University of Mumbai

Institute of Distance and Open Learning (IDOL)

PCP CENTER: DTSS, MALAD

INDEX

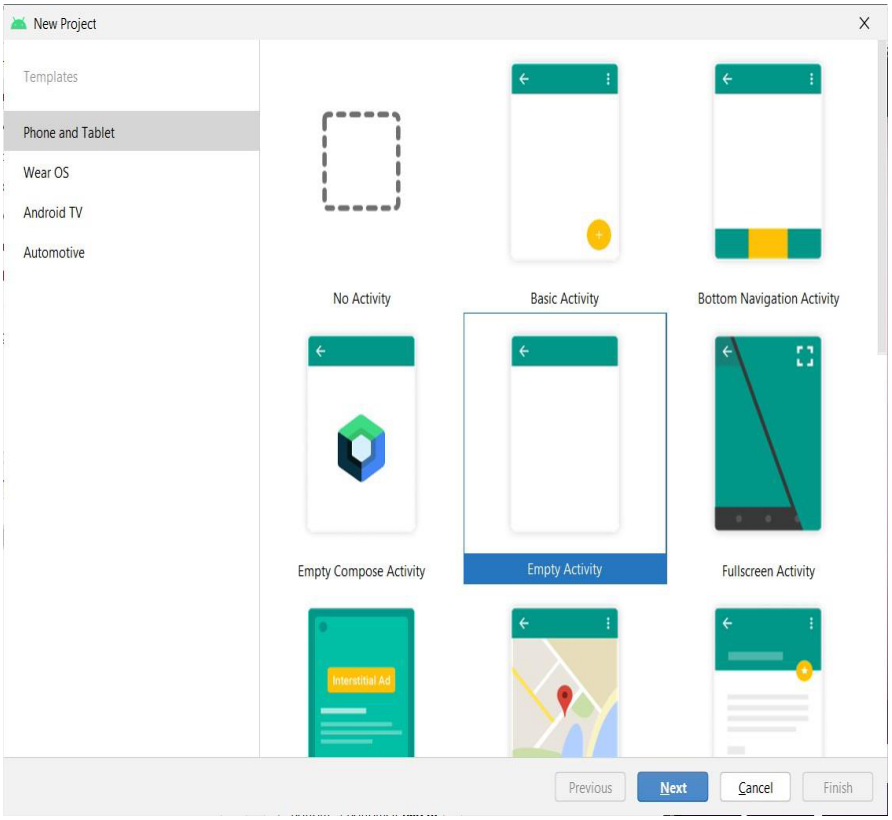
Practical No.	Title	Sign	Remark
1	**Android program using various UI components**		
2	**Android program using different layouts and views**		
3	**Android program based on Intents**		
4	**Android program for notifications and alert box**		
5	**Android program to perform CRUD operation using SQLite DB**		
6	**Android program using Shared Preferences, Internal and External Storage**		
7	**Android program to work with Google Maps and locations**		
8	**Android program to work with images and videos**		
9	**Android program based on RestAPI**		
10	**Flutter program to work with SQLite Database**		

PRACTICAL 1

AIM: Android program using various UI components

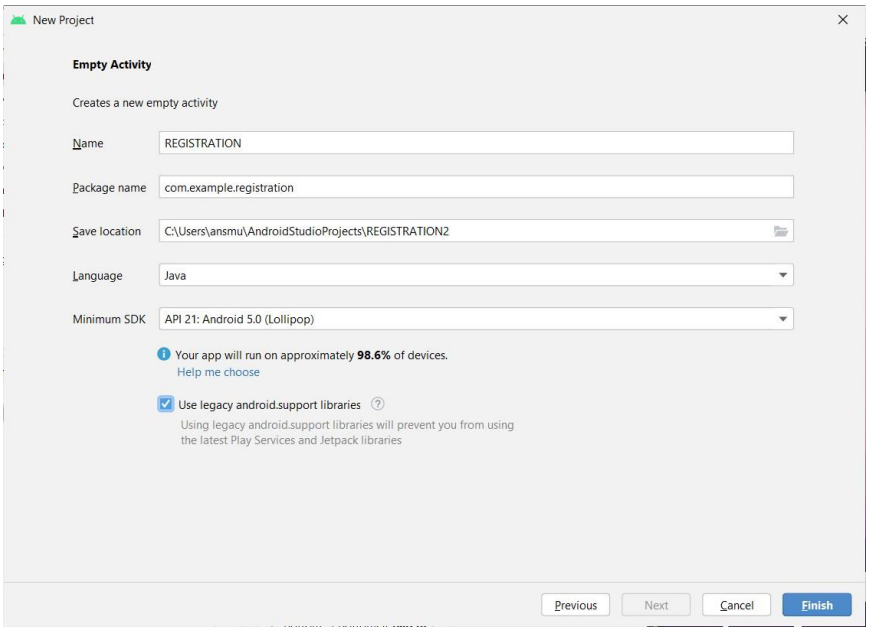
1) To create your new Android project, click on **Create New Project** on the **Welcome to Android Studio** window.
2) If you have a project already opened, select **File > New > New Project**.
3) In the **Select a Project Template** window, select **Empty Activity** andclick **Next**.

1818



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

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

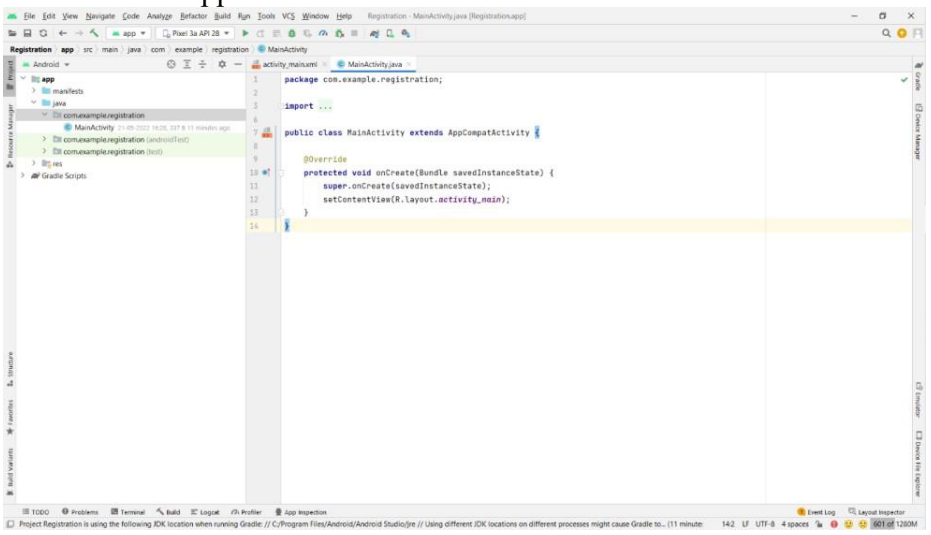


5) Click on finish

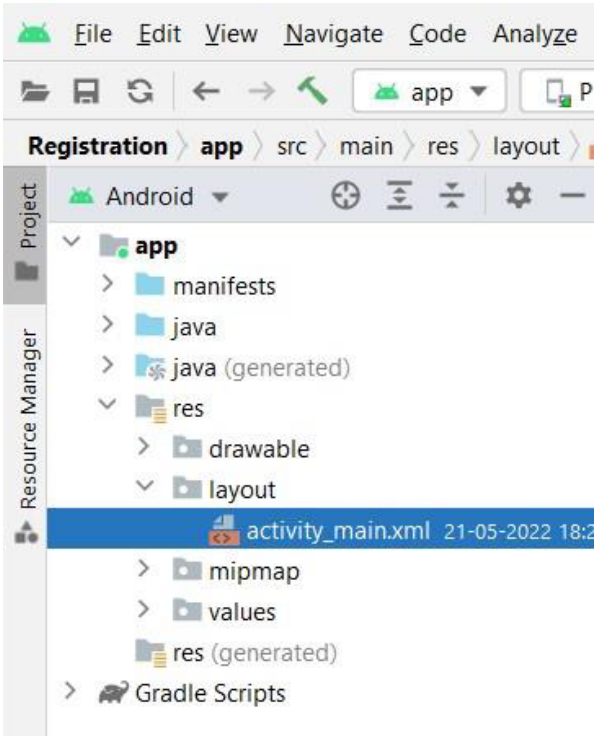
In this program we are going to use the following controls:

1. ImageButton
2. Text(Plain Text)
3. Text (E Mail)
4. Text (Phone)
5. Text (Postal Address)
6. CheckBox
7. RadioGroup with two RadioButton
8. Button

After some processing time, the Android Studio main window appears as follows:



2020 As in this practical we have to make use of different UI controls we will click on **res** folder and then open **activity_main.xml** file and start coding.



We can design activity_main.xml file in three different modes: **Code mode, split mode and Design mode. Code for activity_main.xml file**

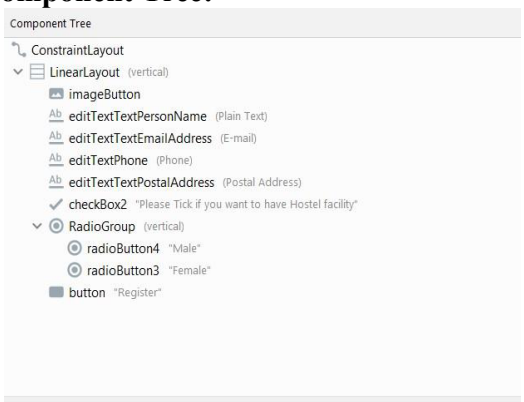
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"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

    <LinearLayout
        android:layout_width="wrap_content"         android:layout_height="wrap_content"
android:orientation="vertical"         app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

                <ImageButton
                    android:id="@+id/imageButton"
            android:layout_width="match_parent"
            android:layout_height="100dp"
            android:scaleType="fitCenter"
            app:srcCompat="@drawable/img" />

                <EditText
```

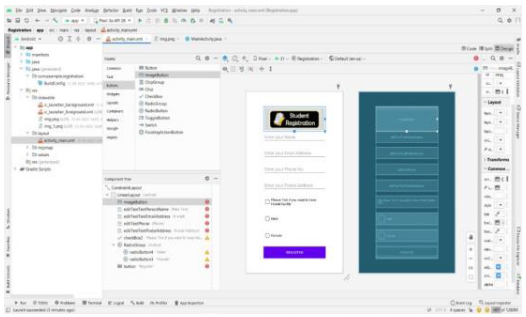
```
                android:id="@+id/editTextTextPersonName"
            android:layout_width="match_parent"
            android:layout_height="match_parent"          android:ems="20"
                android:inputType="textPersonName"
            android:layout_marginBottom="20dp"          android:hint="Enter
            your Name" />

                <EditText
                android:id="@+id/editTextTextEmailAddress"
            android:layout_width="265dp"
            android:layout_height="wrap_content"
            android:hint="Enter your Email Address"
            android:ems="10"
                android:layout_marginBottom="20dp"
            android:inputType="textEmailAddress" />
2222
    <EditText
        android:id="@+id/editTextPhone"          android:layout_width="265dp"
android:layout_height="wrap_content"          android:ems="10"
        android:layout_marginBottom="20dp"          android:hint="Enter your
Phone No"          android:inputType="phone" />

    <EditText
        android:id="@+id/editTextTextPostalAddress"
android:layout_width="match_parent"
android:layout_height="wrap_content"          android:ems="10"
        android:layout_marginBottom="20dp"          android:hint="Enter your
Postal Address"          android:inputType="textPostalAddress" />

    <CheckBox
        android:id="@+id/checkBox2"          android:layout_width="match_parent"
android:layout_height="wrap_content"          android:layout_marginBottom="20dp"
        android:text="Please Tick if you want to have Hostel facility" />

    <RadioGroup
        android:layout_width="match_parent"
android:layout_height="match_parent" >          <RadioButton
                    android:id="@+id/radioButton4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:text="Male" />

                <RadioButton
                    android:id="@+id/radioButton3"
            android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_marginBottom="20dp"
android:text="Female" />

        </RadioGroup>

        <Button
            android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="#F1A7A7"            android:text="Register"
/>

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

2424

Component Tree:





Activity_main.xml in design mode Note:

In this program we are only designing user interface and not providing any functionality to it. That is why, writing java files is not required.

As in this practical we have use ImageButton Control, first we have to add image in drawable folder. Then that image will be displayed on the button.

MainActivity.java (This is autogenerated code,we can add our code as per our program in this file) package com.example.registration;

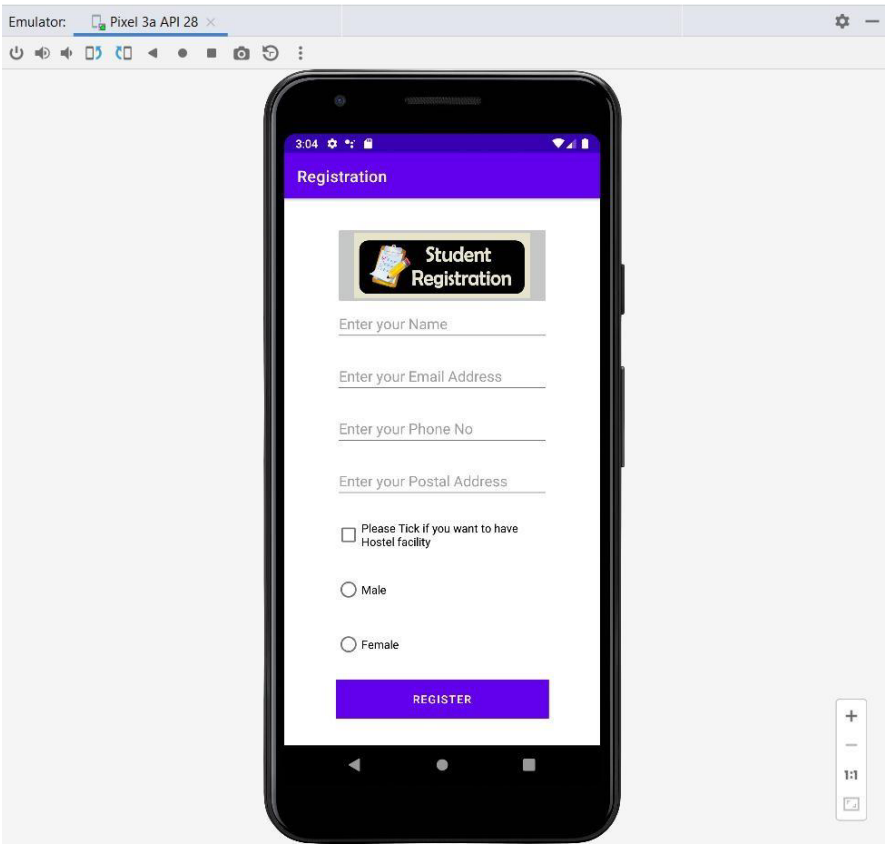
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

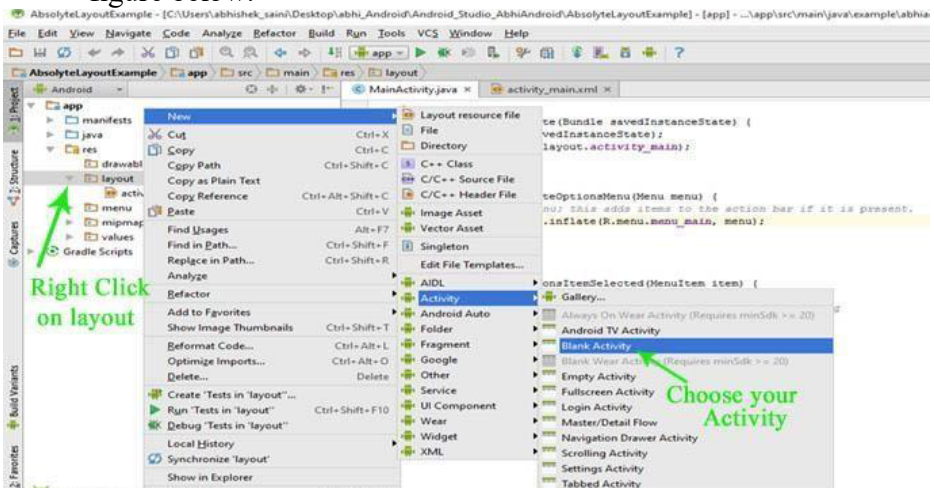
2.4.1 Output :



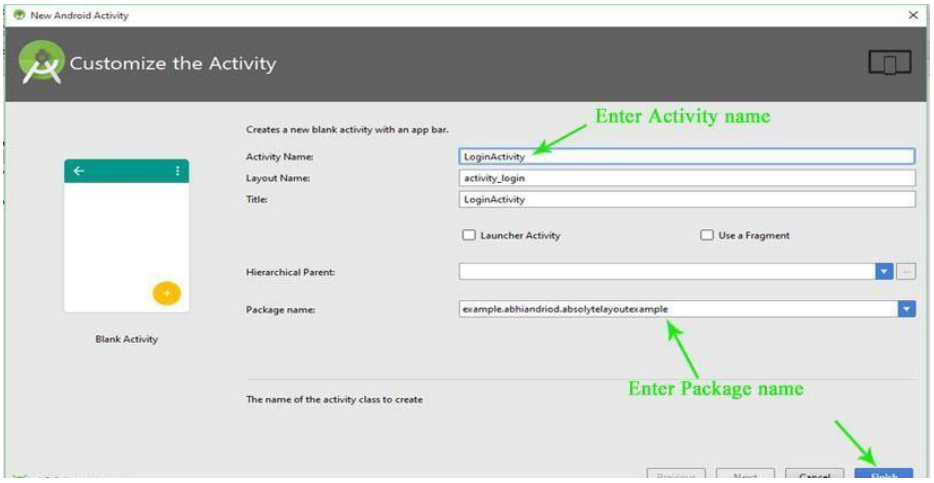
PRACTICAL 2

AIM: Android program using different layouts and views

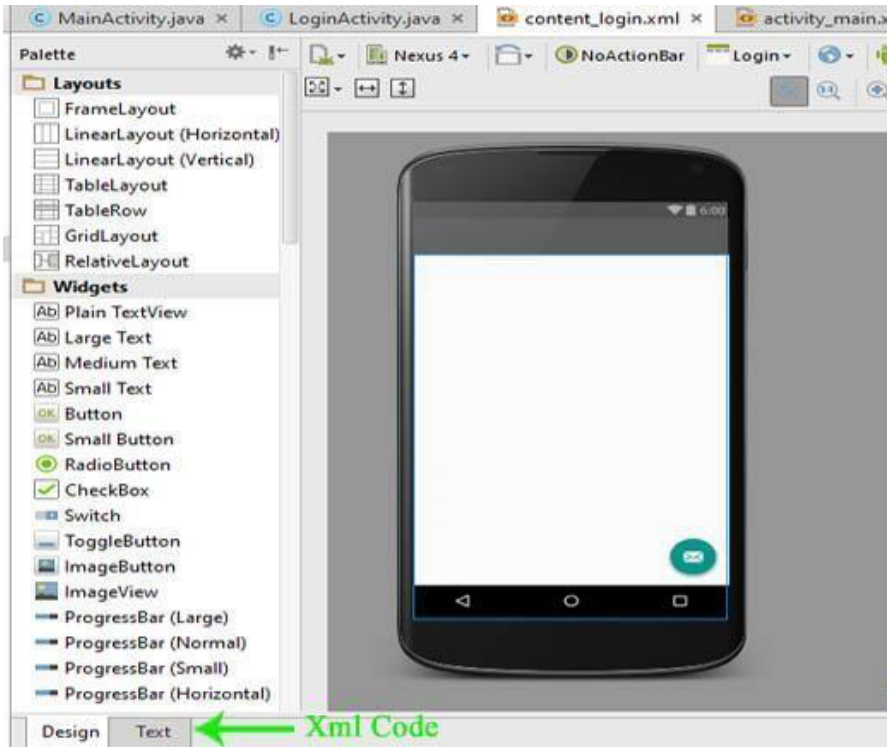
Step 1: Firstly, click on app > res > layout > Right Click on layout. After that Select New > Activity and choose your Activity as per requirement. Here we choose Blank Activity as shown in figure below.



