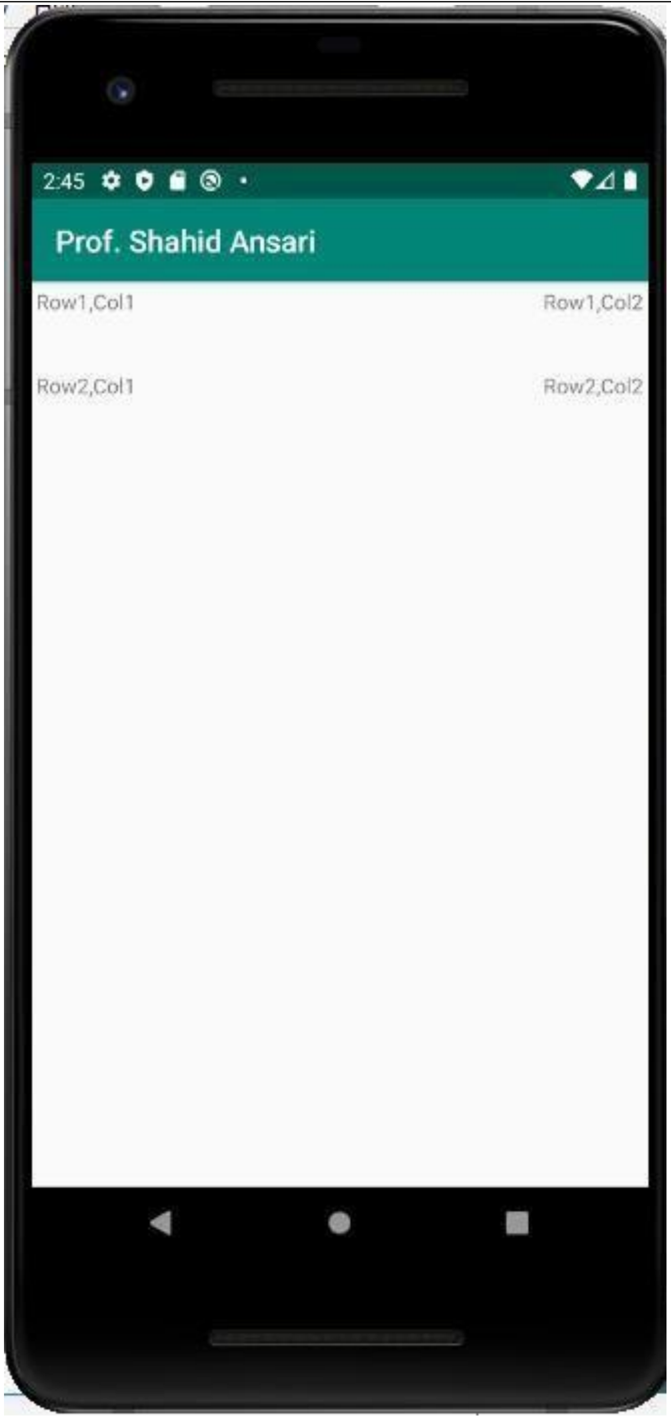
TableLayout containers do not display border lines for their rows, columns, or cells. The table will have as many columns as the row with the most cells. A table can leave cells empty. Cells can span multiple columns, as they can in HTML. You can span columns by using the span field in the TableRow.LayoutParams class.

Main_activity.xml

Remember

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:stretchColumns="1">
   <TableRow>

      <TextView
         android:layout_width="225dp"
         android:layout_height="56dp"
         android:padding="3dip"
         android:text="Row1,Col1" />

      <TextView
         android:layout_height="match_parent"
         android:gravity="right"
         android:padding="3dip"
         android:text="Row1,Col2" />
   </TableRow>

   <TableRow>

      <TextView
         android:layout_width="242dp"
         android:layout_height="64dp"
         android:padding="3dip"
         android:text="Row2,Col1" />

      <TextView
         android:layout_height="match_parent"
         android:gravity="right"
         android:padding="3dip"
         android:text="Row2,Col2" />
```

```
    </TableRow>
</TableLayout>
```



Row1,Col1 Row1,Col2

Row2,Col1 Row2,Col2

Practical 5

Programming UI elements

AppBar, Fragments, UI Components

a) <u>Demonstration of Application Bar</u>

- Create a new project
- Change the following lines in styles.xml
- To change styles.xml goto

<u>ProjectName -> App -> Src -> Main -> Res -> values-> styles.xml</u>

The default line is

```
<style name="AppTheme"
parent="Theme.AppCompat.Light.DarkActionBar">
```

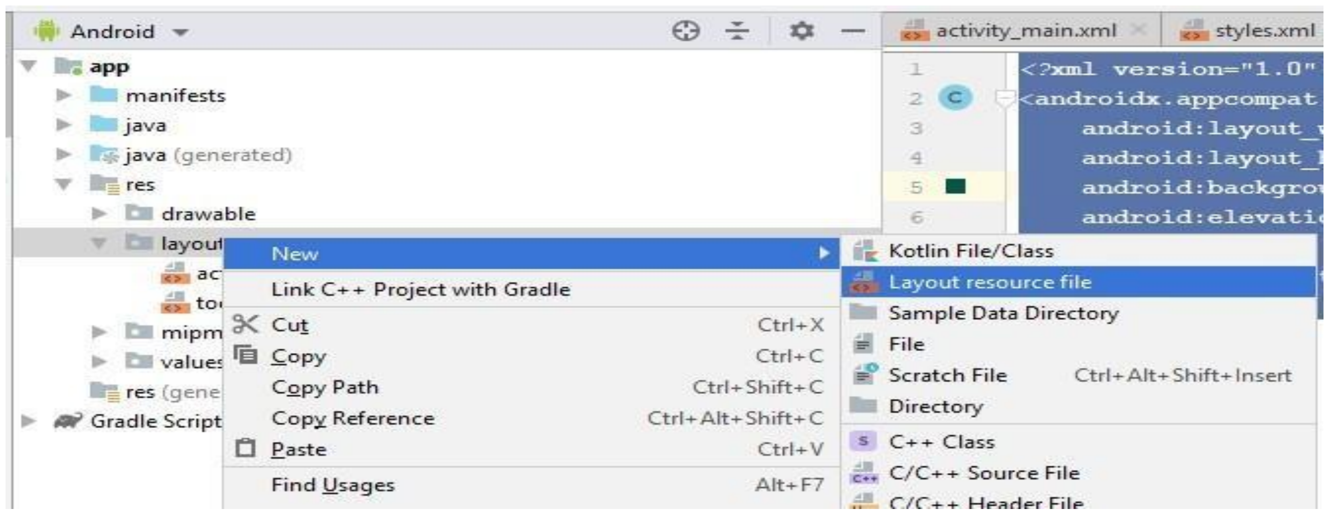
Change to

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

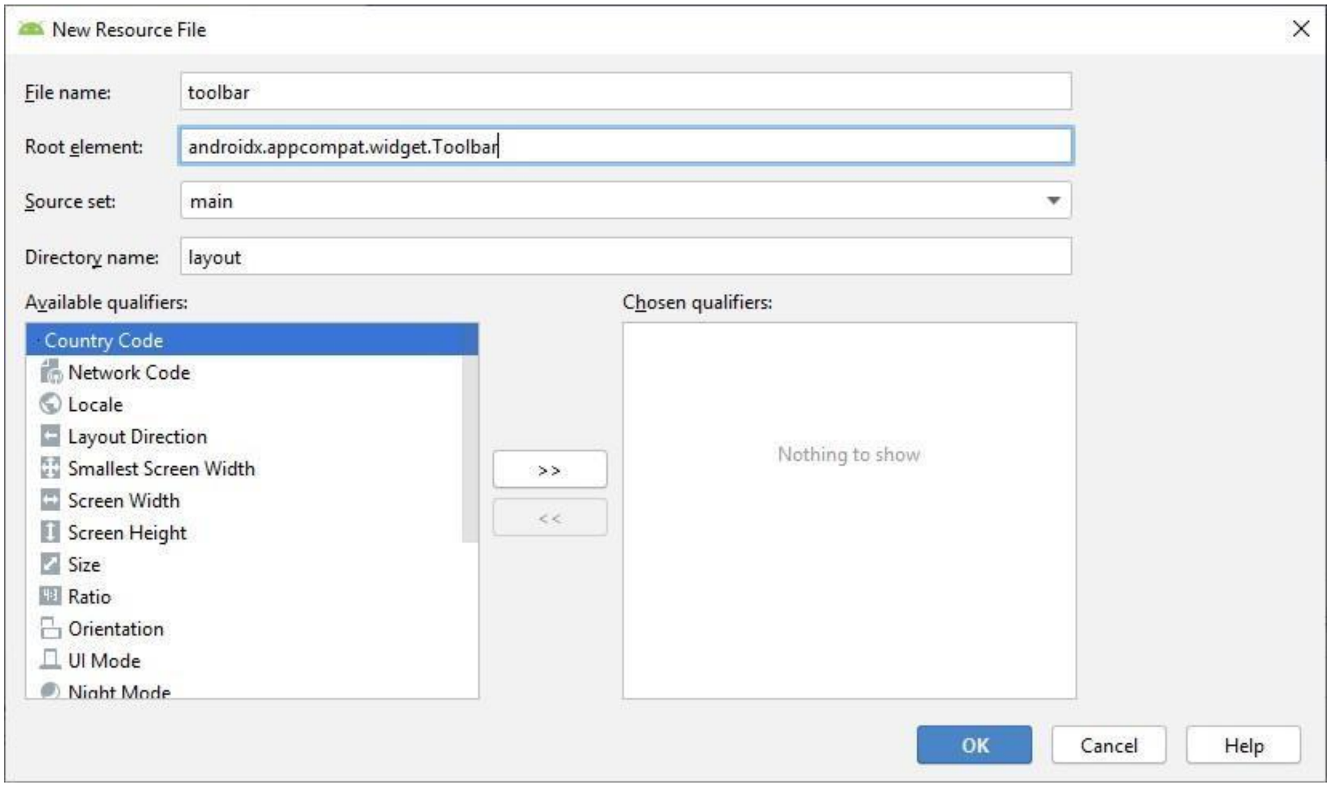
Go to, ProjectName -> App -> Src -> Main -> Res -> Layout

Right Click on layout and add a new file "**toolbar.xml**"

Go to **toolbar.xml** and Change the default layout with this line