Step 2: After that Customize the Activity in Android Studio. Enter the "Activity Name" and "Package name" in the Text box and Click on Finish button.



Step 3: After that your new Activity in Layout will be created. Your XML Code is in Text and your Design Output is in Design.



Design user interface with views, working:

In Android applications, various types of ViewGroups are used to design UI.

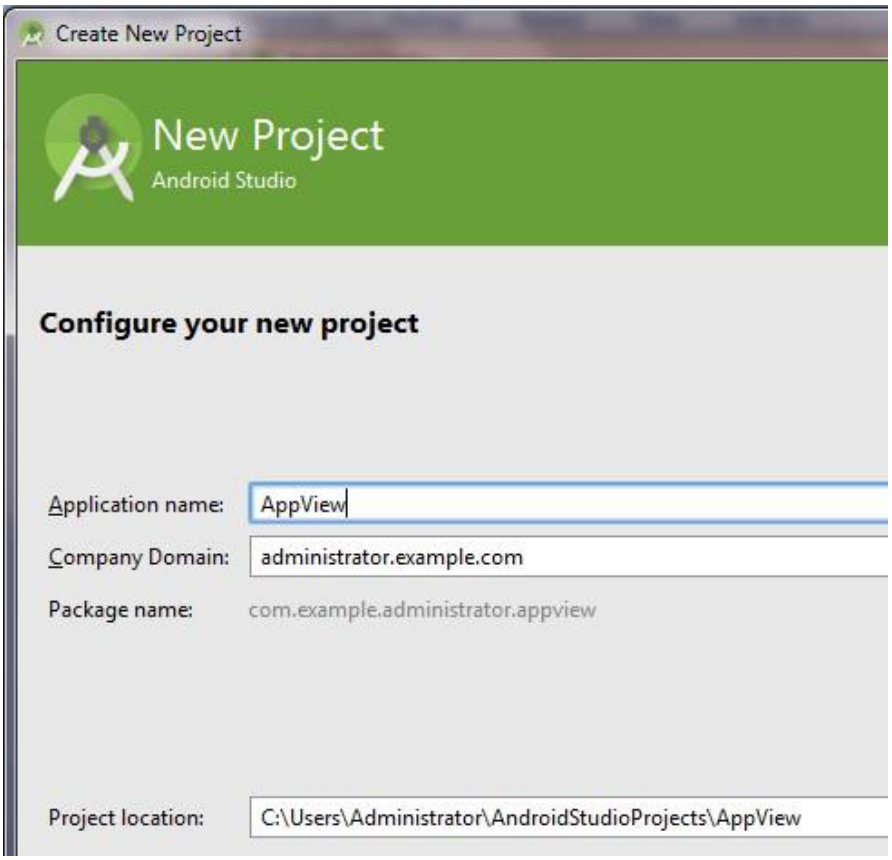
1. Basic Views
2. Picker Views
3. List Views
4. Specialized Fragments
5. Analog and Digital Clock Views

In Android applications, the following are Basic Views.

- TextView
- EditText
- Button
- ImageButton
- CheckBox
- ToggleButton
- RadioButton
- RadioGroup

Implementation:

Create a new Android project called AppView. By default, it creates main.xml file located in the res/layout folder, which contains a <TextView> element.




```
1.  <LinearLayout
2.      xmlns:android="http://schemas.android.com/apk/res/android"
3.      xmlns:tools="http://schemas.android.com/tools"
4.
5.      android:layout_width="fill_parent"
6.      android:layout_height="match_parent"
7.      android:orientation="vertical"
8.      tools:context=".MainActivity">
9.  <TextView  android:text="@string/hello_world"
10.     android:layout_width="fill_content"
11.     android:layout_height="wrap_content"   />
12. </LinearLayout >
```

The TextView is used to display text/caption to the user. This is the most basic View and very frequently used in an application.
The next View is a subclass of TextView and it is EditText. This View allows the user to edit the text displayed.

```
<EditText
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:id="@+id/txtUserName" />
```

Button represents a push-button widget.

```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btnAdd"
    android:text="Add"/>
```

ImageButton is similar to Button View except that it displays an image with text.

```
<ImageButton
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/imgButton"
    android:src="@drawable/abc_ic_menu_copy_mtrl_am_alpha"
/>
```

CheckBox is a type of button that has two states; i.e., checked or unchecked.

```
<CheckBox
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:id="@+id/chkIndia"
  android:text="India"
/>
<CheckBox
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
10.    android:id="@+id/chkUS"
style="?android:attr/starStyle"
11.    android:text="US"

12. />
```

RadioGroup and RadioButton, both have two states: either checked or unchecked. A RadioGroup is used to group together one or more RadioButton Views, thereby allowing only one RadioButton to be checked within the RadioGroup.

```
1.   <RadioGroup
2.      android:layout_width="fill_parent"
3.      android:layout_height="wrap_content"
4.      android:orientation="vertical"
5.      android:id="@+id/rdoGroup">
6.   <RadioButton
7.      android:id="@+id/rbMal e"
8.      android:layout_width="fill_parent"
9.      android:layout_height="wrap_content"
10.     android:text="Male"/>
11.  <RadioButton
12.     android:id="@+id/rbFemale"
13.     android:layout_width="fill_parent"
14.     android:layout_height="wrap_content"
15.     android:text="Female"/>
16.  </RadioGroup>
```

ToggleButton displays checked/unchecked states using a light indicator.

```
1.   <ToggleButton
2.      android:layout_width="fill_parent"
3.      android:layout_height="wrap_content"
4.      android:id="@+id/toggleButton"/>
```

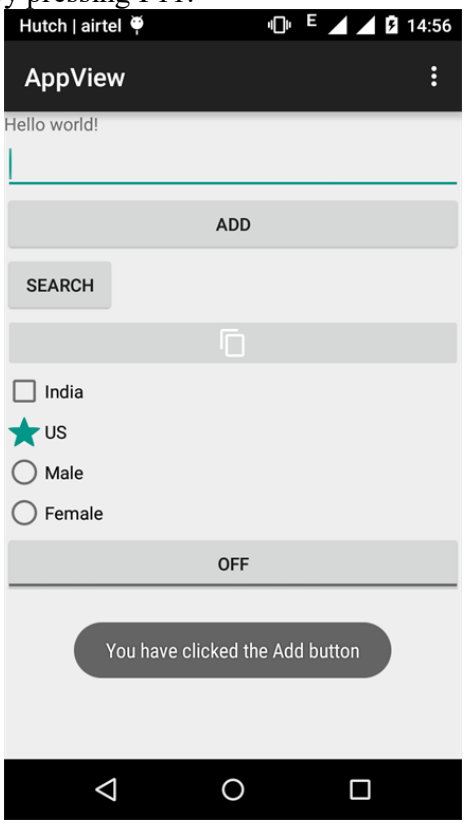

Now, add some code as below to handle View events for elements, like Button, CheckBox etc.

```
1.  Button buttonAdd = (Button) findViewById(R.id.btnAdd);
2.  buttonAdd.setOnClickListener(new View.OnClickListener()
3.    {
4.      @Override
5.      public void onClick(View view)
6.      {
7.          DisplayMessage("You have clicked the Add button");
8.      }
9.  });
```

Write a common method to display the text message as below.

```
1.  private void DisplayMessage(String textMessage) {
2.  Toast.makeText(getBaseContext(), textMessage, Toast.LENGTH
    _SHORT).show();
3.  }
```

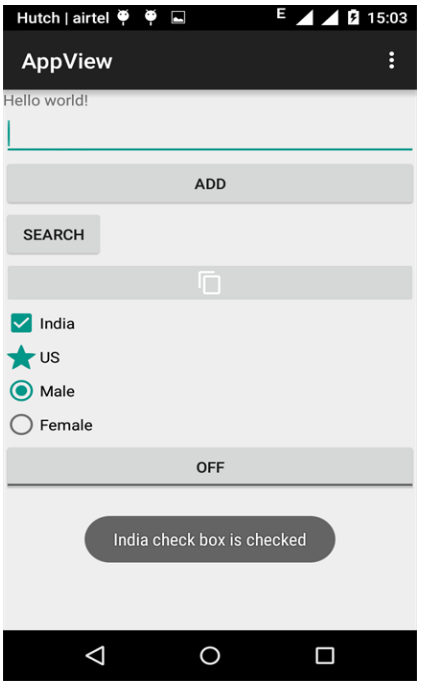
Now, run the application by pressing F11.



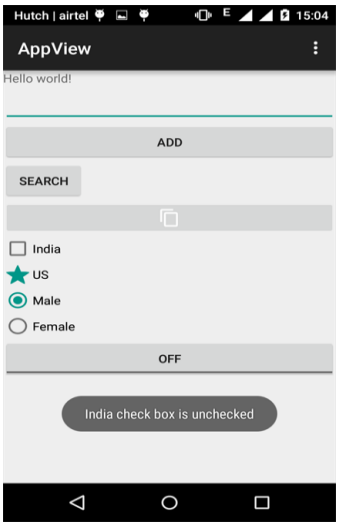
To handle the View events for element CheckBox, add the below code.

```
1.  CheckBox  checkBox  = (CheckBox ) findViewById(R.id.chkIndia);
2.  checkBox.setOnClickListener(  new  View.OnClickListener()
3.  {
4.      @Override
5.      public  void  onClick(View  view)
6.      {
7.          if (((CheckBox ) view).isChecked())  DisplayMessage("India  ch
eck box is checked");
8.          else  DisplayMessage("India  check box is unchecked");
9.      }
10.  });
```

Now, run the application by pressing F11. For example, I have checked the India CheckBox.

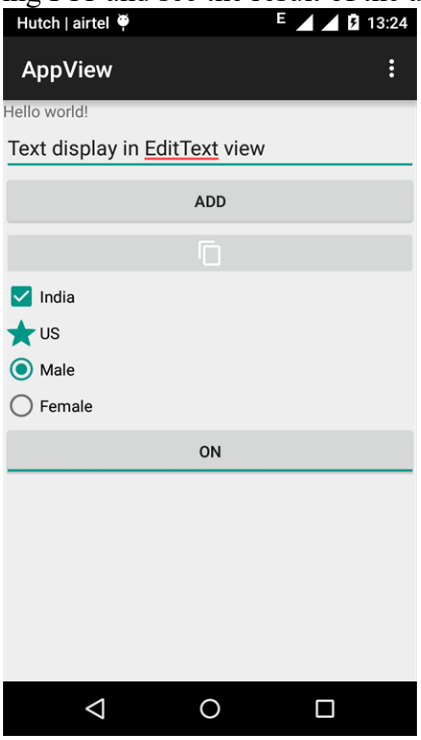


Now, uncheck the India CheckBox.



Explanation:

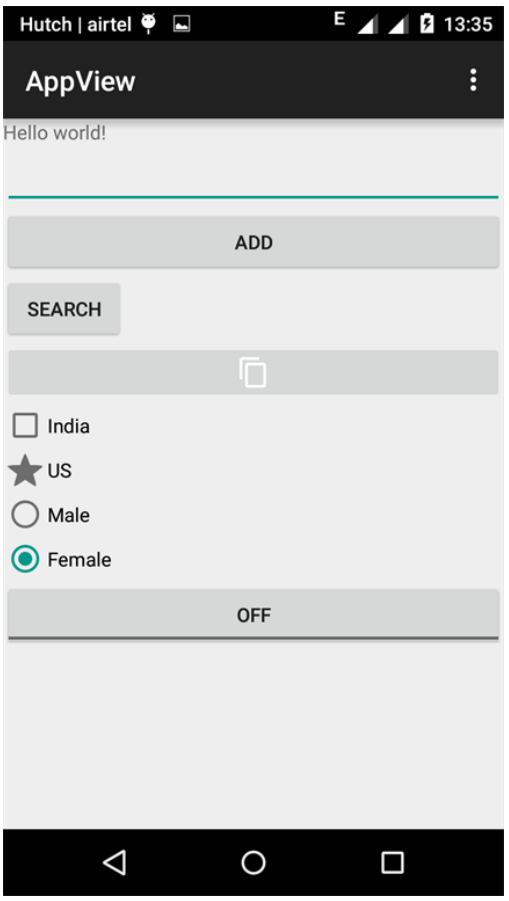
Run the application by pressing F11 and see the result of the above implemented code.



All the Views are relatively straightforward because they are listed using the <LinearLayout> element. So, they are stacked on top of each other when they are displayed in the activity.

```
1.  <Button
2.      android:layout_width="fill_parent"
3.      android:layout_height="wrap_content"
4.      android:id="@+id/btnAdd"
5.      android:text="Add"/>
6.
7.  <Button
8.      android:layout_width="wrap_content"
9.      android:layout_height="wrap_content"
10.     android:id="@+id/btnSearch"
11.     android:text="Search"/>
```

In the above code for first Button, the layout_width attribute is set to fill_parent so that its width occupies the entire width of the screen.

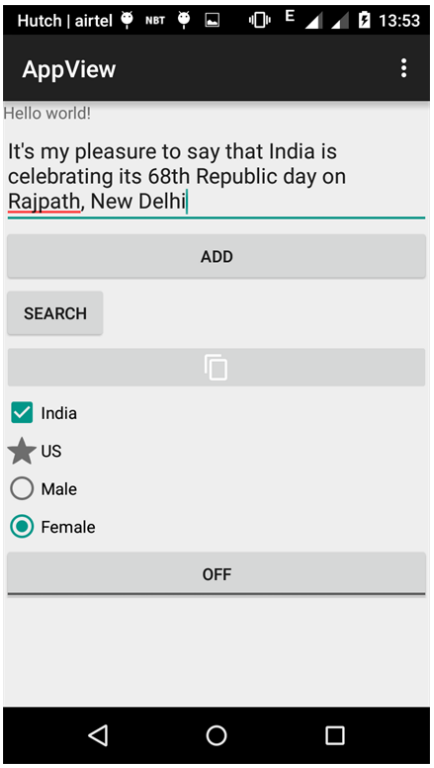


But for the second Button, the layout_width attribute is set to wrap_content so that its width will be the width of its content.
The ImageButton displays a button with an image. The image is set through the src attribute.

```
1.  <ImageButton
2.      android:layout_width="fill_parent"
3.      android:layout_height="wrap_content"
4.      android:id="@+id/imgButton"
5.      android:src="@drawable/abc_ic_menu_copy_mtrl_am_alpha"
6.  />
```

EditText displays a rectangular region where the user can enter some text. Set the attribute of layout_height to wrap_content so that if the user enters a long string of text, its height will automatically be adjusted to fit the content.

```
1.  <EditText
2.      android:layout_width="fill_parent"
3.      android:layout_height="wrap_content"
4.      android:id="@+id/txtUserName"
5.  />
```



The CheckBox displays a checkbox that users can tap on to check or uncheck.

```
1.  <CheckBox
2.    android:layout_width="fill_parent"
3.    android:layout_height="wrap_content"
4.    android:id="@+id/chkIndia"
5.    android:text="India"
6.  />
7.  <CheckBox
8.    android:layout_width="fill_parent"
9.    android:layout_height="wrap_content"
10.   android:id="@+id/chkUS"style="?android:attr/starStyle"
11.   android:text="US"
12. />
```

In the above code, style attribute is set to display another image rather than default one.

There is RadioGroup that encloses two RadioButtons. This is very important because radio buttons are usually used to present multiple options to the user for selection. When a RadioButton in a RadioGroup is selected, all the other RadioButtons are automatically unselected.

```
1.    <RadioGroup
2.    android:layout_width="fill_parent"
android:layout_height="wrap_content" 12.
android:id="@+id/rbFemale"
13.   android:layout_width="fill_parent"
14.   android:layout_height="wrap_content"
15.   android:text="Female"/>
16. </RadioGroup>
```

In the above code, RadioButtons are displayed vertically. To display the list horizontally, just change the orientation attribute to horizontal. If you change the orientation attribute to horizontal, you will have to change the layout_width attribute to wrap_content.

The ToogleButton displays a rectangular button that users can toggle either ON or OFF, by tapping.

```
1.  <ToggleButton
2.    android:layout_width="fill_parent"
3.    android:layout_height="wrap_content"
```

```
4.      android:id="@+id/toggleButton"/>
```

In the above code, each element has the id attribute with a particular value as we can see for the Button View.

```
1.  <Button
2.      android:layout_width="wrap_content"
3.      android:layout_height="wrap_content"
4.      android:id="@+id/btnSearch"
5.      android:text="Search"/>
```

The id attribute is an identifier for a View so that it may later be retrieved using the View.findViewById() methods.

To handle View events for Button element, add the code given below.

```
1. Button buttonAdd=(Button) findViewById(R.id.btnAdd);
```

The setOnClickListener() method registers a callback to be invoked later when the View is clicked.

```
1.  buttonAdd.setOnClickListener(new
    View.OnClickListener() {
2.      @Override
3.      public void onClick(View view) {
4.          DisplayMessage("You have clicked the Add
    button");
5.      }
6.  });
```

The onClick() method is called when the View is clicked.