```
<?xml version="1.0"
<androidx.appcompat
    android:layout_
    android:layout_
    android:backgro
    android:elevati
```



File name: toolbar
Root element: androidx.appcompat.widget.Toolbar
Source set: main
Directory name: layout

toolbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.Toolbar
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/colorPrimaryDark"
android:elevation="4dp"
   >
</androidx.appcompat.widget.Toolbar>
```

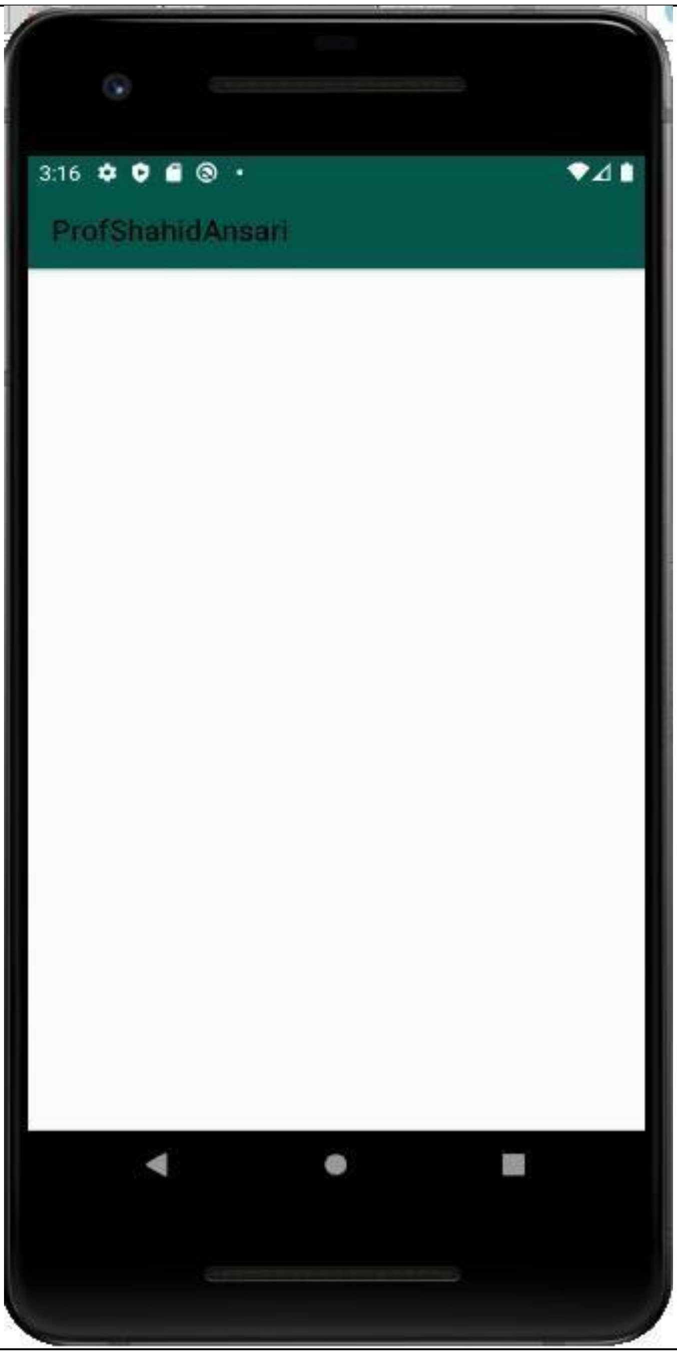
Now go to main_activity.xml and include the toolbar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!" />

<include
android:id="@+id/toolbar"
layout="@layout/toolbar"
 />

</RelativeLayout>
```

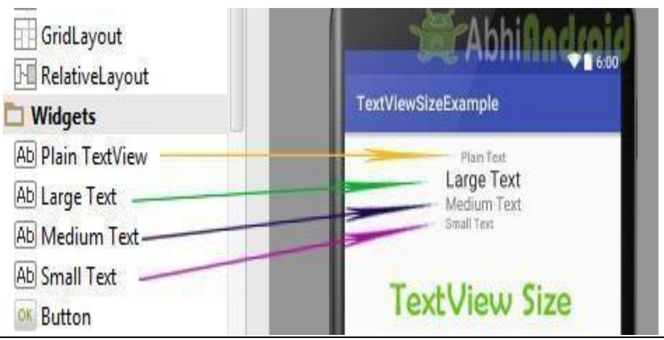
Now go to MainActivity.java and write the following code

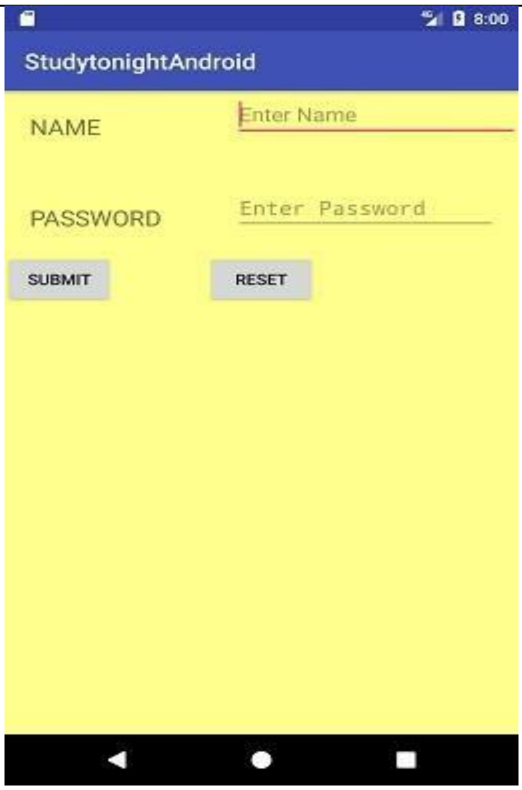
```
package com.example.profshahidansari;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import androidx.appcompat.widget.Toolbar;