The next element is CheckBox for which code is added to handle their View events. To determine the state of the CheckBox, you must have to typecast the argument of the onClick() method to a CheckBox and then click its isChecked() method to see if it is clicked or not.

```
1.      CheckBox checkBox = (CheckBox)
        findViewById(R.id.chkIndia);
2.      checkBox.setOnClickListener(new
        View.OnClickListener() {
3.      @Override
4.      public void onClick(View view) {
5.      if (((CheckBox) view).isChecked())
        DisplayMessage("India ch eck box is checked");
6.      else DisplayMessage("India check box is unchecked");
7.      }
});
```

PRACTICAL 3

AIM: Android program based on Intents

activity_main.xml

File: activity_main.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.and
   roid.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent"
7.     tools:context="example.javatpoint.com.implicitintent.MainActivity">
8.
9.     <EditText
10.        android:id="@+id/editText"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:layout_marginEnd="8dp"
14.        android:layout_marginStart="8dp"
15.        android:layout_marginTop="60dp"
16.        android:ems="10"
17.        app:layout_constraintEnd_toEndOf="parent"
18.        app:layout_constraintHorizontal_bias="0.575"
19.        app:layout_constraintStart_toStartOf="parent"
20.        app:layout_constraintTop_toTopOf="parent" />
21.
22.    <Button
23.        android:id="@+id/button"
24.        android:layout_width="wrap_content"
25.        android:layout_height="wrap_content"
26.        android:layout_marginRight="8dp"
27.        android:layout_marginLeft="156dp"
28.        android:layout_marginTop="172dp"
29.        android:text="Visit"
30.        app:layout_constraintEnd_toEndOf="parent"
31.        app:layout_constraintHorizontal_bias="0.0"
32.        app:layout_constraintStart_toStartOf="parent"
33.        app:layout_constraintTop_toBottomOf="@+id/editText" />
34. </android.support.constraint.ConstraintLayout>
```

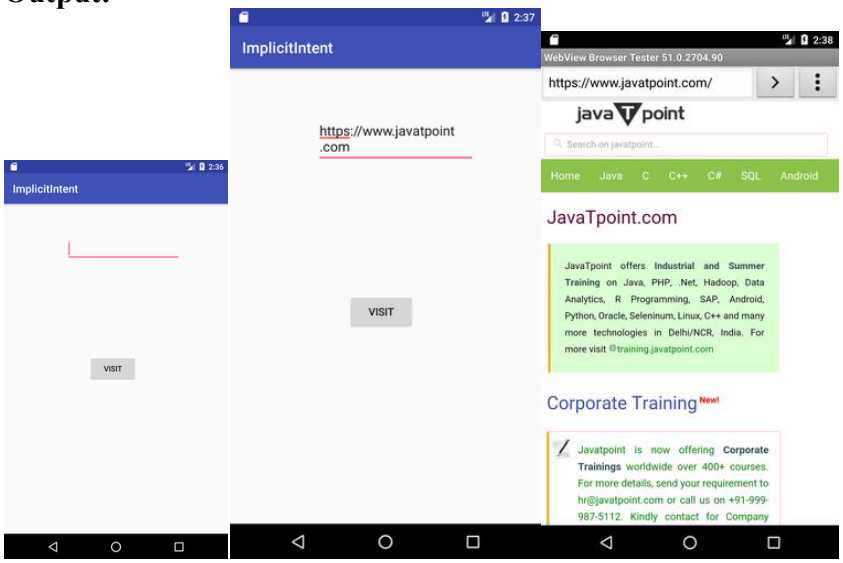
Activity class

File: MainActivity.java

```
1. package example.javatpoint.com.implicitintent;
2.
3. import android.content.Intent;
```

```
4.  import android.net.Uri;
5.  import android.support.v7.app.AppCompatActivity;
6.  import android.os.Bundle;
7.  import android.view.View;
8.  import android.widget.Button;
9.  import android.widget.EditText;
10.
11. public class MainActivity extends AppCompatActivity {
12.
13.     Button button;
14.     EditText editText;
15.
16.     @Override
17.     protected void onCreate(Bundle savedInstanceState) {
18.         super.onCreate(savedInstanceState);
19.         setContentView(R.layout.activity_main);
20.
21.         button = findViewById(R.id.button);
22.         editText =  findViewById(R.id.editText);
23.
24.         button.setOnClickListener(new View.OnClickListener() {
25.             @Override
26.             public void onClick(View view) {
27.                 String url=editText.getText().toString();
28.                 Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
29.                 startActivity(intent);
30.             }
31.         });
32.     }
33. }
```

Output:



PRACTICAL 4

AIM:Android program for notifications and alert box

Step 1: Create a New Project in Android Studio

To create a new project in Android Studio please refer to **How to Create/Start a New Project in Android Studio** . The code for that has been given in both **Java and Kotlin Programming Language for Android.**

Step 2: Working with the XML Files

Next, go to the **activity_main.xml file** , which represents the UI of the project. Below is the code for the **activity_main.xml** file. Comments are added inside the code to understand the code in more detail.

XML

```
1
<?xml version="1.0" encoding="utf-8"?>
2
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3
    xmlns:tools="http://schemas.android.com/tools"
4
    android:layout_width="match_parent"
5
    android:layout_height="match_parent"
6
    tools:context=".MainActivity">
7

8
    <TextView
9
        android:layout_width="wrap_content"
10
        android:layout_height="wrap_content"
11
        android:layout_marginTop="180dp"
12
        android:gravity="center_horizontal"
13
        android:text="Press The Back Button of Your Phone."
14
        android:textSize="30dp"
15
        android:textStyle="bold" />
16
</RelativeLayout>
```

Step 3: Working with the MainActivity File

Go to the MainActivity File and refer to the following code. Below is the code for the MainActivity File. Comments are added inside the code to understand the code in more detail.

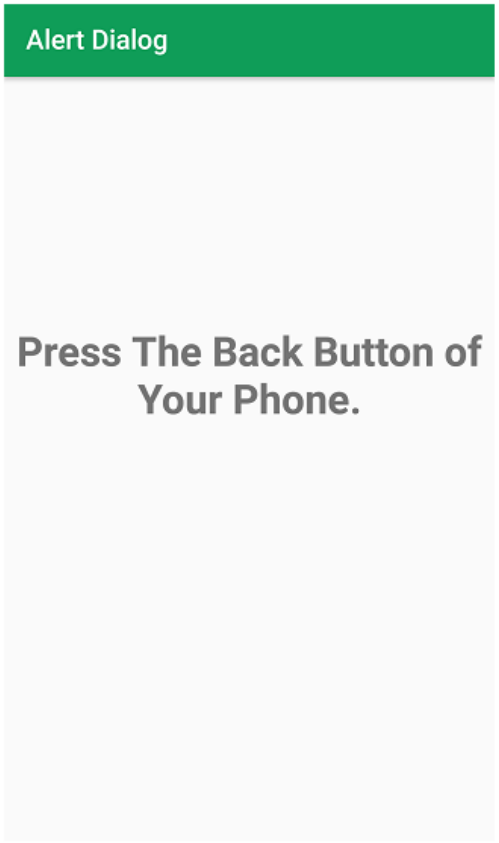
JavaKotlin

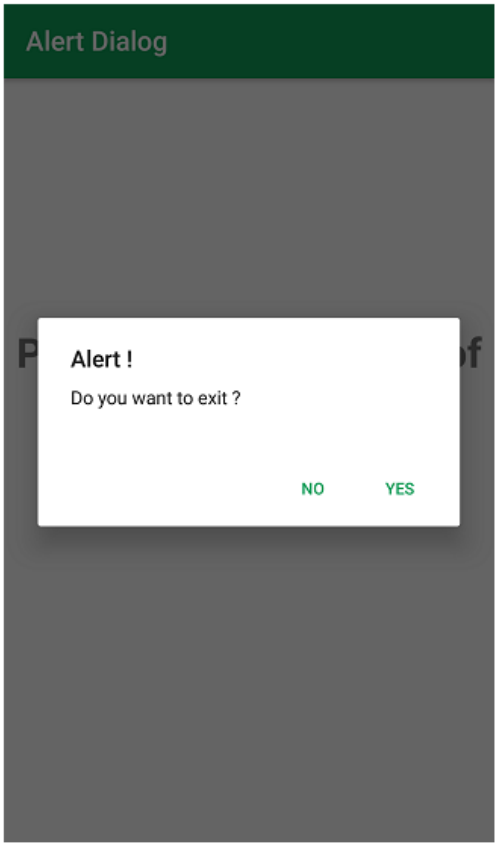
```
1
import android.content.DialogInterface;
2
import android.os.Bundle;
3
import androidx.appcompat.app.AlertDialog;
4
import androidx.appcompat.app.AppCompatActivity;
5

6
public class MainActivity extends AppCompatActivity {
7

8
    @Override
9
    protected void onCreate(Bundle savedInstanceState) {
10
        super.onCreate(savedInstanceState);
11
        setContentView(R.layout.activity_main);
12
    }
13

14
    // Declare the onBackPressed method when the back button is pressed this method will
call
15
    @Override
16
    public void onBackPressed() {
17
        // Create the object of AlertDialog Builder class
18
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
19

20
        // Set the message show for the Alert time
21
```

```
        builder.setMessage("Do you want to exit ?");
22
23
        // Set Alert Title
24
        builder.setTitle("Alert !");
25
26
        // Set Cancelable false for when the user clicks on the outside the Dialog Box then it
will remain show
27
        builder.setCancelable(false);
28
29
        // Set the positive button with yes name Lambda OnClickListener method is use of
DialogInterface interface.
30
        builder.setPositiveButton("Yes", (DialogInterface.OnClickListener) (dialog, which)
-> {
31
            // When the user click yes button then app will close
32
            finish();
33
        });
34
35
        // Set the Negative button with No name Lambda OnClickListener method is use of
DialogInterface interface.
36
        builder.setNegativeButton("No", (DialogInterface.OnClickListener) (dialog, which)
-> {
37
            // If user click no then dialog box is canceled.
38
            dialog.cancel();
39
        });
40
41
        // Create the Alert dialog
```

```
42
        AlertDialog alertDialog = builder.create();
43
        // Show the Alert Dialog box
44
        alertDialog.show();
45
    }
46
}
```

Output:





PRACTICAL 5

AIM: Android program to perform CRUD operation using SQLite DB

Create a New Project and Name it CRUDOperations.

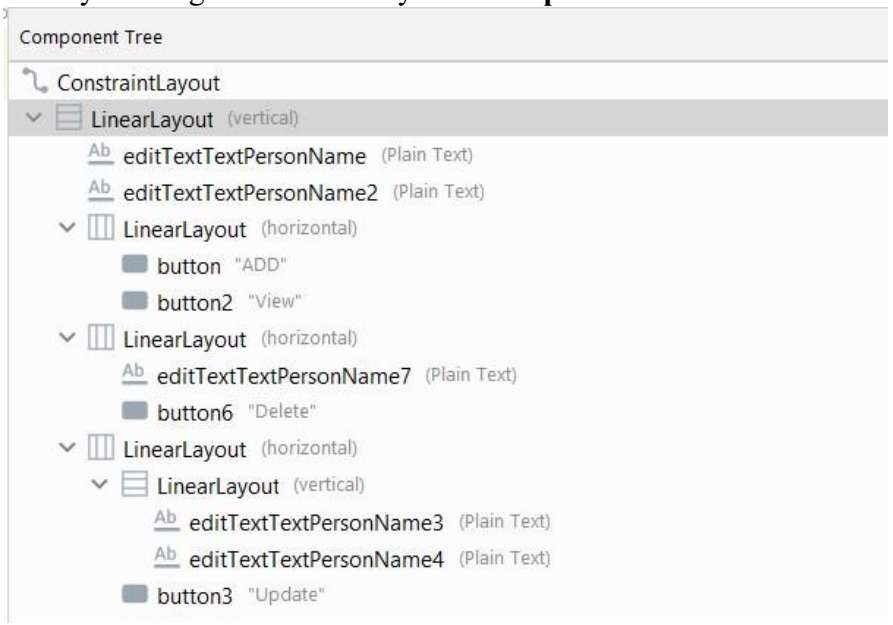
Open res -> layout -> activity_main.xml (or) main.xml and add following code:

Activity_main.xml

```
<?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="match_parent"
android:layout_height="match_parent"         android:orientation="vertical"
android:layout_marginLeft="30sp"         android:layout_marginRight="30sp">

       <EditText             android:id="@+id/editTextTextPersonName"
android:layout_width="match_parent"
android:layout_height="wrap_content"             android:ems="10"
android:hint="Enter User Name"
android:inputType="textPersonName"
android:layout_marginTop="40sp"
android:layout_marginBottom="20sp"/>

       <EditText             android:id="@+id/editTextTextPersonName2"
android:layout_width="match_parent"
android:layout_height="wrap_content"             android:ems="10"
android:hint="Enter Password"
android:inputType="textPersonName"
android:layout_marginTop="20sp"
android:layout_marginBottom="20sp"
          />

       <LinearLayout             android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"             android:layout_marginTop="40sp"
android:layout_marginBottom="40sp">             <Button
android:id="@+id/button"                 android:layout_width="wrap_content"
android:layout_height="wrap_content"                 android:layout_weight="1"
android:text="ADD"                 android:onClick="addUser"
            />

          <Button                 android:id="@+id/button2"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"                android:layout_weight="1"
android:text="View"                android:onClick="viewdata"/>

    </LinearLayout>

    <LinearLayout            android:layout_width="match_parent"
android:layout_height="wrap_content"            android:orientation="horizontal"
android:layout_marginTop="40sp"            android:layout_marginBottom="40sp">

        <EditText
            android:id="@+id/editTextTextPersonName7"
android:layout_width="wrap_content"                android:layout_height="wrap_content"
android:layout_weight="1"                android:ems="10"
            android:hint="Enter User Name to be Deleted"
android:inputType="textPersonName"
             />
        <Button                android:id="@+id/button6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"                android:layout_weight="1"
android:text="Delete"                android:onClick="delete"/>
    </LinearLayout>

    <LinearLayout            android:layout_width="match_parent"
android:layout_height="200dp"            android:layout_weight="1"
android:orientation="horizontal"            android:layout_marginTop="40sp"
android:layout_marginBottom="40sp">

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

            <EditText
                android:id="@+id/editTextTextPersonName3"
android:layout_width="match_parent"
android:layout_height="wrap_content"                    android:ems="10"
                android:inputType="textPersonName"                    android:hint="Enter
Current Username" />

            <EditText
                android:id="@+id/editTextTextPersonName4"
android:layout_width="match_parent"
android:layout_height="wrap_content"                    android:ems="10"
android:inputType="textPersonName"                    android:hint="Enter New User
Name" />
        </LinearLayout>
```

```
        <Button              android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"            android:text="Update"
android:layout_marginTop="30sp"            android:onClick="update"/>
    </LinearLayout>

  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```
Component tree:



Now open app -> java -> package -> MainActivity.java and add the below code.

MainActivity.java:

```
package com.example.crudoperation;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import
android.view.View; import
android.widget.EditText;


public class MainActivity extends AppCompatActivity {
   EditText Name, Pass , updateold, updatenew, delete;
   MyDatabaseAdapter helper;

   @Override     protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
       Name= (EditText) findViewById(R.id.editTextTextPersonName);        Pass=
(EditText) findViewById(R.id.editTextTextPersonName2);
       updateold=                                                    (EditText)
```

```
findViewById(R.id.editTextTextPersonName3);
      updatenew=                                                    (EditText)
findViewById(R.id.editTextTextPersonName4);        delete = (EditText)
findViewById(R.id.editTextTextPersonName7);

      helper = new MyDatabaseAdapter(this);
   }

   public void addUser(View view)
   {
      String t1 = Name.getText().toString();         String t2 =
Pass.getText().toString();         if(t1.isEmpty() || t2.isEmpty())
      {
         Message.message(getApplicationContext(),"Enter Both Name and
Password");
      }         else
      {
         long id = helper.insertData(t1,t2);
         if(id<=0)
         {
            Message.message(getApplicationContext(),"Insertion
Unsuccessful");
            Name.setText("");
            Pass.setText("");
         } else
         {
            Message.message(getApplicationContext(),"Insertion
Successful");
            Name.setText("");
            Pass.setText("");
         }
      }
   }
   public void viewdata(View view)
   {
      String data = helper.getData();
      Message.message(this,data);
   }

   public void update( View view)
   {
      String u1 = updateold.getText().toString();         String u2 =
updatenew.getText().toString();         if(u1.isEmpty() || u2.isEmpty())
      {
         Message.message(getApplicationContext(),"Enter Data");
      }         else
```

```
        {
            int a= helper.updateName( u1, u2);              if(a<=0)
            {
                Message.message(getApplicationContext(),"Unsuccessful");
updateold.setText("");
                updatenew.setText("");
            } else {
                Message.message(getApplicationContext(),"Updated");
updateold.setText("");                  updatenew.setText("");
            }
        }

    }
    public void delete( View view)
    {
        String uname = delete.getText().toString();         if(uname.isEmpty())
        {
            Message.message(getApplicationContext(),"Enter Data");
        }         else{              int a= helper.delete(uname);
if(a<=0)
            {
                Message.message(getApplicationContext(),"Unsuccessful");
delete.setText("");
            }              else
            {
                Message.message(this, "DELETED");                delete.setText("");
            }
        }
    }

}
```