public class MainActivity extends AppCompatActivity {

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

Toolbar tbar=findViewById(R.id.toolbar);
setSupportActionBar(tbar);
  }}
```

Output



b) Demonstration of UI Components(TextView,EditText,Button)

TextView :Label Field In Android, TextView displays text to the user and optionally allows them to edit it programmatically. TextView is a complete text editor, however basic class is configured to not allow editing but we can edit it.	 GridLayout RelativeLayout Widgets Plain TextView Large Text Medium Text Small Text Button AbhiAndroid 6:00 TextViewSizeExample Plain Text Large Text Medium Text Small Text TextView Size

EditText: Input Field In Android, EditText is a standard entry widget in android apps. It is an overlay over TextView that configures it self to be editable. EditText is a subclass of TextView with text editing operations. We often use EditText in our applications in order to provide an input or text field, especially in forms. The most simple example of EditText is Login or Sign-in form.	 8:00 StudytonightAndroid NAME Enter Name PASSWORD Enter Password SUBMIT RESET

Button In Android, Button represents a push button. A Push buttons can be clicked, or pressed by the user to perform an action. There are different types of buttons used in android such as CompoundButton, ToggleButton, RadioButton.	

Calculator Application

Activity_main.xml

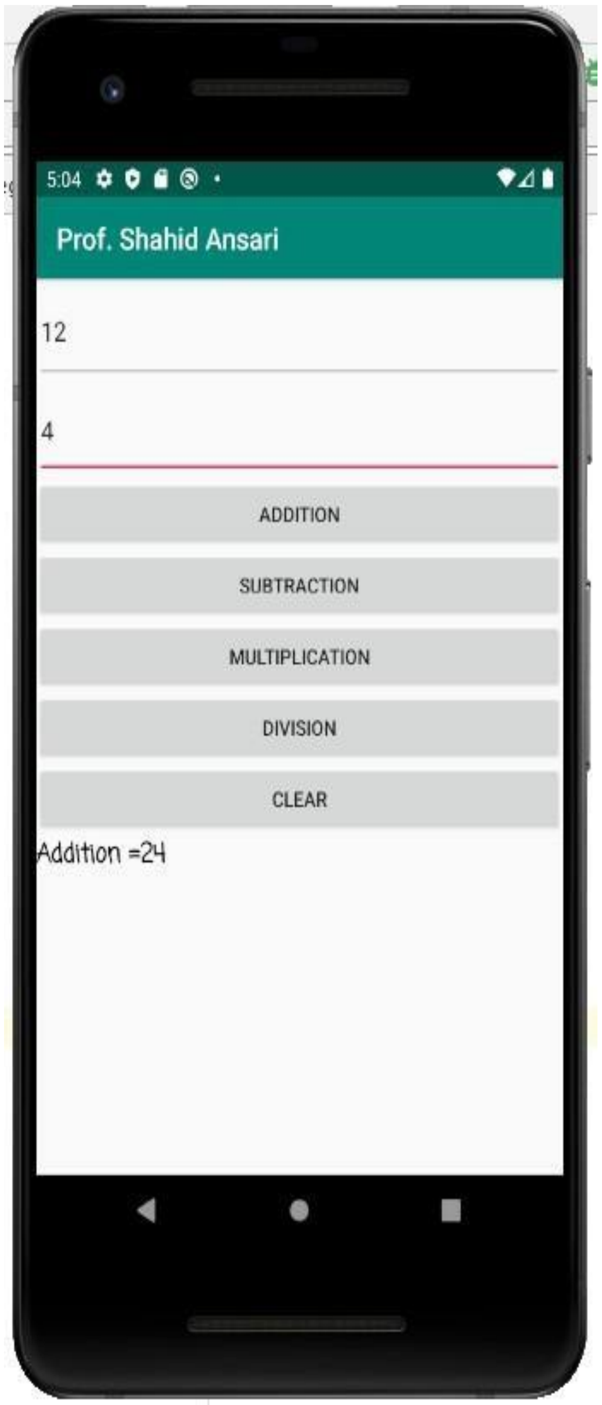
```
 <?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal"
android:stretchColumns="1">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

<EditText
android:id="@+id/et1"
android:layout_width="match_parent"
android:layout_height="70dp"
android:ems="10"
android:inputType="textPersonName"
android:text="Input1" />

<EditText
android:id="@+id/et2"
android:layout_width="match_parent"
android:layout_height="64dp"
android:ems="10"
android:inputType="textPersonName"
```

```
android:text="Input2" />

<Button
android:id="@+id/btnAdd"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Addition" />

<Button
android:id="@+id/btnSub"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Subtraction" />

<Button
android:id="@+id/btnMult"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Multiplication" />

<Button
android:id="@+id/btnDiv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Division" />

<Button
android:id="@+id/btnClear"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Clear" />

<TextView
android:id="@+id/tv1"
android:layout_width="match_parent"
android:layout_height="63dp"
android:text="Output"
android:textColor="@android:color/background_dark"
android:textSize="18sp"
android:textStyle="bold"
app:fontFamily="casual" />
   </LinearLayout>
</LinearLayout>
```

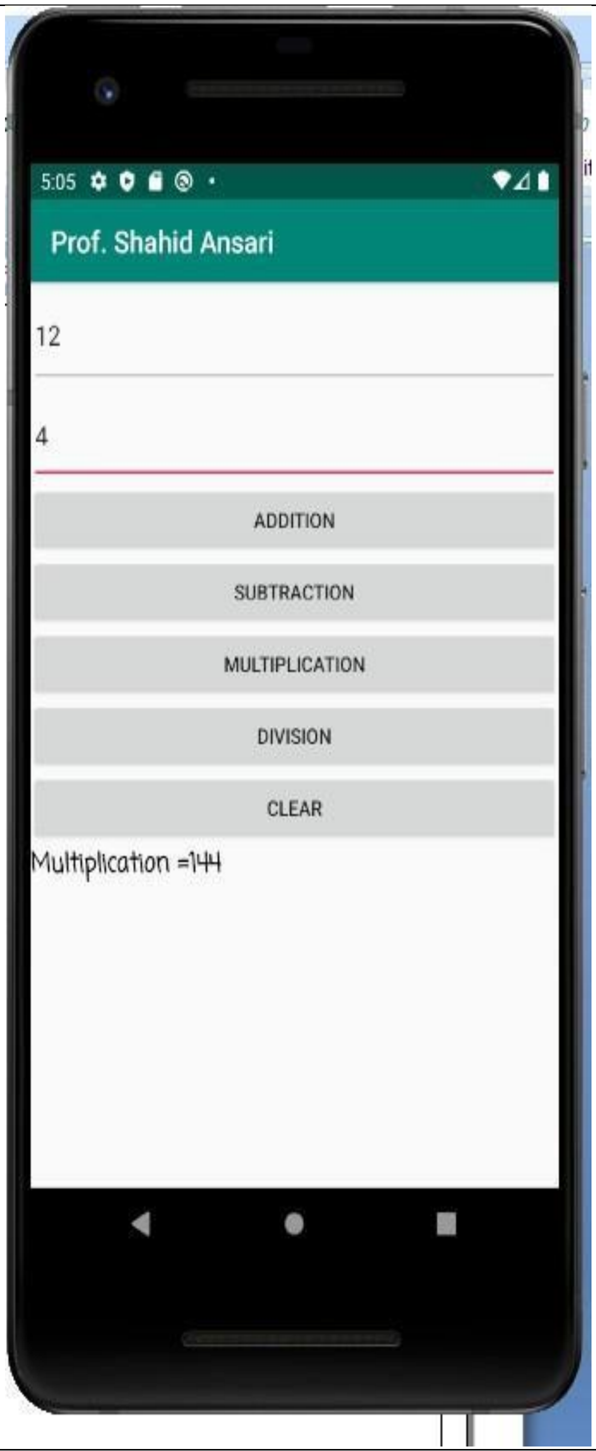
Main_Activity.java

```
package MaharashtraCollege.example.profshahidansari;

importandroidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import static android.view.View.*;

public class MainActivity extends AppCompatActivity  {

EditText t1,t2;
Button    b1,b2,b3,b4,b5;
TextViewtv1;
int n1=0,n2=0;
String s1,s2;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

t1 = (EditText) findViewById(R.id.et1);
t2 = (EditText) findViewById(R.id.et2);

b1 = (Button) findViewById(R.id.btnAdd);
b2 = (Button) findViewById(R.id.btnSub);
b3 = (Button) findViewById(R.id.btnMult);
b4 = (Button) findViewById(R.id.btnDiv);
b5 = (Button) findViewById(R.id.btnClear);

tv1 = (TextView) findViewById(R.id.tv1);




b1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
```

```
try {
String s1 = t1.getText().toString();
String s2 = t2.getText().toString();
n1 = Integer.parseInt(s1);
n2 = Integer.parseInt(s1);
int     sum     =     n1     +     n2;
tv1.setText("Addition ="+sum);
}
catch (NumberFormatException e)
{

}

}
});

b2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

try {
String s1 = t1.getText().toString();
String s2 = t2.getText().toString();
n1 = Integer.parseInt(s1);
n2 = Integer.parseInt(s1);
int     sub     =     n1     -     n2;
tv1.setText("Subtraction ="+sub);
}
catch (NumberFormatException e)
{

}

}
});
b3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

try {
String s1 = t1.getText().toString();
String s2 = t2.getText().toString();
n1 = Integer.parseInt(s1);
n2 = Integer.parseInt(s1);
int     m     =     n1     *     n2;
tv1.setText("Multiplication ="+m);
}
catch (NumberFormatException e)
{

}

}
});

b4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

try {
String s1 = t1.getText().toString();
String s2 = t2.getText().toString();
n1 = Integer.parseInt(s1);
n2 = Integer.parseInt(s1);
int d = n1 / n2;
tv1.setText("Division ="+d);
}
catch (NumberFormatException e)
{

}

}
});


b5.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {

t1.setText(" ");
t2.setText(" ");
tv1.setText(" ");

            }
        });
}
}
```



Addition =24



Multiplication =144

Practical 6

Question :
Create an android application for the following menu items ,the appropriate toast should appear by clicking on the item :
• Settings
• Search
• Compose Email
• FeedBack
(make Compose Email item disabled)

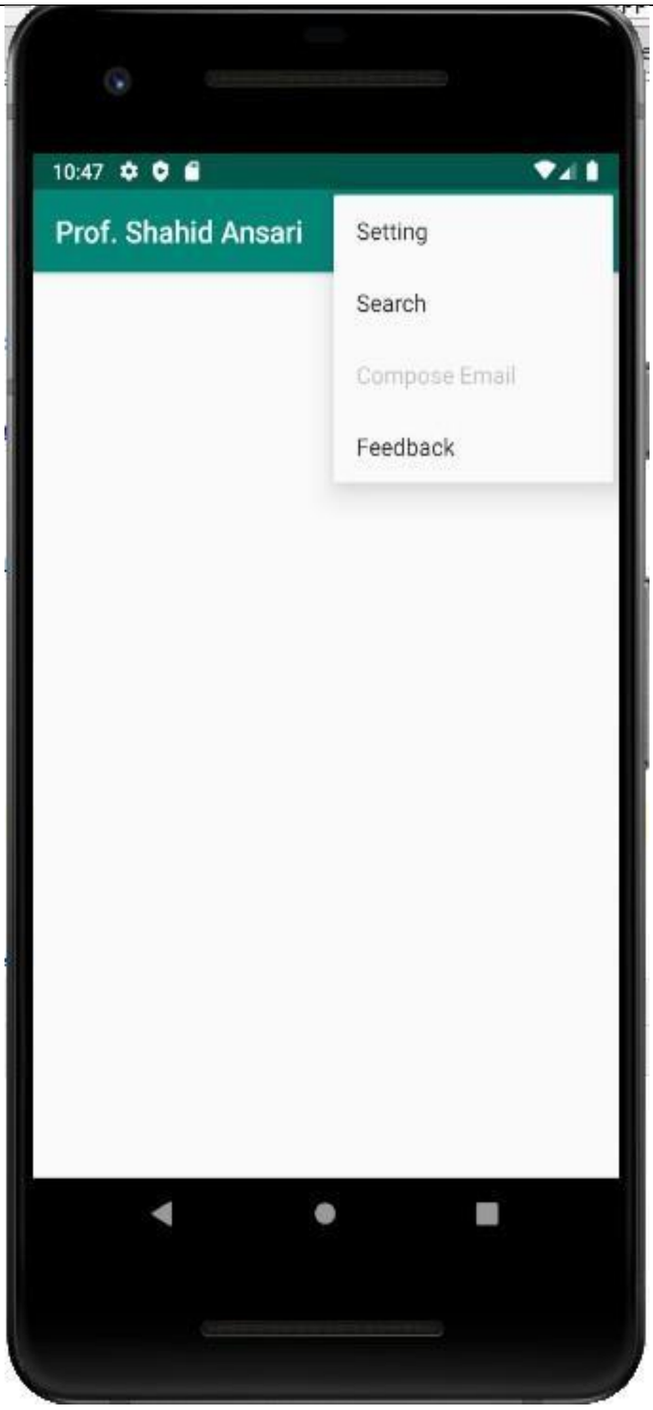
mymenus.xml

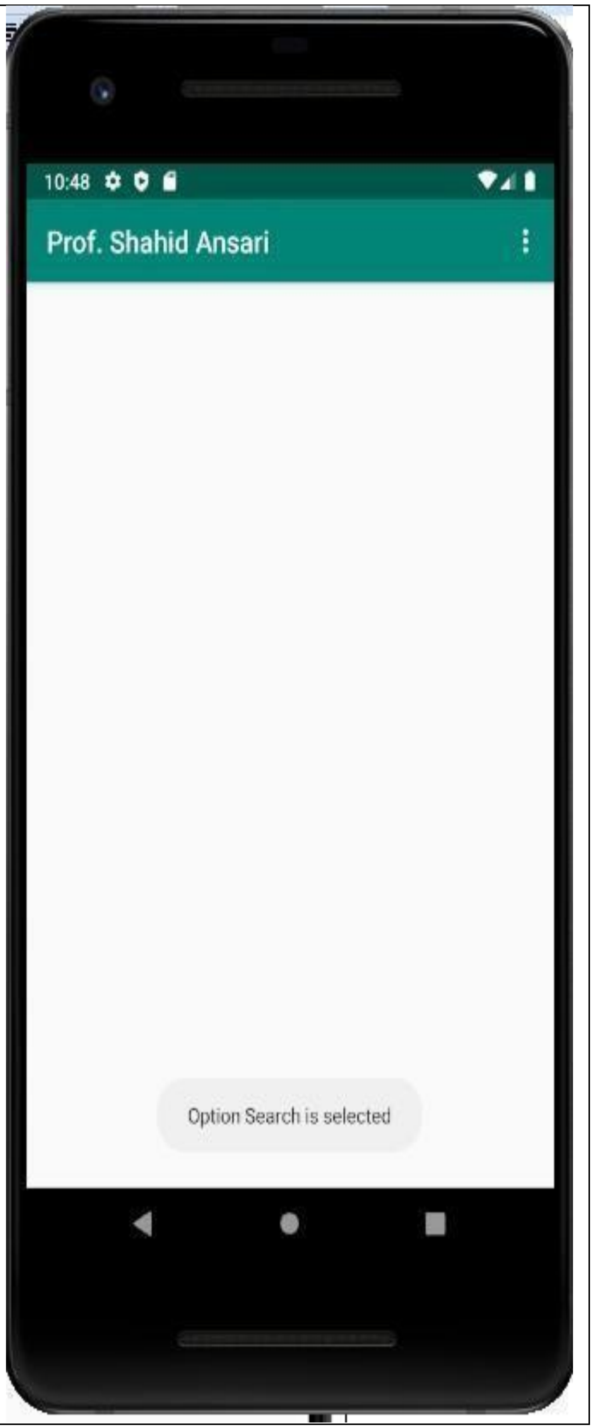
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/settings" android:title="Setting"></item>
    <item android:id="@+id/Search" android:title="Search"></item>
    <item android:id="@+id/CEmail" android:title="Compose Email"
android:enabled="false"></item>
    <item android:id="@+id/Feedback" android:title="Feedback"></item>

</menu>
```

To disable Compose M l

Main_Activity.java

```
package com.maharashtracollege.profshahidansari;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity  {

    MenuInflater mi;

@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);}



public boolean onCreateOptionsMenu(Menu menus)
     {
        mi = getMenuInflater();
        mi.inflate(R.menu.mymenus,menus);

        return true;
    }


  public boolean onOptionsItemSelected(MenuItem  item)
    {



        switch(item.getItemId())
        {
           case R.id.settings:
               Toast.makeText(this,"Option Setting is selected",Toast.LENGTH_SHORT).show();
               return true;
           case R.id.Search:
               Toast.makeText(this,"Option Search is selected",Toast.LENGTH_SHORT).show();
               return true;
           case R.id.Feedback:
               Toast.makeText(this,"Option Feedback is
selected",Toast.LENGTH_SHORT).show();
               return true;

           case R.id.CEmail:


               Toast.makeText(this,"Option Compose Mail is
selected",Toast.LENGTH_SHORT).show();
               return true;

               default:
                  Toast.makeText(this,"Default",Toast.LENGTH_SHORT).show();
                  return super.onOptionsItemSelected(item);


        }
    }
```