Next step is to create a java class myDatabaseAdapter. Java **MyDatabaseAdapter.java:**

```
package com.example.crudoperation;

import android.content.ContentValues; import
android.content.Context; import android.database.Cursor; import
android.database.sqlite.SQLiteDatabase; import
android.database.sqlite.SQLiteOpenHelper;


    public class MyDatabaseAdapter {         myDbHelper
myhelper;         public MyDatabaseAdapter(Context context)
        {
            myhelper = new myDbHelper(context);
        }
```

```
public long insertData(String name, String pass)
{
   SQLiteDatabase dbb = myhelper.getWritableDatabase();  ContentValues
contentValues = new ContentValues();  contentValues.put(myDbHelper.NAME,
name);  contentValues.put(myDbHelper.MyPASSWORD, pass);  long
id  =  dbb.insert(myDbHelper.TABLE_NAME,  null  , contentValues);
   return id;
}

public String getData()
{
   SQLiteDatabase db = myhelper.getWritableDatabase();
            String[]  columns  =
{myDbHelper.UID,myDbHelper.NAME,myDbHelper.MyPASSWORD};
            Cursor  cursor
=db.query(myDbHelper.TABLE_NAME,columns,null,null,null,null,null);
   StringBuffer buffer= new StringBuffer();
   while (cursor.moveToNext())
   {
      int  cid
=cursor.getInt(cursor.getColumnIndex(myDbHelper.UID));
      String  name
=cursor.getString(cursor.getColumnIndex(myDbHelper.NAME));
      String  password
=cursor.getString(cursor.getColumnIndex(myDbHelper.MyPASSWORD)
);
      buffer.append(cid+ "   " + name + "   " + password +" \n");
   }  return buffer.toString();
}

public  int delete(String uname)
{
   SQLiteDatabase db = myhelper.getWritableDatabase();
   String[] whereArgs ={uname};

   int  count  =db.delete(myDbHelper.TABLE_NAME
,myDbHelper.NAME+" = ?",whereArgs);
   return  count;
}

public int updateName(String oldName , String newName)
{
   SQLiteDatabase db = myhelper.getWritableDatabase();  ContentValues
contentValues = new ContentValues();
contentValues.put(myDbHelper.NAME,newName);  String[] whereArgs=
```

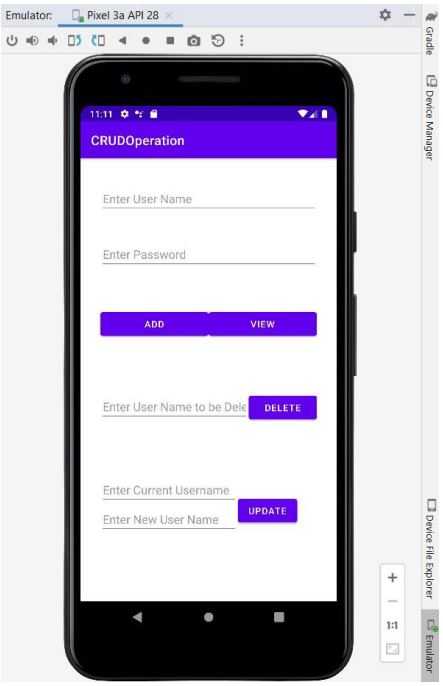
```
{oldName};            int count =db.update(myDbHelper.TABLE_NAME,contentValues,
myDbHelper.NAME+" = ?",whereArgs );
        return count;
    }

    static class myDbHelper extends SQLiteOpenHelper
    {
        private static final String DATABASE_NAME = "myDatabase";
// Database Name
        private static final String TABLE_NAME = "myTable";   // Table Name
        private static final int DATABASE_Version = 1;   // Database Version
private static final String UID="_id";     // Column I (Primary Key)           private
static final String NAME = "Name";    //Column II           private static final String
MyPASSWORD= "Password";    // Column III           private static final String
CREATE_TABLE = "CREATE TABLE
"+TABLE_NAME+
              " ("+UID+" INTEGER PRIMARY KEY AUTOINCREMENT, "+NAME+"
VARCHAR(255) ,"+
MyPASSWORD+" VARCHAR(225));";
        private static final String DROP_TABLE ="DROP TABLE IF EXISTS
"+TABLE_NAME;
        private Context context;

        public myDbHelper(Context context) {
            super(context,        DATABASE_NAME,          null, DATABASE_Version);
            this.context=context;
        }

        public void onCreate(SQLiteDatabase db) {

            try {
                db.execSQL(CREATE_TABLE);
            } catch (Exception e) {
                Message.message(context,""+e);
            }
        }

        @Override           public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
            try {
                Message.message(context,"OnUpgrade");
db.execSQL(DROP_TABLE);                  onCreate(db);
            }catch (Exception e) {
                Message.message(context,""+e);
            }
        }
```

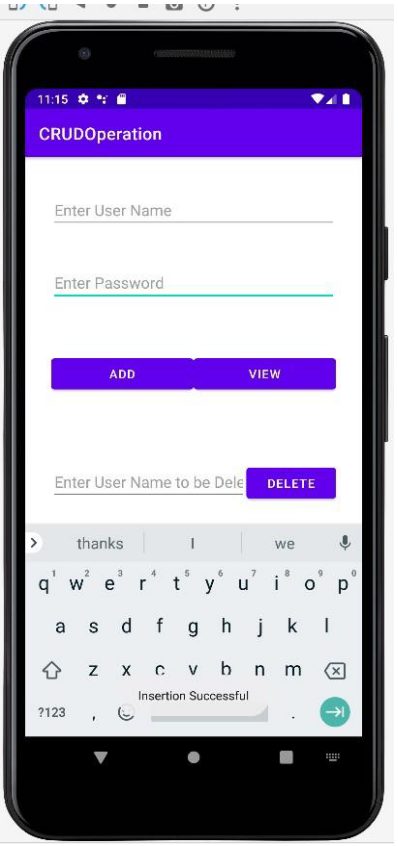
```
    }
}
```

Next step is to create another java class Message.class **Message.java:**

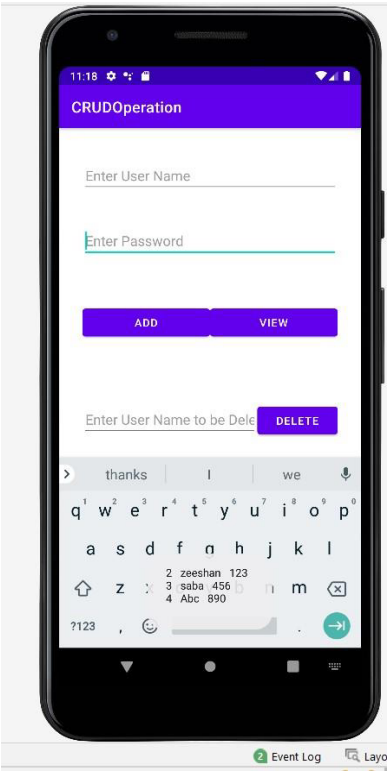
```
package com.example.crudoperation;

import android.content.Context; import
android.widget.Toast;

    public class Message {
        public static void message(Context context, String message) {
            Toast.makeText(context,                    message,
Toast.LENGTH_LONG).show();
        }
```

3.5.5 Output:

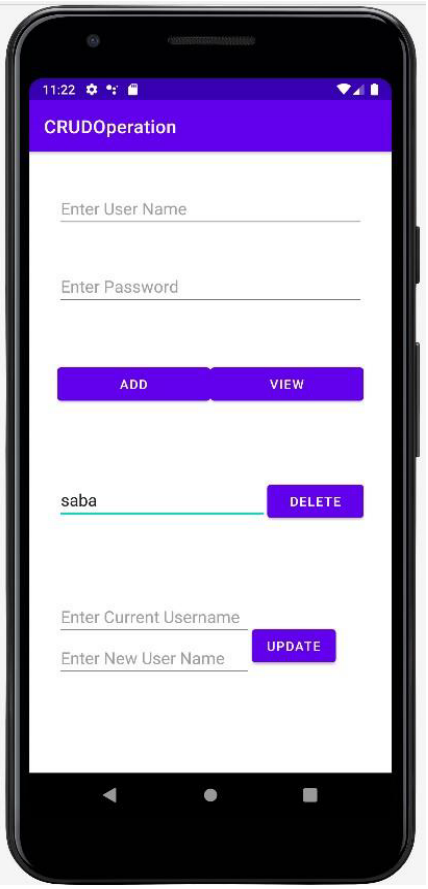




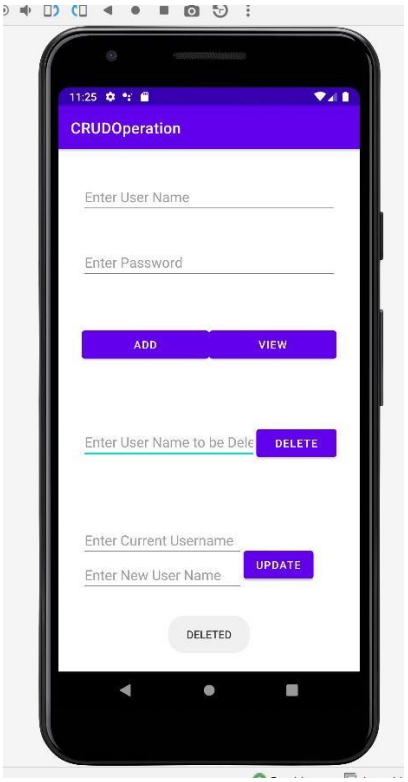
Above output is generated when Add button is clicked.



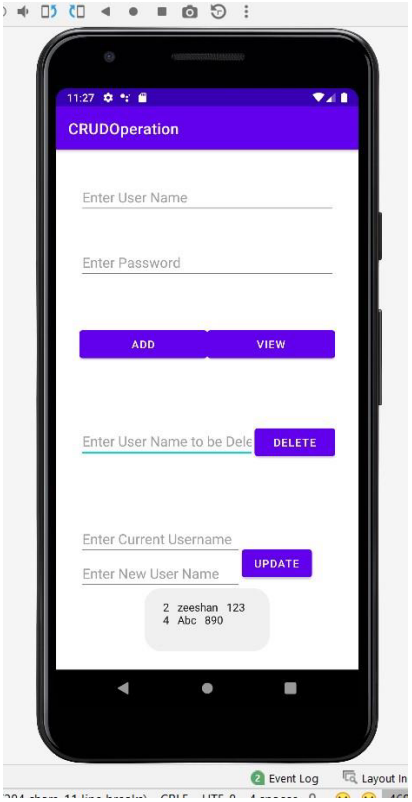
Above output is generated when View Button is pressed.



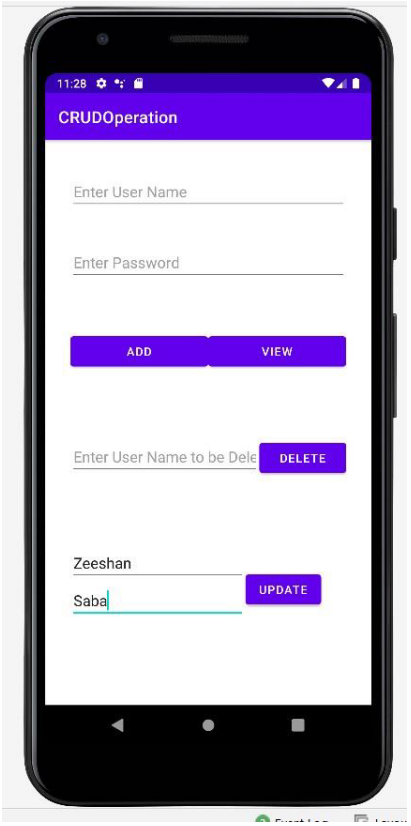
Entering user name saba to delete it.
User name is Deleted.

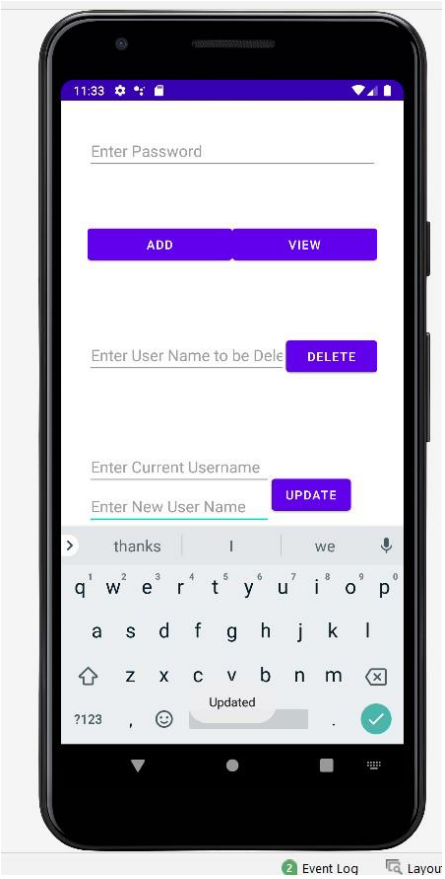


Now again View the data in the database.



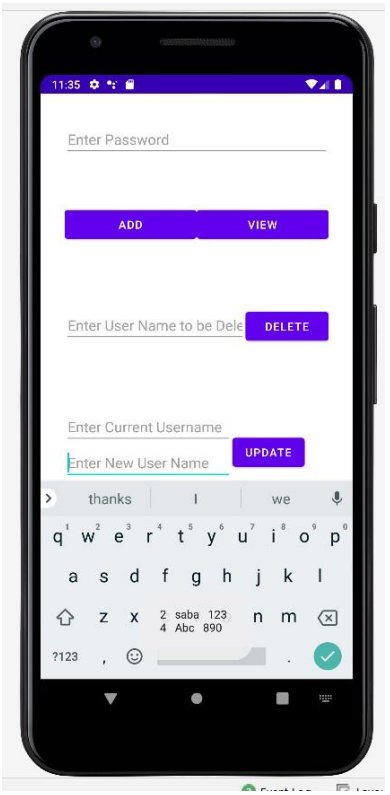
Next we will update the username Zeeshan with new user name saba


11:28
CRUDOperation
Enter User Name
Enter Password
ADD
VIEW
Enter User Name to be Dele
DELETE
Zeeshan
Saba
UPDATE



After clicking on update button

Again we will see the data by clicking on the update button


11:35
Enter Password
ADD
VIEW
Enter User Name to be Dele
DELETE
Enter Current Username
Enter New User Name
UPDATE
thanks
I
we

PRACTICAL 6

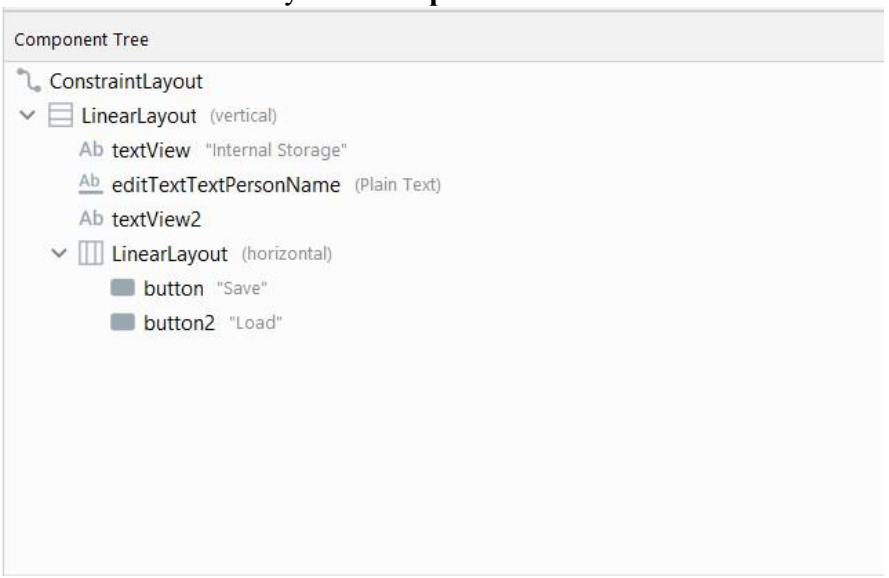
AIM: Android program using Shared Preferences, Internal and External Storage

Create a new project and name it InternalStorage.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.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="match_parent"        android:layout_height="match_parent"
android:orientation="vertical"        android:layout_marginRight="20sp"
android:layout_marginLeft="20sp">

      <TextView
         android:id="@+id/textView"            android:layout_width="match_parent"
android:layout_height="wrap_content"            android:text="Internal Storage"
android:layout_marginTop="20sp"            android:layout_marginBottom="20sp"
android:textSize="40sp"            android:textColor="@color/black"
android:textStyle="bold"
         />

      <EditText
         android:id="@+id/editTextTextPersonName"
android:layout_width="match_parent"            android:layout_height="wrap_content"
android:ems="10"            android:inputType="textPersonName"
android:hint="Enter Text"            android:layout_marginTop="20sp"
android:layout_marginBottom="20sp"            android:textSize="30sp"/>

      <TextView            android:id="@+id/textView2"
android:layout_width="match_parent"            android:layout_height="wrap_content"
android:hint="Read"            android:layout_marginTop="20sp"
android:layout_marginBottom="20sp"            android:textSize="30sp"/>

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

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

```
        <Button
            android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"               android:text="Load" />
      </LinearLayout>
   </LinearLayout>
</android.support.constraint.ConstraintLayout>
```
Component Tree:

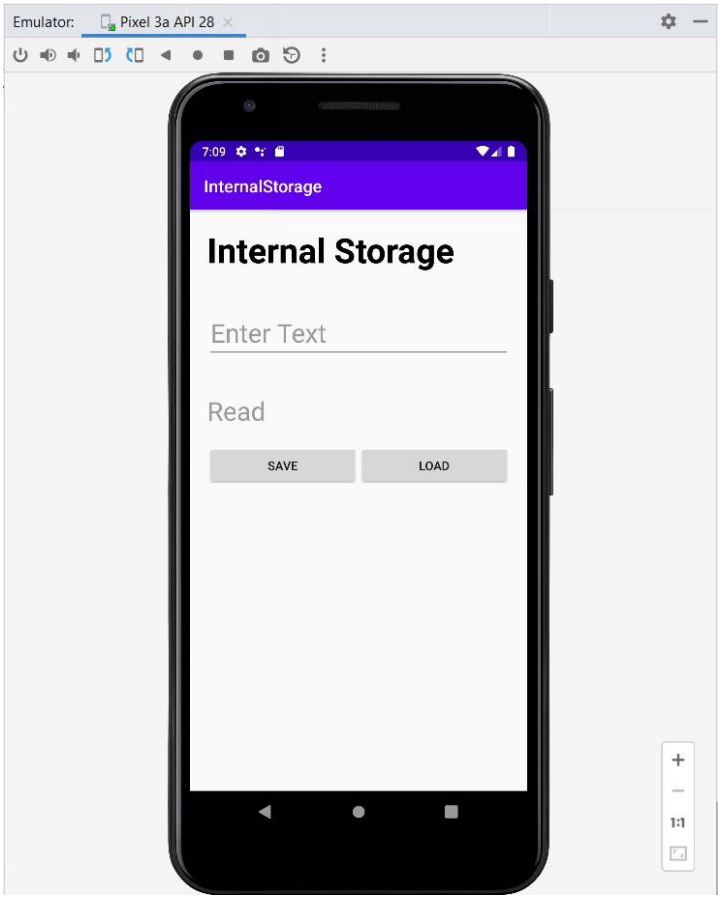


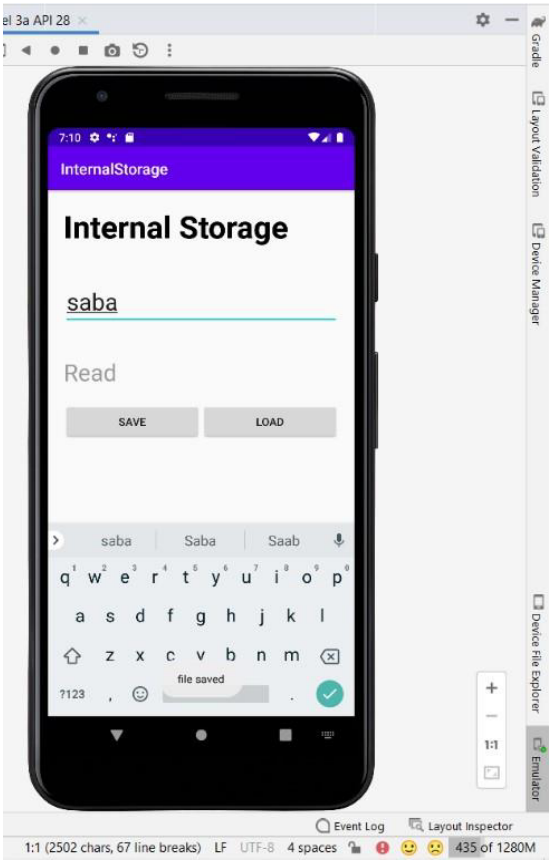
MainActivity.java

```
package com.example.internalstorage; import
android.support.v7.app.AppCompatActivity; import
android.os.Bundle; import android.view.View; import
android.widget.Button; import android.widget.EditText; import
android.widget.TextView; import android.widget.Toast; import
java.io.FileInputStream; import java.io.FileOutputStream;
public class MainActivity extends AppCompatActivity {
   Button b1,b2;
   TextView tv;
   EditText ed1;    String data;    private
String file = "mydata";
   @Override    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
b1=(Button)findViewById(R.id.button);
b2=(Button)findViewById(R.id.button2);
ed1=(EditText)findViewById(R.id.editTextTextPersonName);
tv=(TextView)findViewById(R.id.textView2);        b1.setOnClickListener(new
View.OnClickListener() {
```

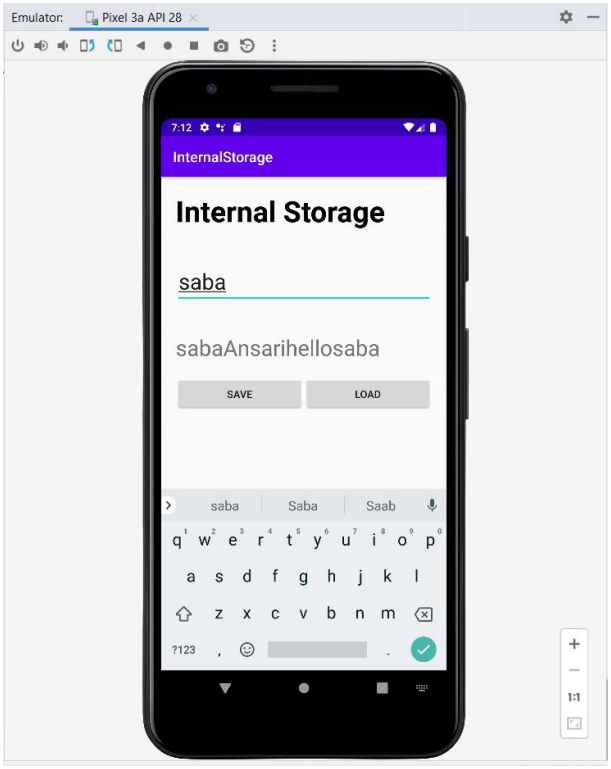
```
        @Override           public void onClick(View
v) {               data=ed1.getText().toString();
try {
               FileOutputStream                fOut                =
openFileOutput(file,MODE_APPEND);
fOut.write(data.getBytes());                fOut.close();
               Toast.makeText(getBaseContext(),"file
saved",Toast.LENGTH_SHORT).show();
            }
            catch (Exception e) {
               // TODO Auto-generated catch block                e.printStackTrace();
            }
         }
      });

      b2.setOnClickListener(new View.OnClickListener() {

         @Override           public void onClick(View
v) {
            try {
               FileInputStream fin = openFileInput(file);                int c = 0;
String t="";                while( (c= fin.read()) != -1) t += Character.toString((char) c);
tv.setText(t);
               Toast.makeText(getBaseContext(),"file
read",Toast.LENGTH_SHORT).show();
            }
            catch(Exception e){
            }
         }
      });   } }
```

1.5 Output:


Emulator:
Pixel 3a API 28
7:09
InternalStorage
Internal Storage
Enter Text
Read
SAVE
LOAD
1:1



Once we click on load button, all the data saved in internal strorage file will be displayed, As we are using Append mode ,it is showing previous data also.



To view the file in internal storage click on Device File Explorer->Data->Data->com.example.internalstorage->files->myData

Create a new android application using android studio and give name as ExternalStorage.

Providing Access Permission to External Storage:

To read and write data to external storage, the app required WRITE_EXTERNAL_STORAGE and READ_EXTERNAL_STORAGE system permission. These permissions are added to the AndroidManifest.xml file. Add these permissions just after the package name.

AndroidManifest.xml

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

    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
```

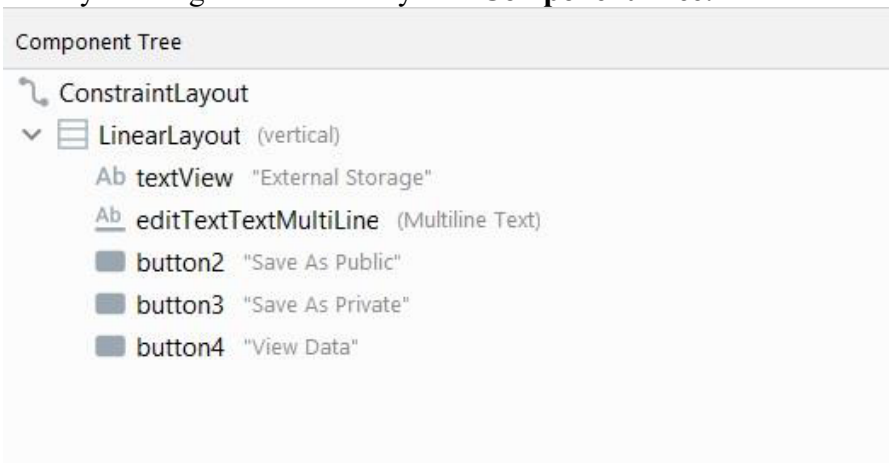
```
        android:allowBackup="true"        android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"        android:supportsRtl="true"
        android:theme="@style/Theme.ExternalStrorage">
        <activity
            android:name=".ViewData"
            android:exported="false"                                        />
        <activity
            android:name=".MainActivity"            android:exported="true">
            <intent-filter>
                <action      android:name="android.intent.action.MAIN"        />

                <category
android:name="android.intent.category.LAUNCHER"                        />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Once we have added the permission open activity_main.xml file from \res\layout folder path and write the code as shown below. activity_main.xml

```
<?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=".ViewData">

    <LinearLayout         android:layout_width="match_parent"
android:layout_height="match_parent"
        android:orientation="vertical"    android:layout_marginLeft="20sp"
android:layout_marginRight="20sp">

        <TextView             android:id="@+id/textView3"
android:layout_width="match_parent"
            android:layout_height="wrap_content"
                        android:text="Retreiving             Saved             Text             Data"
            android:layout_marginTop="30dp"
            android:layout_marginBottom="30dp"/>

        <TextView             android:id="@+id/textView4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TextView"
android:layout_marginTop="30dp"
```

```
        android:layout_marginBottom="30dp"/>

    <Button
        android:id="@+id/button"          android:layout_width="match_parent"
        android:layout_height="wrap_content"
                  android:text="Retreive              Private              data"
        android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"          android:onClick="showPrivateData"/>

    <Button
        android:id="@+id/button5"          android:layout_width="match_parent"
        android:layout_height="wrap_content"
                  android:text="Retreive              Public              Data"
        android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"
        android:onClick="showPublicData"/>

    <Button
        android:id="@+id/button6"
android:layout_width="match_parent"
android:layout_height="wrap_content"          android:text="Home"
android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"          android:onClick="back"/>
  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Component Tree:



Create a new Empty Activity:

Now we will create another empty activity by right clicking on App folder on the top->new->Activity->EmptyActivity and name it as ViewData.

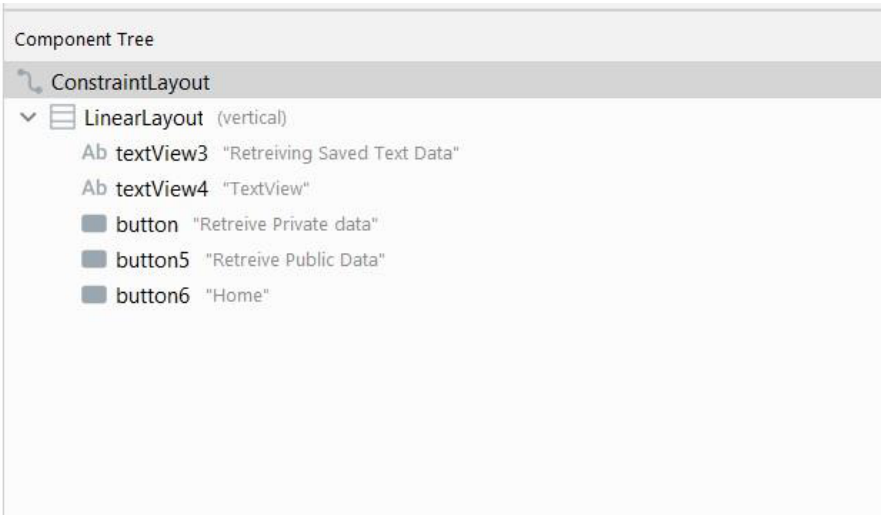
We use this activity to display the saved data from the external storage.

Code for Activity_View_Data.xml

```
<?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=".ViewData">

   <LinearLayout
      android:layout_width="match_parent"        android:layout_height="match_parent"
      android:orientation="vertical" android:layout_marginLeft="20sp"
android:layout_marginRight="20sp">
      <TextView            android:id="@+id/textView3"
android:layout_width="match_parent"            android:layout_height="wrap_content"
android:text="Retreiving Saved Text Data"            android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"/>

      <TextView            android:id="@+id/textView4"
android:layout_width="match_parent"            android:layout_height="wrap_content"
android:text="TextView"            android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"/>

      <Button            android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"            android:text="Retreive Private data"
android:layout_marginTop="30dp"            android:layout_marginBottom="30dp"
android:onClick="showPrivateData"/>

      <Button            android:id="@+id/button5"
android:layout_width="match_parent"
android:layout_height="wrap_content"            android:text="Retreive Public Data"
android:layout_marginTop="30dp"            android:layout_marginBottom="30dp"
android:onClick="showPublicData"/>

      <Button
         android:id="@+id/button6"
android:layout_width="match_parent"
android:layout_height="wrap_content"            android:text="Home"
android:layout_marginTop="30dp"
android:layout_marginBottom="30dp"            android:onClick="back"/>
   </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Component Tree:



MainActivity.java:

```
package com.example.externalstrorage;

import static android.os.Environment.DIRECTORY_DOWNLOADS; import
androidx.appcompat.app.AppCompatActivity;

import android.Manifest; import android.content.Intent;
import android.os.Bundle; import
android.os.Environment; import android.view.View;
import android.widget.EditText; import
android.widget.Toast; import
androidx.core.app.ActivityCompat; import java.io.File;
import java.io.FileOutputStream; import
java.io.IOException;

public class MainActivity extends AppCompatActivity {     private int
EXTERNAL_STORAGE_PERMISSION_CODE = 23;
   EditText editText;     @Override     protected void onCreate(Bundle
savedInstanceState) {         super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);         editText = (EditText)
findViewById(R.id.editTextTextMultiLine);


   }
   public void savePublicly(View view) {
      // Requesting Permission to access External Storage
                  ActivityCompat.requestPermissions(this,                    new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
            EXTERNAL_STORAGE_PERMISSION_CODE);
      String editTextData = editText.getText().toString();
      // getExternalStoragePublicDirectory() represents root of external storage, we are
using DOWNLOADS
```

```
        // We can use following directories: MUSIC, PODCASTS, ALARMS,
RINGTONES, NOTIFICATIONS, PICTURES, MOVIES
        File                                folder                                =
Environment.getExternalStoragePublicDirectory(DIRECTORY_DOWNL OADS);

        // Storing the data in file with name as sabaext.txt        File file =
new File(folder, "sabaext.txt");        writeTextData(file, editTextData);
editText.setText("");
    }
    public void savePrivately(View view) {
        String editTextData = editText.getText().toString();

        // Creating folder with name ExternalStorage
        File folder = getExternalFilesDir("ExternalStorage");

        // Creating file with name xyz.txt        File file =
new File(folder, "xyz.txt");        writeTextData(file,
editTextData);        editText.setText("");
    }
    public void viewInformation(View view) {
        // Creating an intent to start a new activity
        Intent intent = new Intent(MainActivity.this, ViewData.class);
startActivity(intent);
    }
    private void writeTextData(File file, String data) {
FileOutputStream fileOutputStream = null;        try {
            fileOutputStream = new FileOutputStream(file);
fileOutputStream.write(data.getBytes());            Toast.makeText(this,        "Done"
+        file.getAbsolutePath(),
Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
e.printStackTrace();        } finally {            if
(fileOutputStream != null)
            {               try {
fileOutputStream.close();               } catch
(IOException e) {                   e.printStackTrace();
                }
            }
        }
    }
}
```

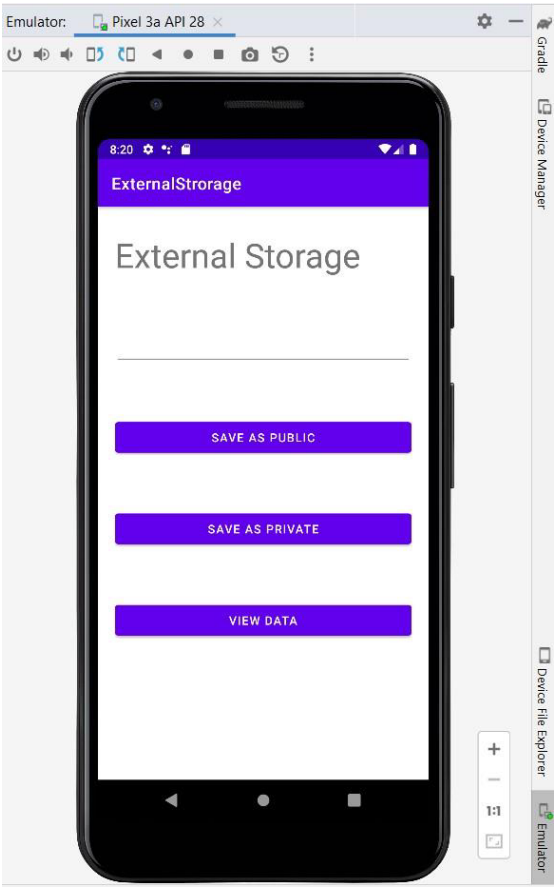
ViewData.java:

```
package com.example.externalstrorage;

import static android.os.Environment.DIRECTORY_DOWNLOADS;
```

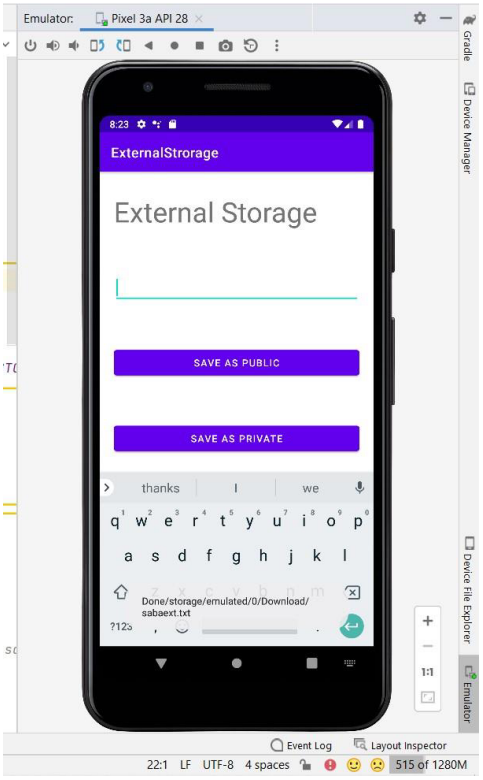
```
import androidx.appcompat.app.AppCompatActivity; import
android.content.Intent; import android.os.Bundle; import
android.os.Environment; import android.view.View; import
android.widget.TextView; import java.io.File; import
java.io.FileInputStream; import java.io.IOException;

public class ViewData extends AppCompatActivity {
   TextView textView;
   @Override     protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_view_data);        textView =
(TextView) findViewById(R.id.textView4);

   }
   public void showPublicData(View view) {
      // Accessing the saved data from the downloads folder
      File                                folder                                =
Environment.getExternalStoragePublicDirectory(DIRECTORY_DOWNL OADS);

      // sabaext represent the file data that is saved publicly
      File file = new File(folder, "sabaext.txt");        String
data = getdata(file);        if (data != null) {
textView.setText(data);
      } else {           textView.setText("No Data Found");
      }
   }
   public void showPrivateData(View view) {

      // ExternalStorage represent the folder name to access privately saved data
      File folder = getExternalFilesDir("ExternalStorage");

      // xyz.txt is the file that is saved privately
      File file = new File(folder, "xyz.txt");        String
data = getdata(file);        if (data != null) {
textView.setText(data);
      } else {
         textView.setText("No Data Found");
      }
   }
   public void back(View view) {
      Intent intent = new Intent(ViewData.this, MainActivity.class);
startActivity(intent);
   }
   // getdata() is the method which reads the data     // the data that is
saved in byte format in the file     private String getdata(File myfile) {
FileInputStream fileInputStream = null;        try {
```

```
            fileInputStream = new FileInputStream(myfile);            int i = -1;
            StringBuffer buffer = new StringBuffer();            while ((i =
fileInputStream.read()) != -1) {                buffer.append((char) i);
            }            return buffer.toString();         } catch
(Exception e) {            e.printStackTrace();         } finally
{            if (fileInputStream != null) {                try {
fileInputStream.close();                } catch (IOException e)
{                    e.printStackTrace();
                }
            }        }        return null;
    } }
```