Option Search is selected

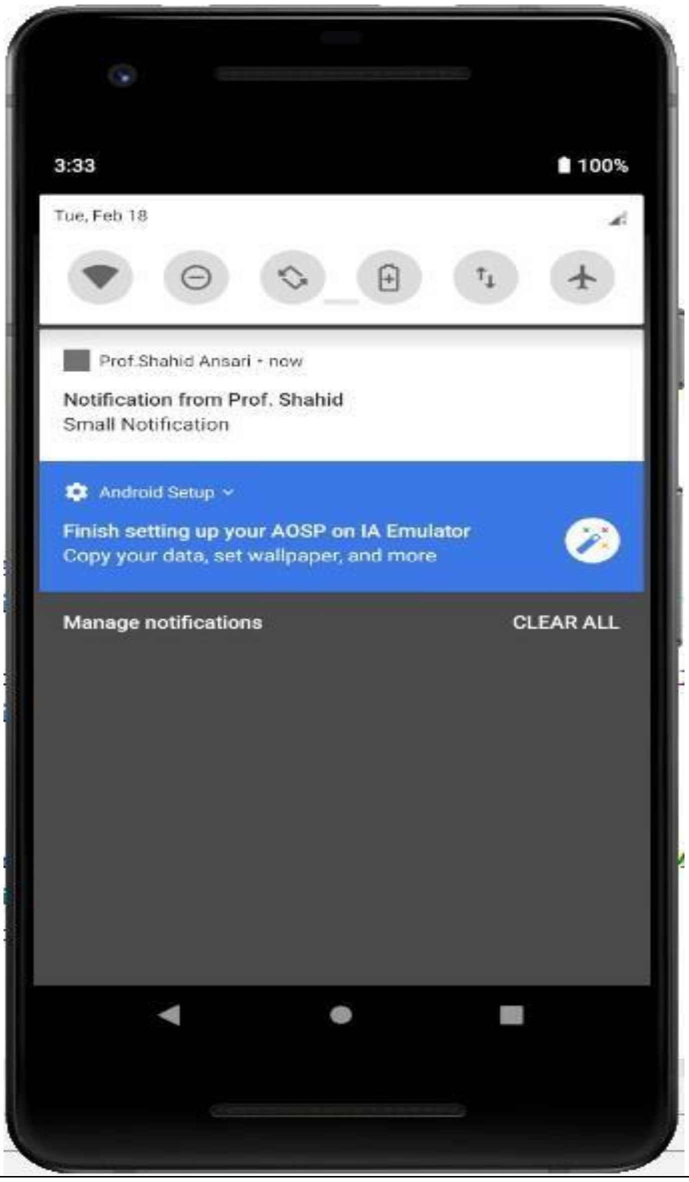
Practical 7

Question:
Create an android application to generate notification

Main_Activity.java

```
package com.maharashtracollege.profshahidansari;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.NotificationChannel;
import
android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import
android.widget.Button;

import java.nio.channels.Channel;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.O)
        {

            String description="This is description";
            int importance= NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel notificationChannel=new NotificationChannel("My
Notification","Simple Notification",importance);
            notificationChannel.setDescription(description);
             NotificationManager
notificationManager=(NotificationManager)getSystemService(NOTIFICATION_SERVICE);
            notificationManager.createNotificationChannel(notificationChannel);
        }
        NotificationCompat.Builder builder=new NotificationCompat.Builder(this,"My Notification");
        builder.setContentTitle("Notification from Prof. Shahid");
        builder.setSmallIcon(R.drawable.ic_launcher_background);
        builder.setAutoCancel(true);
        builder.setContentText("Small Notification");



        NotificationManagerCompat manager = NotificationManagerCompat.from(this);
        manager.notify(999,builder.build());
    }
}
```

Output



```
3:33                                   100%
Tue, Feb 18
Prof.Shahid Ansari • now
Notification from Prof. Shahid
Small Notification
Android Setup
Finish setting up your AOSP on IA Emulator
Copy your data, set wallpaper, and more
Manage notifications              CLEAR ALL
```

Practical 8

Question:
Create an android application that opens the website www.google.com in the browser on the click of a button using intents

Activity_main.xml

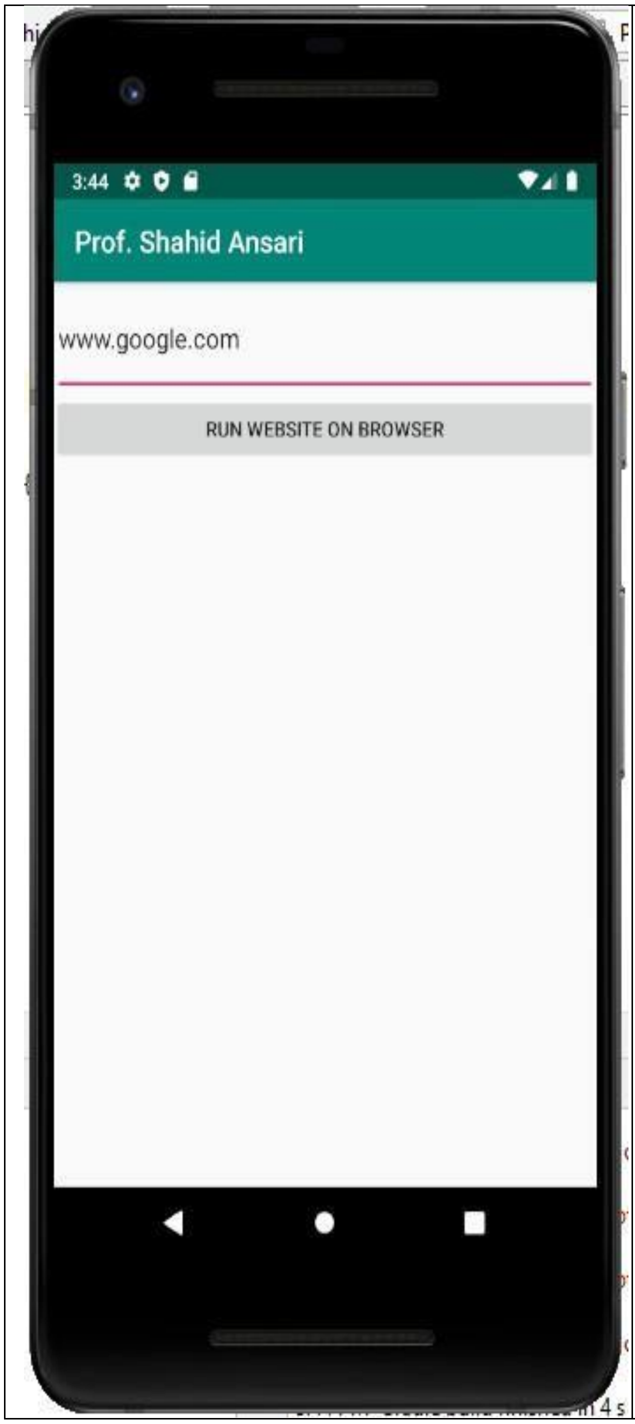
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical">


      <EditText
         android:hint="Type URL Here"
         android:id="@+id/website"
         android:layout_width="fill_parent"
         android:layout_height="75dp"
         android:ems="5"></EditText>
      <Button
         android:id="@+id/runWebsite"
         android:layout_width="fill_parent"
         android:layout_height="45dp"
         android:text="Run WebSite on Browser" />
   </LinearLayout>
</RelativeLayout>
```

Main_Activity.java

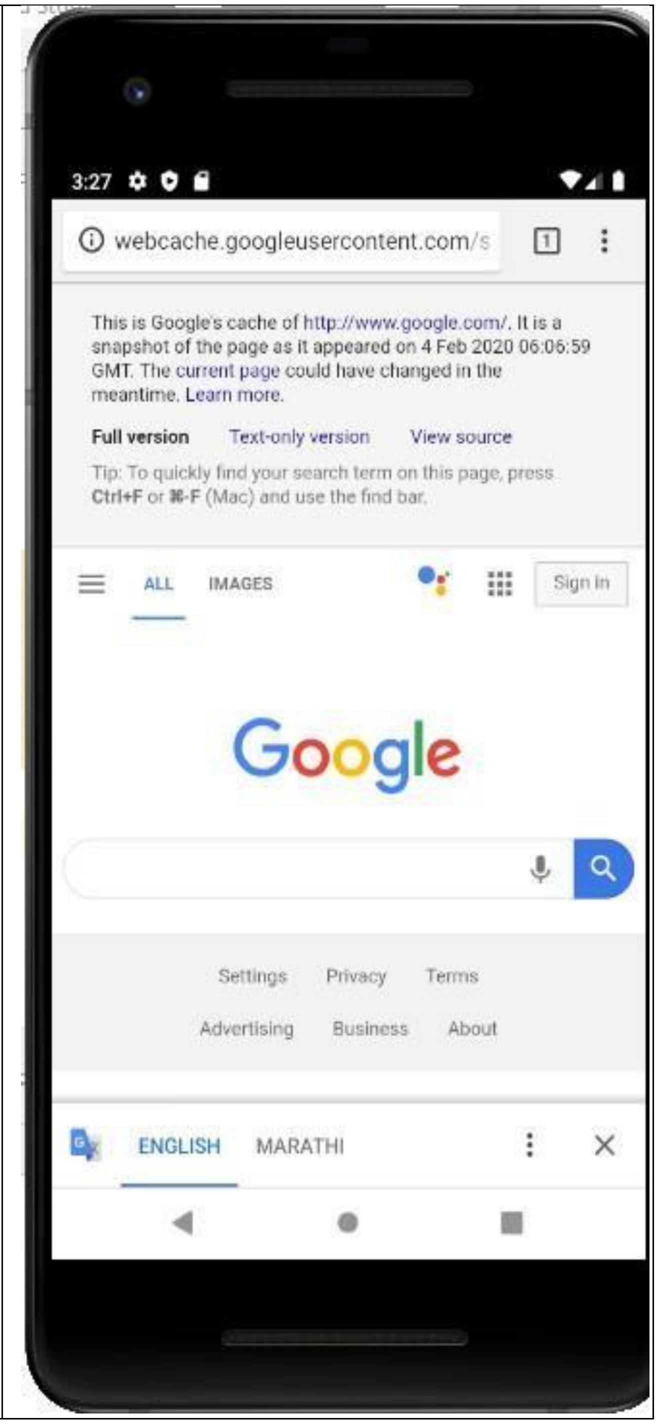
```
package com.maharashtracollege.profshahidansari;
import
androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;