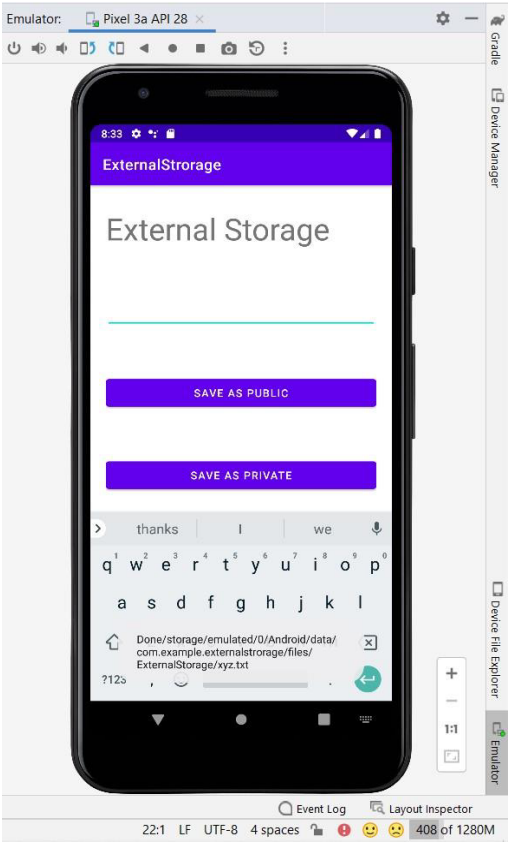
3.2.5 Output:



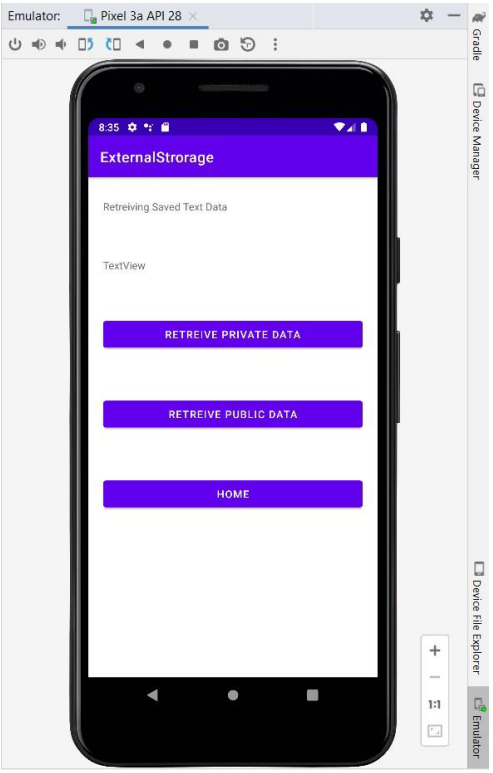
when Clicked on Save as Public button:



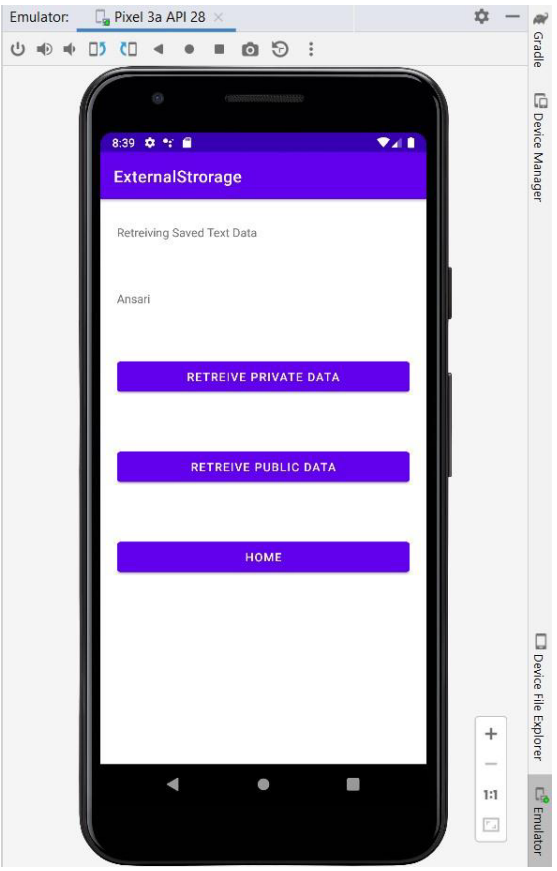
When Clicked on Save As Private Button:



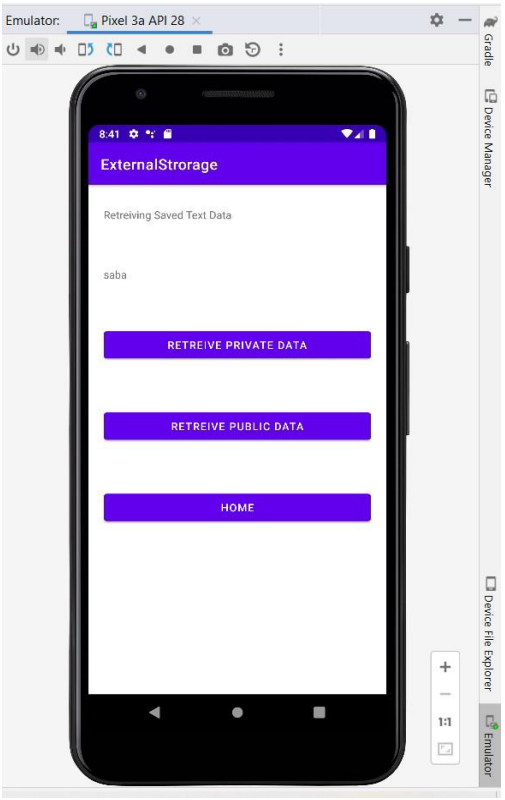
When clicked on View Data button is clicked another activity that view data activity is started.



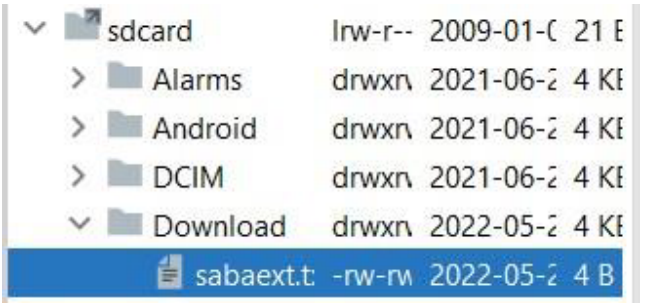
When clicked on retrieve Private data Button:



When we clicked on Retrieve public data button:



To View data saved as publicly click on Device File Explorer->SD Card>Downloads and you will see sabaext.txt file.



To View data saved as privately click on Device File Explorer->Storage>Android->Data->com.examples.externalstorage(your Application folder)
->Files. In files folder you can see folder is created

You can save something in the sharedpreferences by using SharedPreferences.Editor class.

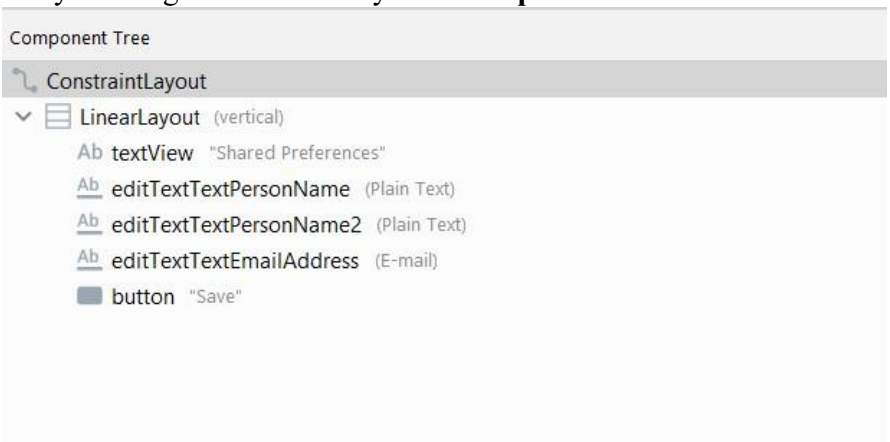
3.3.4 Program:

Create a new android application using android studio and give name as sharedpreferences. Once we create an application, open activity_main.xml file from \res\layout folder path and write the code as shown below.

Activity_main.xml:

```
<?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="match_parent"
android:layout_height="match_parent"        android:orientation="vertical"
android:layout_marginLeft="30sp"        android:layout_marginRight="30sp">
<TextView
         android:id="@+id/textView"            android:layout_width="match_parent"
android:layout_height="wrap_content"           android:text="Shared Preferences"
android:textSize="40sp"           android:textStyle="bold"
android:textColor="@color/black"/>

      <EditText           android:id="@+id/editTextTextPersonName"
android:layout_width="match_parent"           android:layout_height="wrap_content"
android:layout_marginTop="20dp"           android:layout_marginBottom="20dp"
android:ems="10"           android:hint="Enter your Name"
android:inputType="textPersonName" />

      <EditText           android:id="@+id/editTextTextPersonName2"
android:layout_width="match_parent"           android:layout_height="wrap_content"
android:ems="10"           android:inputType="textPersonName"
android:hint="Enter your Age"           android:layout_marginTop="20dp"
android:layout_marginBottom="20dp"/>

      <EditText           android:id="@+id/editTextTextEmailAddress"
android:layout_width="match_parent"           android:layout_height="wrap_content"
android:ems="10"           android:inputType="textEmailAddress"
android:hint="Enter Your Email"           android:layout_marginTop="20dp"
android:layout_marginBottom="20dp"/>

      <Button           android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Save"           android:layout_marginTop="20dp"
android:layout_marginBottom="20dp"/>
   </LinearLayout>
```

</androidx.constraintlayout.widget.ConstraintLayout> **Component tree:**



MainActivity.java: package com.example.sharedpreferences; import androidx.appcompat.app.AppCompatActivity; import android.content.Context; import android.os.Bundle; import android.content.SharedPreferences; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;

public class MainActivity extends AppCompatActivity { private EditText name, age,email; Button b1; public static final String MyPREFERENCES = "MyPrefsData" ; public static final String Name = "nameKey"; public static final String Age = "ageKey"; public static final String Email = "emailKey"; SharedPreferences sharedpreferences; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); name =(EditText) findViewById(R.id.editTextTextPersonName); age = (EditText)findViewById(R.id.editTextTextPersonName2); email=(EditText) findViewById(R.id.editTextTextEmailAddress); b1=(Button) findViewById(R.id.button);
sharedpreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);

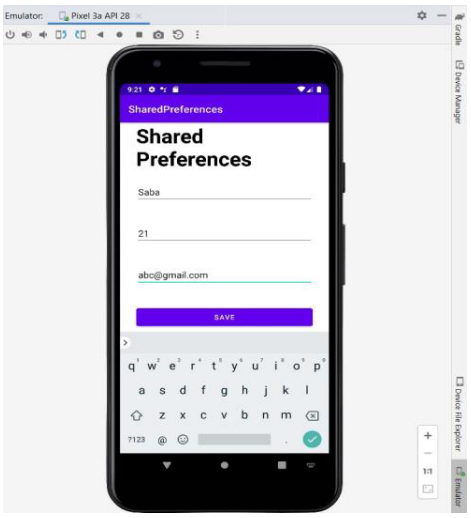
b1.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { String n = name.getText().toString();
String a = age.getText().toString();
String e = email.getText().toString();

SharedPreferences.Editor editor = sharedpreferences.edit(); editor.putString(Name, n); editor.putString(Age, a); editor.putString(Email, e); editor.commit(); Toast.makeText(MainActivity.this,"Thanks",Toast.LENGTH_LONG).sho w();

```
        }
    });
} }
```

3.3.5 Output:

Step1: User enters the data in text field



Now when you press save button, the text will be saved in the shared preferences. Now press back button and exit the application.

To Check Shared preferences, click on

Device File Explorer->Data->Data->Com.example.sharedpreferences(your application folder)-> shared_prefs->MyPrefsData.xml

Content of MyPrefsData.xml:

MyPrefsData.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="ageKey">21</string>
    <string name="nameKey">saba</string>
    <string name="emailKey">abc@gmail.com</string>
</map>
```

PRACTICAL 7

AIM: Android program to work with Google Maps and locations

Write a program to find your location in the Map package com.abhiandroid.GoogleMaps.googlemaps; import android.Manifest; import android.content.Context; import android.content.pm.PackageManager; import android.location.Address; import android.location.Criteria;

114114

```
import android.location.Geocoder; import
android.location.Location; import
android.location.LocationManager; import
android.os.Build; import android.os.Bundle;
import android.support.v4.app.ActivityCompat; import
android.support.v4.app.FragmentActivity; import
android.support.v4.content.ContextCompat; import
android.widget.Toast;
import com.google.android.gms.common.ConnectionResult; import
com.google.android.gms.common.api.GoogleApiClient; import
com.google.android.gms.location.LocationListener; import
com.google.android.gms.location.LocationRequest; import
com.google.android.gms.location.LocationServices; import
com.google.android.gms.maps.CameraUpdateFactory; import
com.google.android.gms.maps.GoogleMap; import
com.google.android.gms.maps.OnMapReadyCallback; import
com.google.android.gms.maps.SupportMapFragment; import
com.google.android.gms.maps.model.BitmapDescriptorFactory; import
com.google.android.gms.maps.model.LatLng; import
com.google.android.gms.maps.model.Marker; import
com.google.android.gms.maps.model.MarkerOptions; import
com.abhiandroid.GoogleMaps.googlemaps.R;

import java.io.IOException; import
java.util.List; import java.util.Locale;
public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback,
    GoogleApiClient.ConnectionCallbacks,
GoogleApiClient.OnConnectionFailedListener, LocationListener {
        public static final int
        MY_PERMISSIONS_REQUEST_LOCATION =
        99;
        GoogleApiClient mGoogleApiClient;
        Location mLastLocation;
        Marker mCurrLocationMarker;
        LocationRequest mLocationRequest;
        private GoogleMap mMap;
        @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_maps);

    if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
      checkLocationPermission();
    }
    SupportMapFragment mapFragment =
(SupportMapFragment)          getSupportFragmentManager()
                .findFragmentById(R.id.map);

    mapFragment.getMapAsync(this);
  }
  @Override
  public void onMapReady(GoogleMap googleMap)
{     mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
mMap.getUiSettings().setZoomControlsEnabled(true);
mMap.getUiSettings().setZoomGesturesEnabled(true);
116116
    mMap.getUiSettings().setCompassEnabled(true);
    //Initialize Google Play Services
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
      if (ContextCompat.checkSelfPermission(this,
          Manifest.permission.ACCESS_FINE_LOCATION)          ==
PackageManager.PERMISSION_GRANTED) {
buildGoogleApiClient();          mMap.setMyLocationEnabled(true);
      }
    } else {
      buildGoogleApiClient();       mMap.setMyLocationEnabled(true);
    }
  }
  protected synchronized void buildGoogleApiClient() {     mGoogleApiClient = new
GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
    mGoogleApiClient.connect();
  }
  @Override
  public void onConnected(Bundle bundle) {     mLocationRequest = new
LocationRequest();     mLocationRequest.setInterval(1000);
mLocationRequest.setFastestInterval(1000);
mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED
```

```
_POWER_ACCURACY);
                    if (ContextCompat.checkSelfPermission(this,
                            Manifest.permission.ACCESS_FINE_LOCATION)
                            == PackageManager.PERMISSION_GRANTED) {

                    LocationServices.FusedLocationApi.requestLocationUpdates(mG
                    oogleAp iClient,
                            mLocationRequest, this);
                    }
                  }
                  @Override
                  public void onConnectionSuspended(int i) {
                  }
                  @Override
                  public void onLocationChanged(Location
                  location) {        mLastLocation = location;
                  if (mCurrLocationMarker != null) {
                  mCurrLocationMarker.remove();
                    }
                  //Showing Current Location Marker on Map
                    LatLng    latLng =        new
                  LatLng(location.getLatitude(), location.getLongitude());
                    MarkerOptions markerOptions = new MarkerOptions();
                  markerOptions.position(latLng);
                    LocationManager locationManager = (LocationManager)
                  getSystemService(Context.LOCATION_SERVICE);
                    String provider = locationManager.getBestProvider(new
                  Criteria(), true);
                    if (ActivityCompat.checkSelfPermission(this,
                   Manifest.permission.ACCESS_FINE_LOCATION)              !=
                  PackageManager.PERMISSION_GRANTED &&
                            ActivityCompat.checkSelfPermission(this,
                  Manifest.permission.ACCESS_COARSE_LOCATION)
118118
              != PackageManager.PERMISSION_GRANTED) {          return;
    }
    Location                          locations                          =
locationManager.getLastKnownLocation(provider);
    List<String> providerList = locationManager.getAllProviders();
    if (null != locations && null != providerList && providerList.size() >
0) {
      double longitude = locations.getLongitude();          double
latitude = locations.getLatitude();
      Geocoder geocoder = new Geocoder(getApplicationContext(),
Locale.getDefault());          try {
        List<Address>                  listAddresses                  =
```

```
geocoder.getFromLocation(latitude,
                    longitude, 1);
             if (null != listAddresses && listAddresses.size() > 0) {                    String
state = listAddresses.get(0).getAdminArea();
                  String country = listAddresses.get(0).getCountryName();                    String
subLocality = listAddresses.get(0).getSubLocality();
                  markerOptions.title("" + latLng + "," + subLocality + "," + state
                        + "," + country);
               }
            } catch (IOException e) {                e.printStackTrace();
            }
        }

markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDesc
riptorFactory.HUE_BLUE));
       mCurrLocationMarker = mMap.addMarker(markerOptions);
mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

                              mMap.animateCamera(CameraUpdateFactory.zoomTo(11));
                              if (mGoogleApiClient != null) {

                              LocationServices.FusedLocationApi.removeLocationUpdates(mG
                              oogleAp iClient,
                                          this);
                                 }
                              }
                              @Override
                              public void onConnectionFailed(ConnectionResult
                              connectionResult) {
                              }
                              public boolean
                              checkLocationPermission() {        if
                              (ContextCompat.checkSelfPermission(this
                              ,
                                          Manifest.permission.ACCESS_FINE_LOCATION)
                                          != PackageManager.PERMISSION_GRANTED) {