import android.app.Activity;
import
android.content.Intent;
```

```
import android.net.Uri;
import android.os.Bundle;
```

```
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import
android.widget.EditText;

public class MainActivity extends AppCompatActivity  {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        Button runWebsite = (Button) findViewById(R.id.runWebsite);
        runWebsite.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                EditText website = (EditText) findViewById(R.id.website);
                String strURL = website.getText().toString();
                if (strURL.indexOf("http://www") < 0) {
                    strURL = "http://www." + strURL;
                }
                Intent implicit = new Intent(Intent.ACTION_VIEW, Uri.parse(strURL));
                startActivity(implicit);
            }
        });

        }
}
```



www.google.com



This is Google's cache of http://www.google.com/. It is a snapshot of the page as it appeared on 4 Feb 2020 06:06:59 GMT. The current page could have changed in the meantime. Learn more.

Full version Text-only version View source

Tip: To quickly find your search term on this page, press **Ctrl+F** or **⌘-F** (Mac) and use the find bar.

Practical 9

Question:
Create an android application that displays a login form with text for username and password and button for submit and reset. On submitting, toast should be displayed accordingly i.e. “Correct username and password” if username and password match and “Incorrect username/password” if username and password do not match.

activity_main.xml

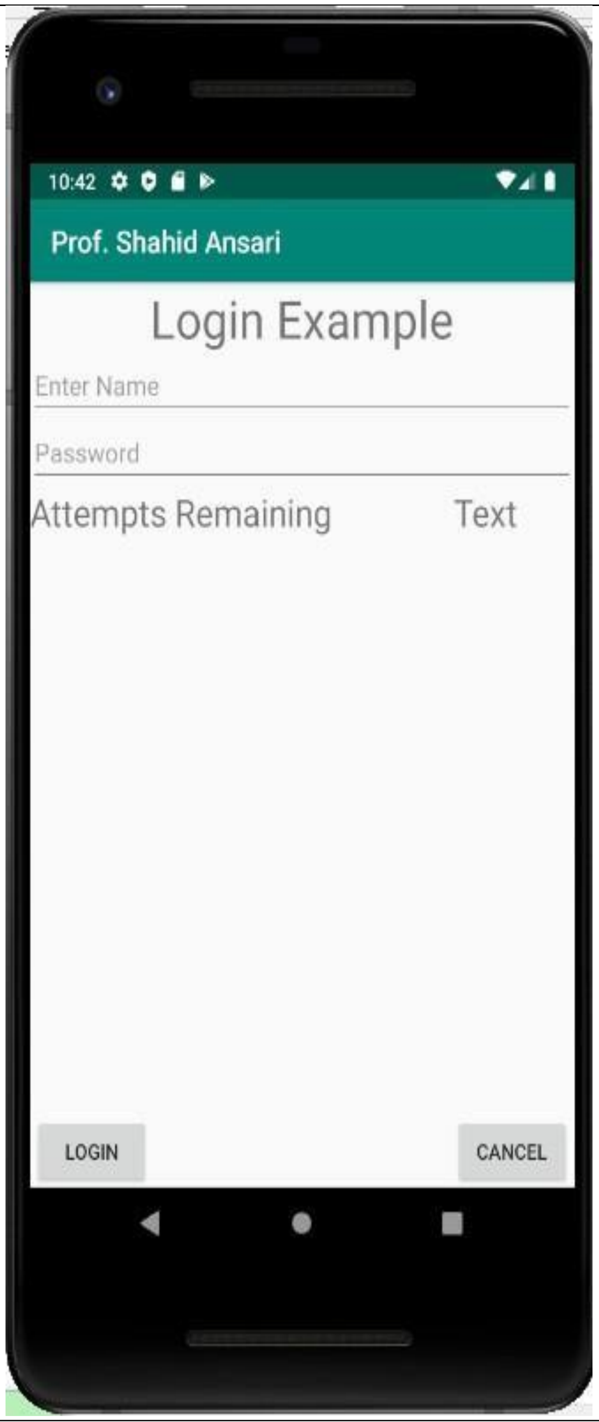
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

TextView android:id="@+id/textview"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_alignParentTop="true"
 android:layout_centerHorizontal="true"
 android:text="Login Example"
 android:textSize="35dp" />

    <EditText
 android:layout_width = "wrap_content"
 android:layout_height = "wrap_content"
 android:id = "@+id/editText" android:hint =
 "Enter Name" android:focusable = "true"
 android:textColorHighlight = "#ff7eff15"

 android:layout_marginTop = "46dp"

 android:layout_alignParentLeft = "true" android:layout_alignParentStart =
 "true" android:layout_alignParentRight = "true"
 android:layout_alignParentEnd = "true" />

EditText android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:inputType="textPassword"
 android:ems="10"
```

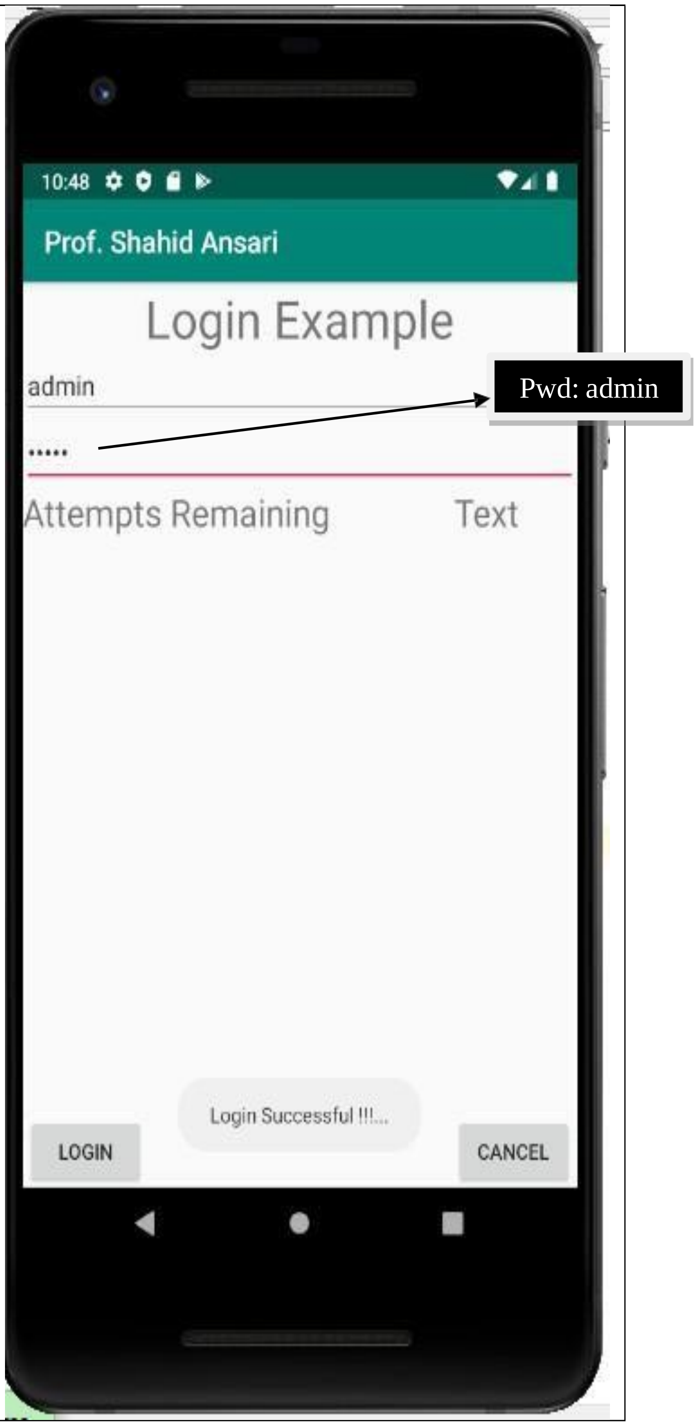
```
        android:id="@+id/editText2"
        android:layout_below="@+id/editText"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignRight="@+id/editText"
        android:layout_alignEnd="@+id/editText"

        android:hint="Password" />

    <TextView  android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText2"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:text="Attempts Remaining"
        android:textSize="25dp" />

    <TextView  android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/textView2"
        android:layout_alignBottom="@+id/textView2"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_toEndOf="@+id/textview"
        android:layout_toRightOf="@+id/textview"
        android:text="Text"
        android:textSize="25dp" />

    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="login" android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/textview"
        android:layout_toStartOf="@+id/textview" />


  <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel" android:id="@+id/button2"
        android:layout_alignParentBottom="true"
        android:layout_toRightOf="@+id/textview"
        android:layout_toEndOf="@+id/textview" />

</RelativeLayout>
```