                                     if
                              (ActivityCompat.shouldShowRequestPermissionRationale(this,

                              Manifest.permission.ACCESS_FINE_LOCATION)) {
                              ActivityCompat.requestPermissions(this,
                              new
                              String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                                                MY_PERMISSIONS_REQUEST_LOCATION);
                                     } else {
```

```
                        ActivityCompat.requestPermissions(this,
                        new
                        String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                                        MY_PERMISSIONS_REQUEST_LOCATION);
                            }
                            return false;
                        }
else { 120120
            return true;
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode,
                                String permissions[], int[] grantResults) {         switch
(requestCode) {
            case MY_PERMISSIONS_REQUEST_LOCATION: {
                if (grantResults.length > 0
                        &&                          grantResults[0]                          ==
PackageManager.PERMISSION_GRANTED) {
                    if (ContextCompat.checkSelfPermission(this,
                            Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {                        if
(mGoogleApiClient == null) {                        buildGoogleApiClient();
                        }
                        mMap.setMyLocationEnabled(true);
                    }
                } else {
                    Toast.makeText(this, "permission denied",
                            Toast.LENGTH_LONG).show();
                }               return;
            }
        }
    }
}
```

Output:

Now run the App. If you are connected to internet and provide access to your location then in Map you will see your current location.

Step 1: Create a new project in Android Studio, go to File ⇒ New Project and fill all required details to create a new project.

Step 2: Add the following code to res/layout/activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
android:layout_marginTop="20dp"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Get Current Location and City
Name"    android:textAlignment="center"
android:layout_centerHorizontal="true"
android:textSize="20sp" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/textView"
android:layout_centerInParent="true"
android:textSize="16sp"
android:textStyle="bold"/>
</RelativeLayout>
```

122122

Step 3: Add the following dependency in Gradle

```
implementation 'com.google.android.gms:play-services-location:17.0.0'
```

Step 4: Add the following code to src/MainActivity.java

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.os.Handler;
import android.os.ResultReceiver;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
public class MainActivity extends AppCompatActivity {
    private FusedLocationProviderClient fusedLocationClient;
    private static final int LOCATION_PERMISSION_REQUEST_CODE = 2;
    private LocationAddressResultReceiver addressResultReceiver;
    private TextView currentAddTv;
    private Location currentLocation;
                private LocationCallback locationCallback;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
      addressResultReceiver = new
  LocationAddressResultReceiver(new Handler());
      currentAddTv = findViewById(R.id.textView);
 fusedLocationClient                                                    =
  LocationServices.getFusedLocationProviderClient(this);
  locationCallback = new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult
  locationResult) {          currentLocation =
  locationResult.getLocations().get(0);          getAddress();
        }
      };
      startLocationUpdates();
    }

  @SuppressWarnings("MissingPermission")
  private void startLocationUpdates() {
      if (ContextCompat.checkSelfPermission(this,
  Manifest.permission.ACCESS_FINE_LOCATION) !=
      PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new

  String[]{Manifest.permission.ACCESS_FINE_LOCATION},
        LOCATION_PERMISSION_REQUEST_CODE);

  }
  e
  l
  s
  e
  {
        LocationRequest locationRequest = new LocationRequest();
124124
    locationRequest.setInterval(2000);       locationRequest.setFastestInterval(1000);

locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCUR ACY);
    fusedLocationClient.requestLocationUpdates(locationRequest, locationCallback,
null);
   }
  }
  @SuppressWarnings("MissingPermission")   private
void getAddress() {     if (!Geocoder.isPresent()) {
```

```
        Toast.makeText(MainActivity.this, "Can't find current address, ",
Toast.LENGTH_SHORT).show();
        return;
      }
      Intent intent = new Intent(this, GetAddressIntentService.class);
intent.putExtra("add_receiver", addressResultReceiver);
intent.putExtra("add_location", currentLocation);       startService(intent);
   }
   @Override
   public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults)
{
   if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
      if      (grantResults.length      >      0      &&      grantResults[0]      ==
PackageManager.PERMISSION_GRANTED) {
        startLocationUpdates();
      }       else {
Toast.makeText(this,
"Location permission
not granted, " + "restart
the app if you want the
                                    feature", Toast.LENGTH_SHORT).show();
                                }

                            }

                            }

                            private class LocationAddressResultReceiver extends
                            ResultReceiver {    LocationAddressResultReceiver(Handler
                            handler) {       super(handler);
                              }
                              @Override
                              protected void onReceiveResult(int resultCode, Bundle
                            resultData) {       if (resultCode == 0) {
                                    Log.d("Address", "Location null retrying");
                            getAddress();
                                }
                                if (resultCode == 1) {
                                  Toast.makeText(MainActivity.this,       "Address         not
                            found,  ", Toast.LENGTH_SHORT).show();
                                }
                                String currentAdd = resultData.getString("address_result");
                            showResults(currentAdd);
                                }
                              }
```

```
    private void showResults(String currentAdd) {
  currentAddTv.setText(currentAdd);
    }
    @Override
    protected void onResume() {
  super.onResume();
126126
    startLocationUpdates();
  }
  @Override    protected void onPause() {
super.onPause();
    fusedLocationClient.removeLocationUpdates(locationCallback);
  }
}
```

Step 5: Create a new java class GetaddressIntentService.java and add the following code

```
package app.com.sample; import android.app.IntentService; import
android.content.Intent; import android.location.Address; import
android.location.Geocoder; import android.location.Location; import android.os.Bundle;
import android.os.ResultReceiver; import android.util.Log; import java.util.List; import
java.util.Locale; import java.util.Objects; import androidx.annotation.Nullable;
public class GetAddressIntentService extends IntentService {    private static final
String IDENTIFIER = "GetAddressIntentService";    private ResultReceiver
addressResultReceiver;    public GetAddressIntentService() {
super(IDENTIFIER);
  }
  @Override
    protected void onHandleIntent(@Nullable Intent intent)
  {      String msg;
addressResultReceiver                                                    =
  Objects.requireNonNull(intent).getParcelableExtra("add_recei
  ver");      if (addressResultReceiver == null) {
        Log.e("GetAddressIntentService", "No receiver, not
  processing the request further");        return;
    }
    Location location =
  intent.getParcelableExtra("add_location");      if (location
  == null) {
        msg = "No location, can't go further without
  location";        sendResultsToReceiver(0, msg);
  return;
    }
    Geocoder geocoder = new Geocoder(this,
  Locale.getDefault());    List<Address> addresses = null;
  try {
    addresses = geocoder.getFromLocation(location.getLatitude(),
  location.getLongitude(), 1);
```

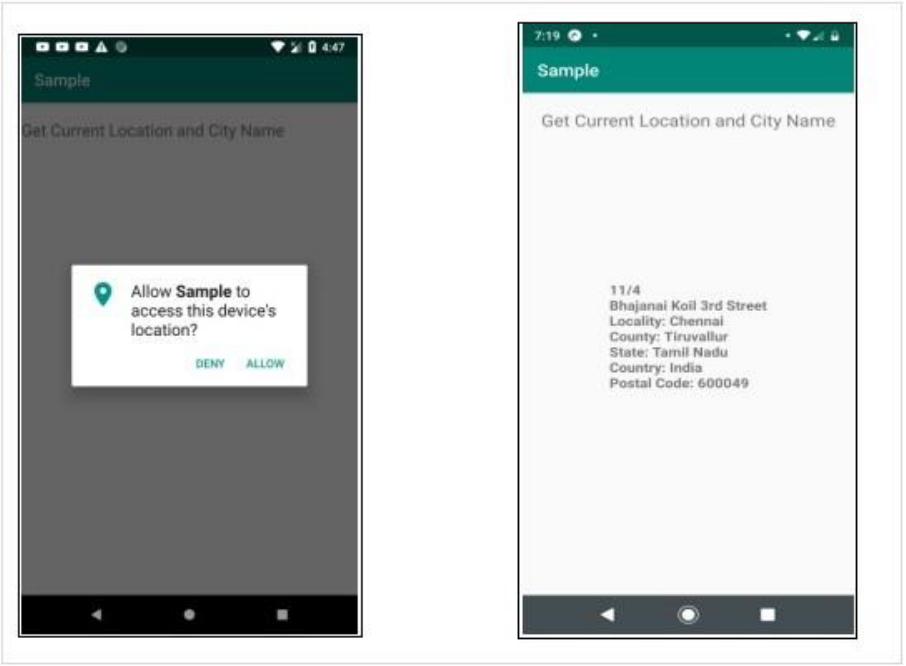
```
  }
  catch (Exception ioException) {
    Log.e("", "Error in getting address for the location");
  }
  if (addresses == null || addresses.size() ==
0) {      msg = "No address found for the
location";      sendResultsToReceiver(1,
msg);

}

e
l
s
e

{

    Address address = addresses.get(0);
128128
    String      addressDetails      =      address.getFeatureName()      +      "\n"
+ address.getThoroughfare() + "\n" +
    "Locality: " + address.getLocality() + "\n" + "County: " +
address.getSubAdminArea() + "\n" +
    "State: " + address.getAdminArea() + "\n" + "Country: " +
address.getCountryName() + "\n" +
    "Postal Code: " + address.getPostalCode() + "\n";      sendResultsToReceiver(2,
addressDetails);
  } }
private void sendResultsToReceiver(int resultCode, String message) {    Bundle
bundle = new Bundle();    bundle.putString("address_result", message);
addressResultReceiver.send(resultCode, bundle);
  }
}
```

Step 6: Add the following code to androidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="app.com.sample">
  <application
    android:allowBackup="true"      android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"      android:supportsRtl="true"
android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
```

```
            <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
   <service android:name=".GetAddressIntentService" />
</application>
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION
" />
<uses-permission
android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATI
ON" />
</manifest>
```

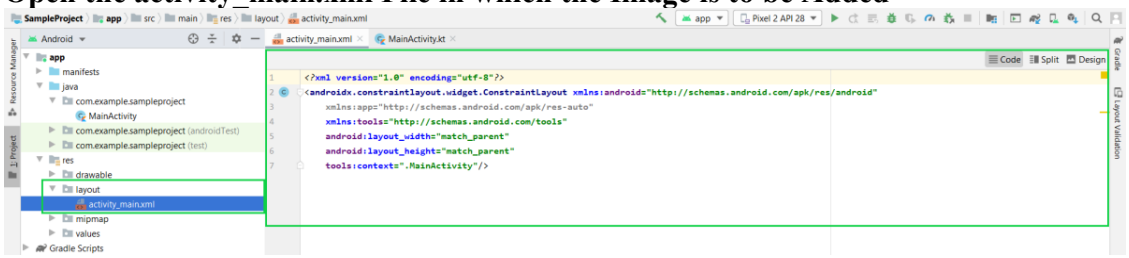
To run the app from the android studio, open one of your project's activity files and click the RunPlay Icon icon from the toolbar. Select your mobile device as an option and then check your mobile device which will display your default screen**:**



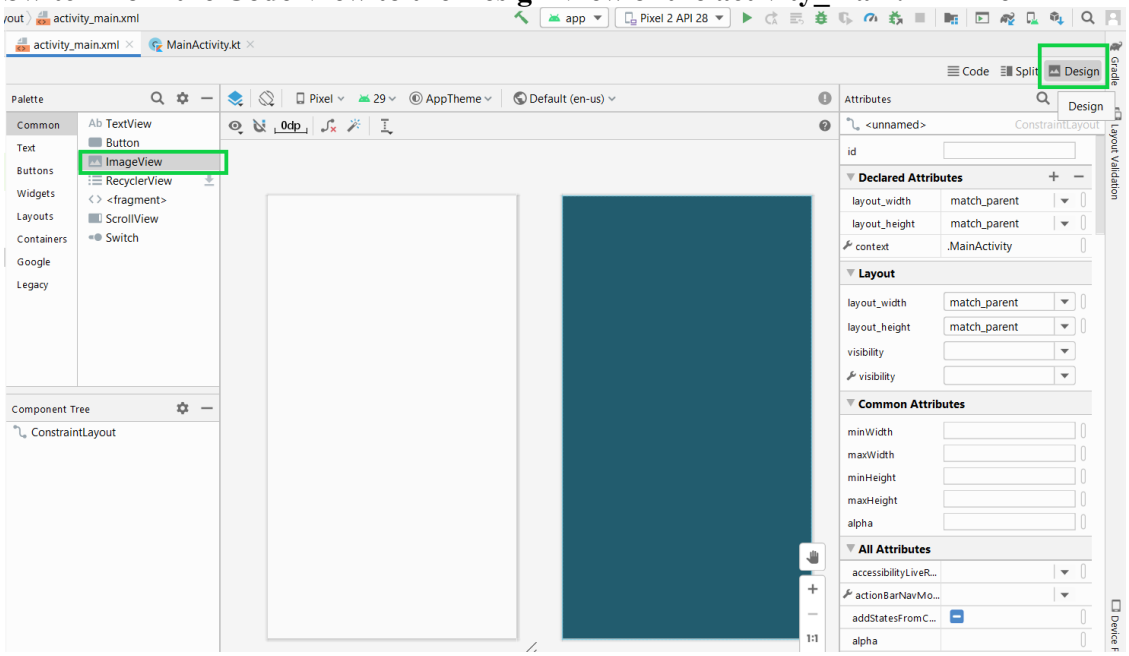
PRACTICAL 8

AIM:Android program to work with images and videos

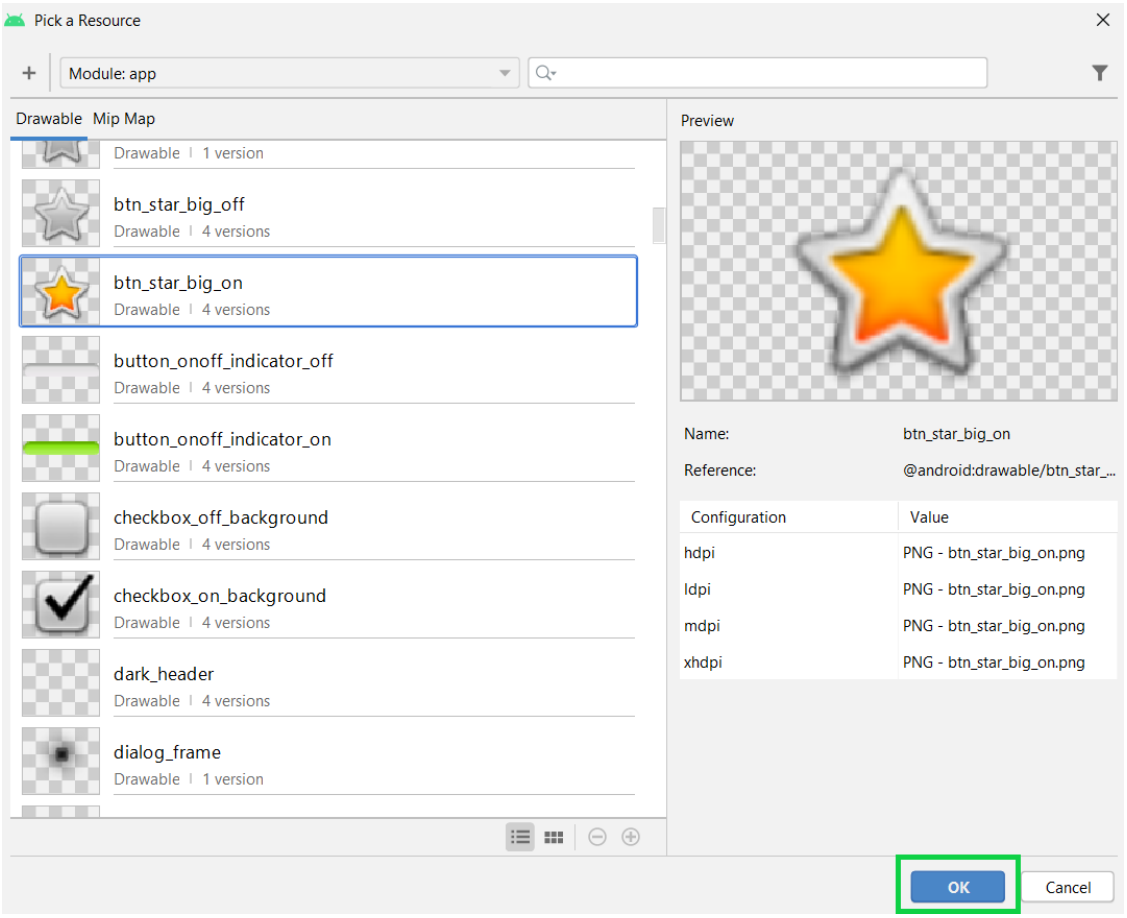
Open the activity_main.xml File in which the Image is to be Added



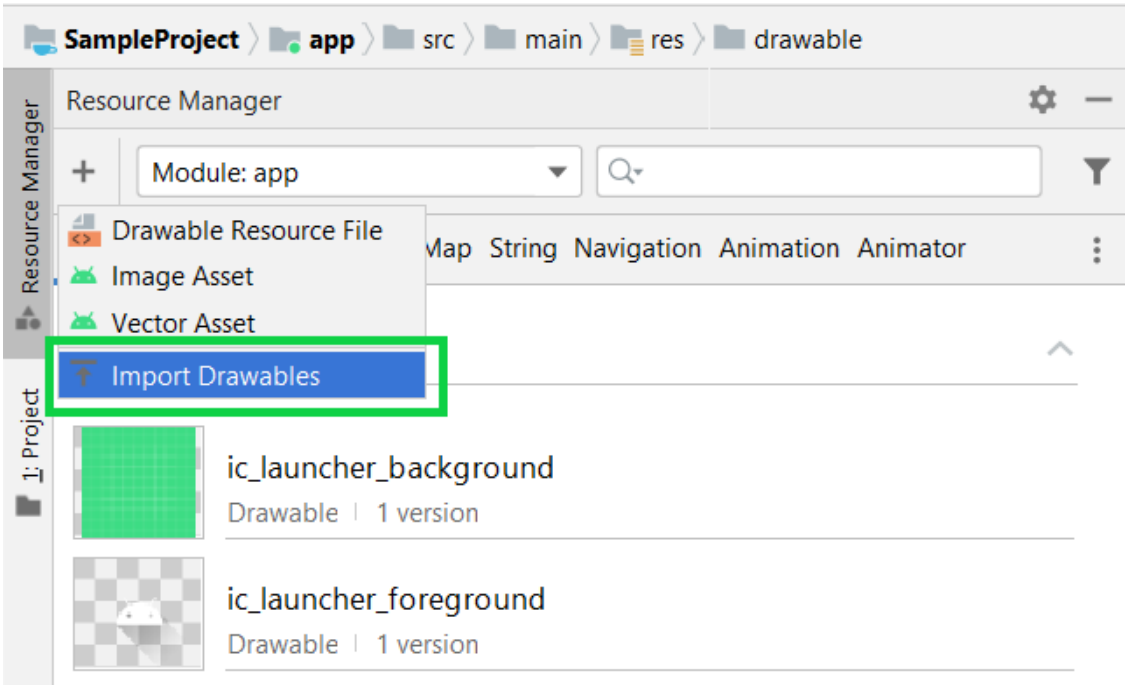
Switch from the Code View to the Design View of the activity_main.xml File



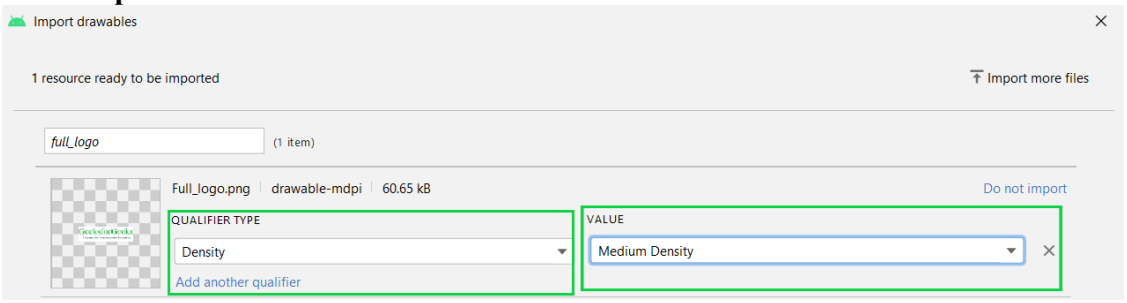
For adding an image from Android Studio, Drag the ImageView widget to the activity area of the application, a pop-up dialogue box will open choose from the wide range of drawable resources and click "OK".