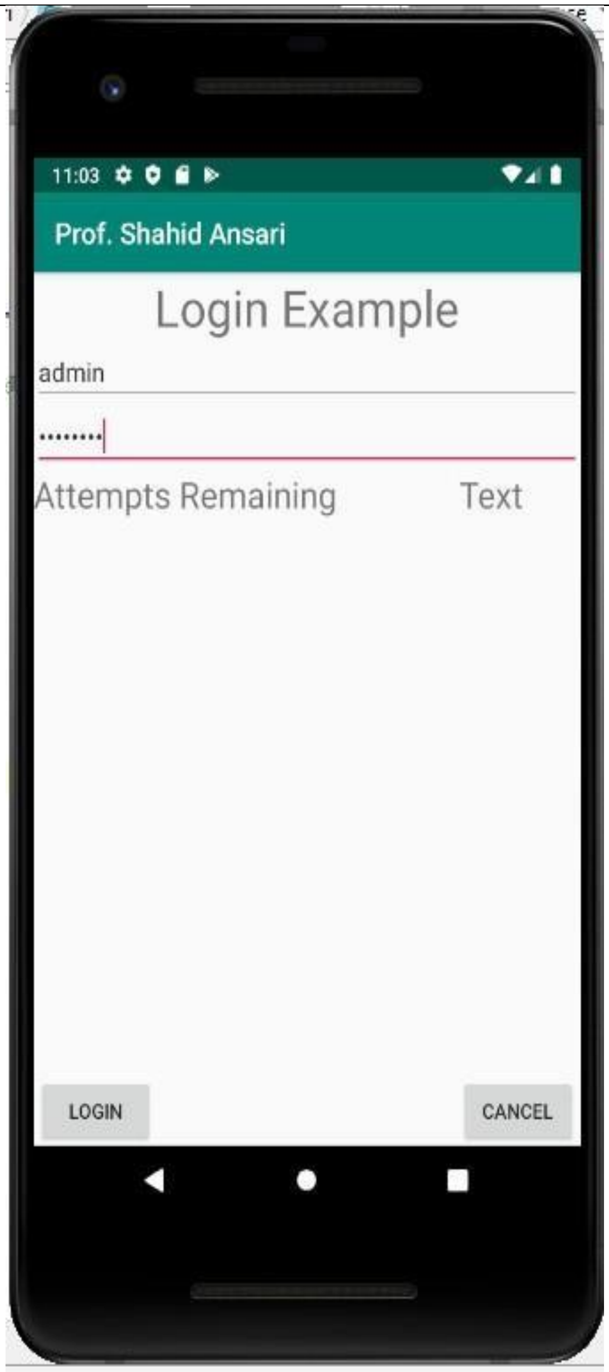
Main_Activity.java

```
package com.maharashtracollege.profshahidansari;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;


import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

   Button b1,b2;
   EditText ed1,ed2;

   TextView tx1;
   int counter = 3;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);


      b1 = (Button)findViewById(R.id.button);
      ed1 = (EditText)findViewById(R.id.editText);
      ed2 = (EditText)findViewById(R.id.editText2);

      b2 = (Button)findViewById(R.id.button2);
      tx1 = (TextView)findViewById(R.id.textView3);


      b1.setOnClickListener(new View.OnClickListener() {
         @Override
```

```
            public void onClick(View v) {
                if(ed1.getText().toString().equals("admin")  &&
                        ed2.getText().toString().equals("admin")) {
                    Toast.makeText(getApplicationContext(),
                            "Redirecting...",Toast.LENGTH_SHORT).show();
                }else{
                    Toast.makeText(getApplicationContext(), "Wrong
Credentials",Toast.LENGTH_SHORT).show();

                            tx1.setVisibility(View.VISIBLE);
                    tx1.setBackgroundColor(Color.RED);
                    counter--;
                    tx1.setText(Integer.toString(counter));

                    if (counter == 0) {
                        b1.setEnabled(false);
                    }
                }
            }
        });

        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

    }
}
```


10:42
Prof. Shahid Ansari
Login Example
Enter Name
Password
Attempts Remaining
Text
LOGIN
CANCEL


10:48
Prof. Shahid Ansari
Login Example
admin
•••••
Pwd: admin
Attempts Remaining
Text
Login Successful !!!...
LOGIN
CANCEL


11:03

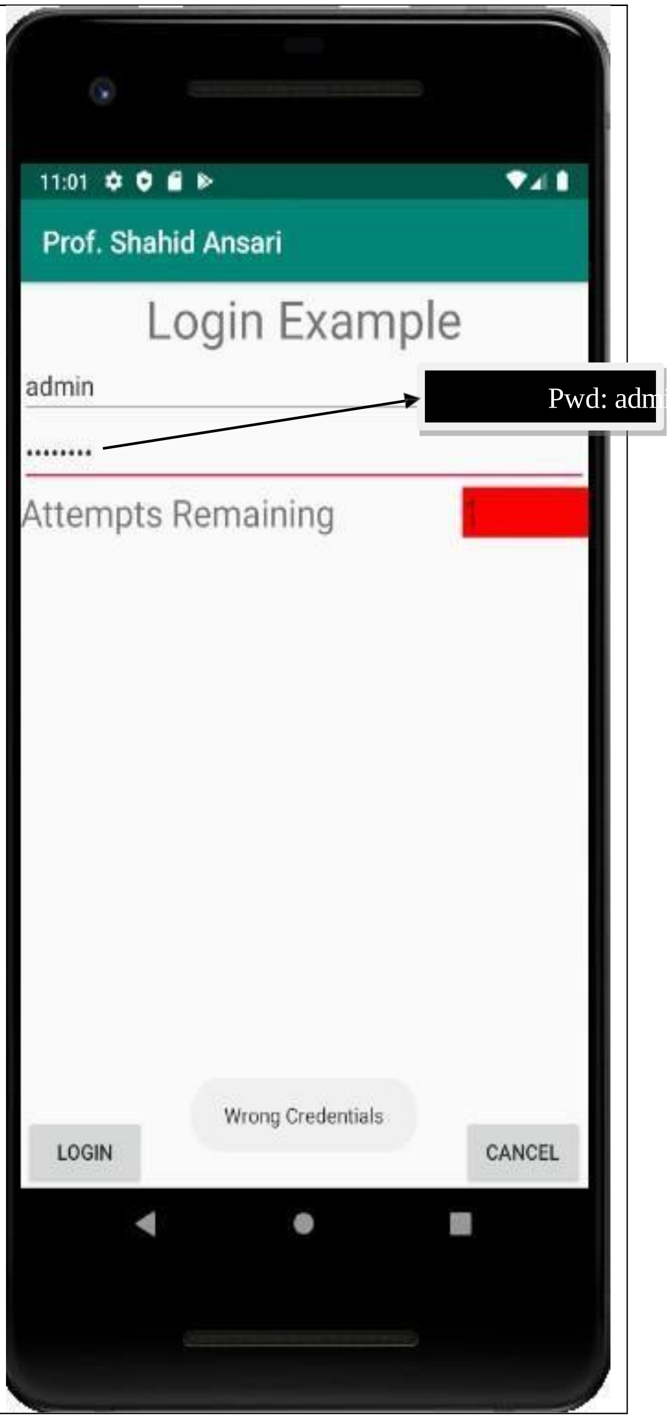
Prof. Shahid Ansari

Login Example

admin

••••••••

Attempts Remaining Text


11:01

Prof. Shahid Ansari

Login Example

admin

Pwd: adm

•••••••••

Attempts Remaining 1

Wrong Credentials

Practical 10

Question:
Create the media API in android to play a video file.

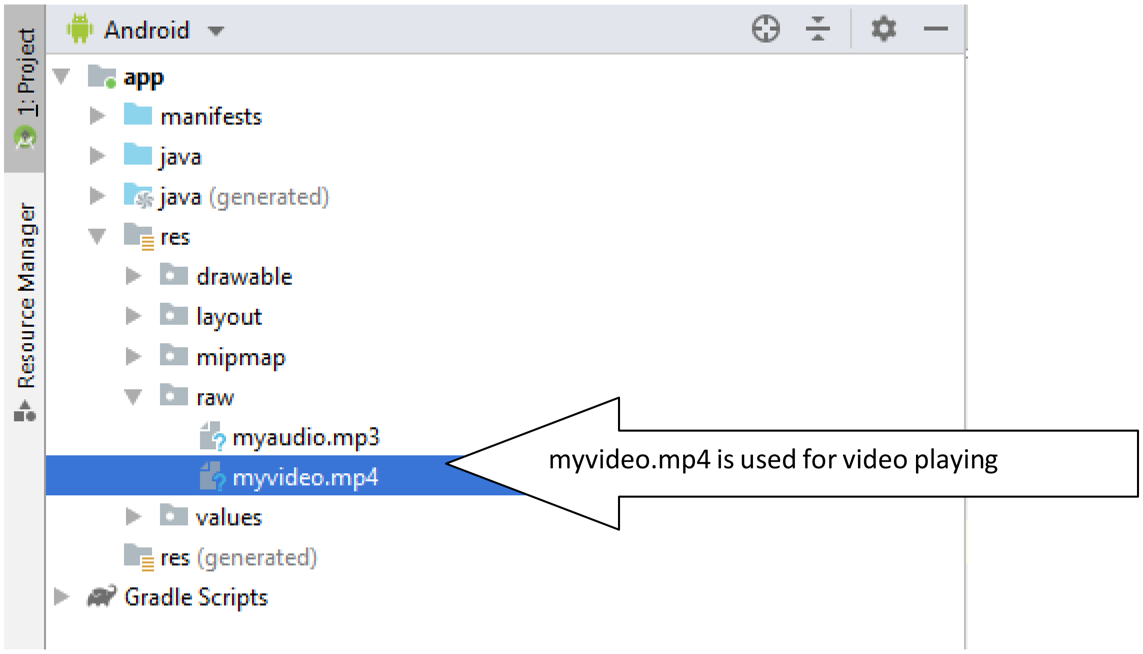
Activity_main.xml
Drag and drop VideoView and Button from Palette

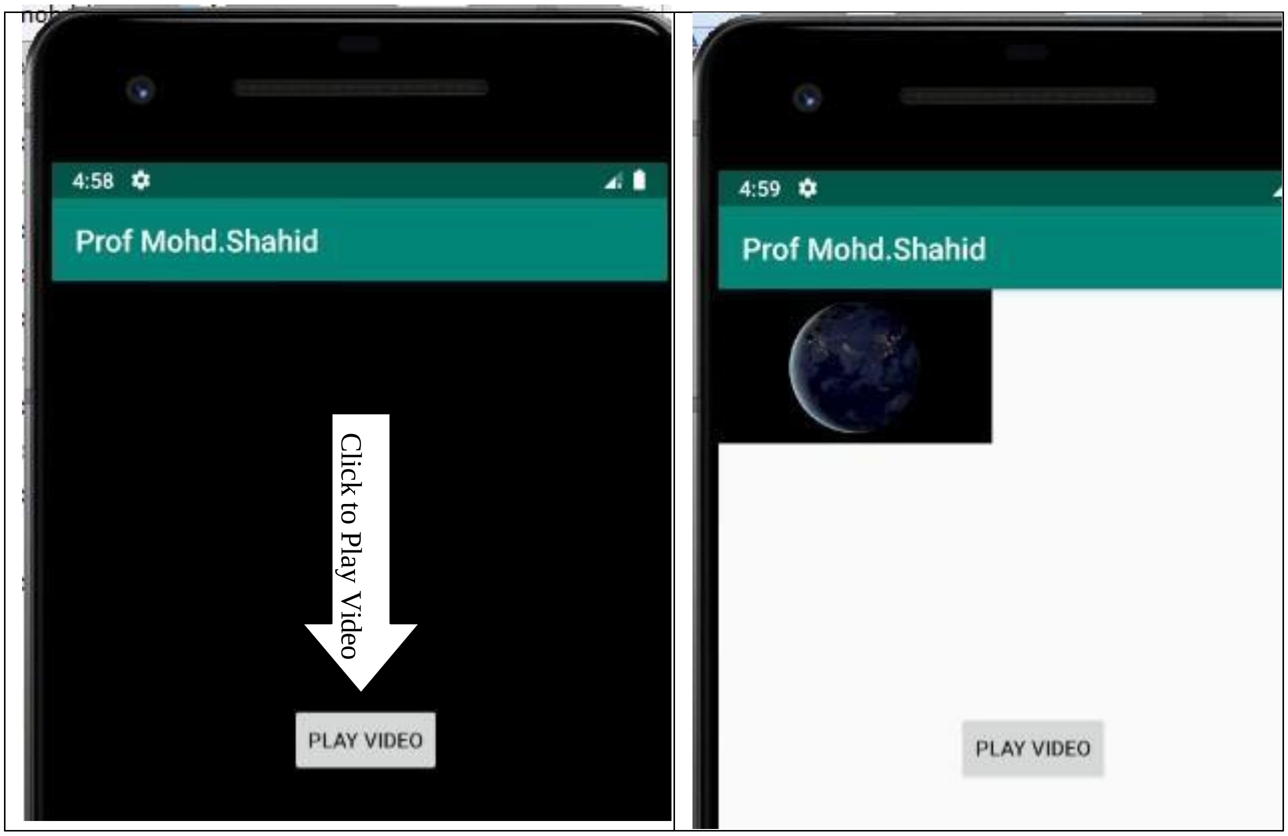
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <VideoView
      android:id="@+id/videoView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"  />

   <Button
      android:id="@+id/btnPlay"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="151dp"
      android:layout_marginRight="151dp"
      android:layout_marginBottom="274dp"
      android:onClick="playVideo"
      android:text="Play Video" />

</RelativeLayout>
```

Main_Activity.java

```
package com.maharashtracollege.profmohdshahid;
import androidx.appcompat.app.AppCompatActivity;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import
android.widget.ImageButton;
import
android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {

    VideoView videoview;
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }
    public void playVideo(View view)
    {
        videoview = (VideoView) findViewById(R.id.videoView);

        try {

            MediaController mediacontroller = new
                    MediaController( MainActivity.this);
            mediacontroller.setAnchorView(videoview);

            Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.myvideo);
            videoview.setMediaController(mediacontrol
            ler); videoview.setVideoURI(uri);

        } catch (Exception e) {
            Log.e("Error",
            e.getMessage());
            e.printStackTrace();
        }
        videoview.requestFocus();
        videoview.setOnPreparedListener(new  MediaPlayer.OnPreparedListener()  {
            public void onPrepared(MediaPlayer mp) {
                videoview.start();
  }
 }
);
    }}
```

Note: For this practical we to copy the .mp4 file and create the folder raw in the res as follows


1: Project
Resource Manager
Android
app
manifests
java
java (generated)
res
drawable
layout
mipmap
raw
myaudio.mp3
myvideo.mp4
values
res (generated)
Gradle Scripts
myvideo.mp4 is used for video playing


4:58
Prof Mohd.Shahid
Click to Play Video
PLAY VIDEO
4:59
Prof Mohd.Shahid
PLAY VIDEO