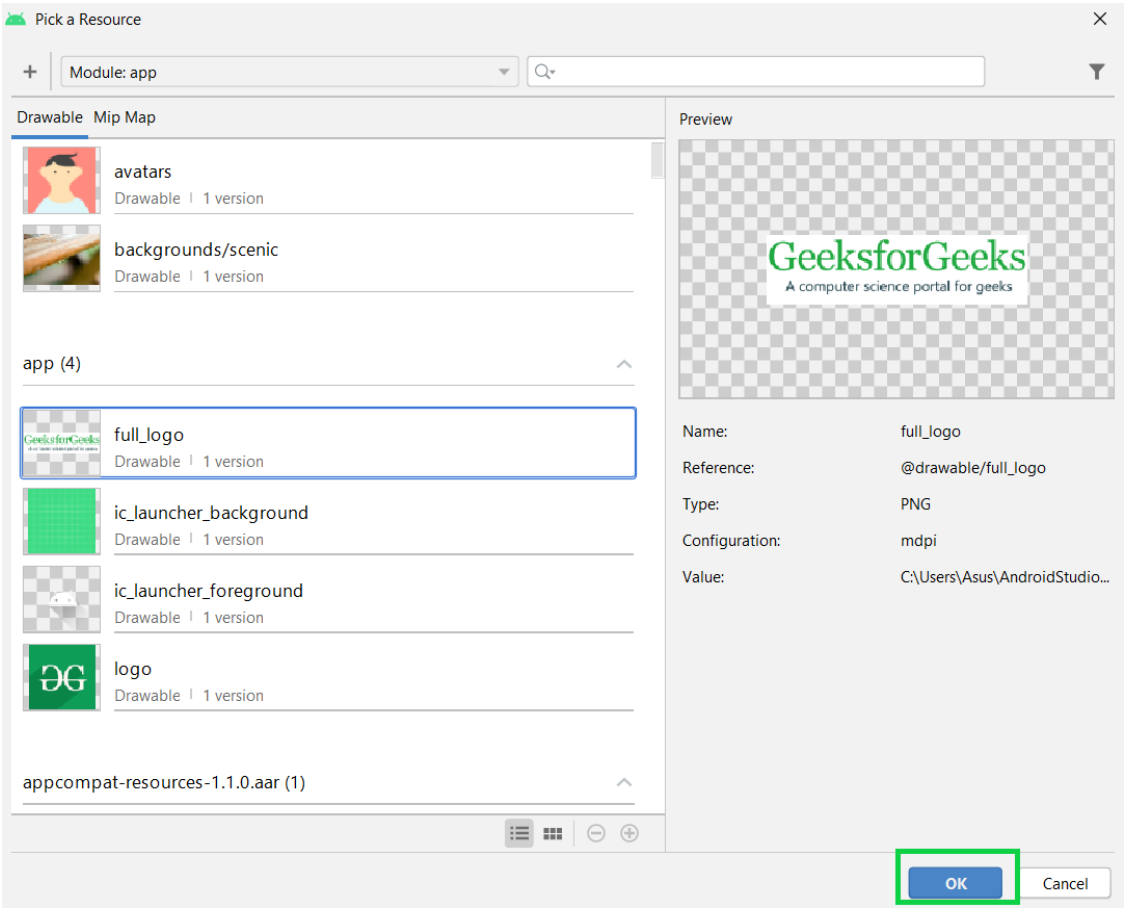
For Adding an Image File other than Android Studio Drawable Resources:
Click on the "**Resource Manager**" tab on the leftmost panel and select the "**Import Drawables**" option.



Select the path of the image file on your computer and click "**OK**". After that set, the "**Qualifier type**" and "**value**" of the image file according to your need and click "**Next**" then "**Import**".



Drag the ImageView class in the activity area, a pop-up dialogue box will appear which contains your imported image file. Choose your image file and click "**OK**", your image will be added to the activity.



Step 1: Create a New Project in Android Studio

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio.

Step 2: Working with the activity_main.xml file

Navigate to app > res > layout > activity_main.xml and add the code below. Comments are added in the code to get to know in detail.

- XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/idRLContainer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!--on below line we are creating a simple text view-->
    <TextView
        android:id="@+id/idTVHeading"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:gravity="center"
        android:padding="10dp"
        android:text="Video View in Android"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold" />

    <!-- adding VideoView to the layout -->
    <VideoView
        android:id="@+id/idVideoView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/idTVHeading"
        android:layout_centerInParent="true" />

</RelativeLayout>
```

Step 3: Working with the MainActivity file

Navigate to app > java > your app's package name > MainActivity file and add the below code to it. Comments are added in the code to get to know in detail.

- Kotlin
- Java

```
package com.gtappdevelopers.kotlingfgproject

import android.net.Uri
import android.os.Bundle
import android.widget.MediaController
import android.widget.VideoView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    // on below line we are creating a variable.
    lateinit var videoView: VideoView
    val videoUrl = "Paste Your Video URL Here"

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

```
        // on below line we are initializing our variables.
        videoView = findViewById(R.id.idVideoView)

        // Uri object to refer the
        // resource from the videoUrl
        val uri = Uri.parse(videoUrl)

        // sets the resource from the
        // videoUrl to the videoView
        videoView.setVideoURI(uri)

        // creating object of
        // media controller class
        val mediaController = MediaController(this)

        // sets the anchor view
        // anchor view for the videoView
        mediaController.setAnchorView(videoView)

        // sets the media player to the videoView
        mediaController.setMediaPlayer(videoView)

        // sets the media controller to the videoView
        videoView.setMediaController(mediaController);

        // starts the video
        videoView.start();

    }
}
```

Now run your application to see the output of it.

PRACTICAL 9

AIM: Android program based on RestAPI

Step 1: Create a New Project

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. Note that select Java as the programming language.

Step 2: Add the below dependency in your build.gradle file

Navigate to the Gradle Scripts > build.gradle(Module:app) and add the below dependency in the dependencies section.

```
// below dependency for using the retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
```

After adding this dependency sync your project and now move towards the AndroidManifest.xml part.

Step 3: Adding permissions to the internet in the AndroidManifest.xml file

Navigate to the app > AndroidManifest.xml and add the below code to it.

XML

```
1
<!--permissions for INTERNET-->
2
<uses-permission android:name="android.permission.INTERNET"/>
```

Step 4: Working with the activity_main.xml file

Navigate to the app > res > layout > activity_main.xml and add the below code to that file. Below is the code for the activity_main.xml file.

XML

```
1
<?xml version="1.0" encoding="utf-8"?>
2
<LinearLayout
3
   xmlns:android="http://schemas.android.com/apk/res/android"
4
   xmlns:tools="http://schemas.android.com/tools"
5
   android:layout_width="match_parent"
6
   android:layout_height="match_parent"
7
   android:orientation="vertical"
8
   tools:context=".MainActivity">
9

10
   <!--edit text field for adding name-->
11
   <EditText
```

```
12
    android:id="@+id/idEdtName"
13
    android:layout_width="match_parent"
14
    android:layout_height="wrap_content"
15
    android:layout_margin="10dp"
16
    android:layout_marginTop="40dp"
17
    android:hint="Enter your name" />
18

19
  <!--edit text for adding job-->
20
  <EditText
21
    android:id="@+id/idEdtJob"
22
    android:layout_width="match_parent"
23
    android:layout_height="wrap_content"
24
    android:layout_margin="10dp"
25
    android:hint="Enter your job" />
26

27
  <!--button for adding data-->
28
  <Button
29
    android:id="@+id/idBtnPost"
30
    android:layout_width="match_parent"
31
    android:layout_height="wrap_content"
32
    android:layout_margin="20dp"
33
    android:text="Send Data to API"
34
    android:textAllCaps="false" />
```

```
35

36
   <!--text view for displaying our API response-->
37
   <TextView
38
       android:id="@+id/idTVResponse"
39
       android:layout_width="match_parent"
40
       android:layout_height="wrap_content"
41
       android:layout_margin="10dp"
42
       android:gravity="center_horizontal"
43
       android:text="Response"
44
       android:textAlignment="center"
45
       android:textSize="15sp" />
46

47
   <!--progress bar for loading -->
48
   <ProgressBar
49
       android:id="@+id/idLoadingPB"
50
       android:layout_width="wrap_content"
51
       android:layout_height="wrap_content"
52
       android:layout_gravity="center"
53
       android:visibility="gone" />
54

55
</LinearLayout>
```

Step 5: Creating a modal class for storing our data

Navigate to the app > java > your app's package name > Right-click on it > New > Java class and name it as DataModal and add the below code to it. Comments are added inside the code to understand the code in more detail.

```
Java
1
public class DataModal {
2

3
    // string variables for our name and job
4
    private String name;
5
    private String job;
6

7
    public DataModal(String name, String job) {
8
        this.name = name;
9
        this.job = job;
10
    }
11

12
    public String getName() {
13
        return name;
14
    }
15

16
    public void setName(String name) {
17
        this.name = name;
18
    }
19

20
    public String getJob() {
21
        return job;
22
    }
23
```

```
24
   public void setJob(String job) {
25
       this.job = job;
26
   }
27

28
}
```

Step 6: Creating an Interface class for our API Call

Navigate to the app > java > your app's package name > Right-click on it > New > Java class select it as Interface and name the file as RetrofitAPI and add below code to it. Comments are added inside the code to understand the code in more detail.

Java

```
1
import retrofit2.Call;
2
import retrofit2.http.Body;
3
import retrofit2.http.POST;


public interface RetrofitAPI {

   // as we are making a post request to post a data
8
   // so we are annotating it with post
9
   // and along with that we are passing a parameter as users
10
   @POST("users")
11

12
   //on below line we are creating a method to post our data.
13
   Call<DataModal> createPost(@Body DataModal dataModal);
14
}
```

Step 7: Working with the MainActivity.java file

Go to the MainActivity.java file and refer to the following code. Below is the code for the MainActivity.java file. Comments are added inside the code to understand the code in more detail.

Java

```
1
import android.os.Bundle;
2
import android.view.View;
3
import android.widget.Button;
4
import android.widget.EditText;
5
import android.widget.ProgressBar;
6
import android.widget.TextView;
7
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
10

11
import retrofit2.Call;
12
import retrofit2.Callback;
13
import retrofit2.Response;
14
import retrofit2.Retrofit;
15
import retrofit2.converter.gson.GsonConverterFactory;
16

17
public class MainActivity extends AppCompatActivity {
18

19
    // creating variables for our edittext,
20
    // button, textview and progressbar.
21
    private EditText nameEdt, jobEdt;
22
    private Button postDataBtn;
23
    private TextView responseTV;
24
    private ProgressBar loadingPB;
25
```

```
26
   @Override
27
   protected void onCreate(Bundle savedInstanceState) {
28
      super.onCreate(savedInstanceState);
29
      setContentView(R.layout.activity_main);
      // initializing our views
32
      nameEdt = findViewById(R.id.idEdtName);
33
      jobEdt = findViewById(R.id.idEdtJob);
34
      postDataBtn = findViewById(R.id.idBtnPost);
35
      responseTV = findViewById(R.id.idTVResponse);
36
      loadingPB = findViewById(R.id.idLoadingPB);
37


38
      // adding on click listener to our button.
39
      postDataBtn.setOnClickListener(new View.OnClickListener() {
40
         @Override
41
         public void onClick(View v) {
42
            // validating if the text field is empty or not.
43
            if (nameEdt.getText().toString().isEmpty() &&
jobEdt.getText().toString().isEmpty()) {
44
               Toast.makeText(MainActivity.this, "Please enter both the values",
Toast.LENGTH_SHORT).show();
45
               return;
46
            }
47
            // calling a method to post the data and passing our name and job.
48
            postData(nameEdt.getText().toString(), jobEdt.getText().toString());
```

```
49
        }
50
    });
51
  }
52

53
  private void postData(String name, String job) {
54

55
    // below line is for displaying our progress bar.
56
    loadingPB.setVisibility(View.VISIBLE);
57

58
    // on below line we are creating a retrofit
59
    // builder and passing our base url
60
    Retrofit retrofit = new Retrofit.Builder()
61
        .baseUrl("https://reqres.in/api/")
62
        // as we are sending data in json format so
63
        // we have to add Gson converter factory
64
        .addConverterFactory(GsonConverterFactory.create())
65
        // at last we are building our retrofit builder.
66
        .build();
67
    // below line is to create an instance for our retrofit api class.
68
    RetrofitAPI retrofitAPI = retrofit.create(RetrofitAPI.class);
69

70
    // passing data from our text fields to our modal class.
71
    DataModal modal = new DataModal(name, job);
```

```
    // calling a method to create a post and passing our modal class.
74
    Call<DataModal> call = retrofitAPI.createPost(modal);
75

76
    // on below line we are executing our method.
77
    call.enqueue(new Callback<DataModal>() {
78
        @Override
79
        public void onResponse(Call<DataModal> call, Response<DataModal>
response) {
80
            // this method is called when we get response from our api.
81
            Toast.makeText(MainActivity.this, "Data added to API",
Toast.LENGTH_SHORT).show();
82

83
            // below line is for hiding our progress bar.
84
            loadingPB.setVisibility(View.GONE);
85

86
            // on below line we are setting empty text
87
            // to our both edit text.
88
            jobEdt.setText("");
89
            nameEdt.setText("");
90

91
            // we are getting response from our body
92
            // and passing it to our modal class.
93
            DataModal responseFromAPI = response.body();
94

95
```

```
            // on below line we are getting our data from modal class and adding it to
our string.
96
            String responseString = "Response Code : " + response.code() + "\nName :
" + responseFromAPI.getName() + "\n" + "Job : " + responseFromAPI.getJob();
97

98
            // below line we are setting our
99
            // string to our text view.
100
            responseTV.setText(responseString);
101
        }
102

103
        @Override
104
        public void onFailure(Call<DataModal> call, Throwable t) {
105
            // setting text to our text view when
106
            // we get error response from API.
107
            responseTV.setText("Error found is : " + t.getMessage());
108
        }
109
    });
110
  }
111
}
```

Now run your app and see the output of the app.

PRACTICAL 10

AIM: Flutter program to work with SQLite Database

Step 1: Set Up the Flutter Project

Create a new Flutter project named 'sql_example' using the following command:

flutter create sql_example

cd sql_example

Example :

```
C:\Users\Asus>flutter create sql_example
Creating project sql_example...
Resolving dependencies in `sql_example`... (2.5s)
Downloading packages...
Got dependencies in `sql_example`.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

  $ cd sql_example
  $ flutter run

Your application code is in sql_example\lib\main.dart.


C:\Users\Asus>cd sql_example

C:\Users\Asus\sql_example>
```

Step 2 : Adding Dependencies

To get started, open up your **pubsec.yaml** file in the project structure. Now, you'll want to add the following dependencies:

```
dependencies:
  flutter:
    sdk: flutter
  sqflite: ^2.2.6
  path: ^1.8.3
  path_provider: ^2.0.14
```

- **sqflite** is the package that provides SQLite integration for Flutter.
- **path** is a package that helps in locating the database file path.

Run **flutter pub get** to install the dependencies.

Step 3 : Defining the User Model

Create a file in the '**lib/user.dart**' to define a model class to represent user data. Here's an example of a simple gfg user model class, that has user id , username and an email, along with a constructor that initialize the data members :

user.dart

```
class User {
  final int? id;
  final String username;
```

```
  final String email;

  User({this.id, required this.username, required this.email});

  Map<String, dynamic> toMap() {
    return {'id': id, 'username': username, 'email': email};
  }

  factory User.fromMap(Map<String, dynamic> map) {
    return User(
      id: map['id'],
      username: map['username'],
      email: map['email'],
    );
  }
}
```

Step 4 : Defining the Database Class

In Database helper class we have all the functions are implemented here, Database class has the following methods

1. **initDb()**: function is used to initialize the database. It checks if the 'gfg_users' table exists, and if it doesn't, it creates it.
2. **_onCreate():** It is a callback function that gets executed when the database is created. Its purpose is to define the schema of the 'users' table.
3. **insertUser()**: To insert a new user into the 'gfg_users' table .
4. **queryAllUsers()**: Retrieves all users from the 'gfg_users' table.
5. **updateUser()**: To Update an existing user in the 'gfg_users' table.
6. **deleteUser()**: Deletes a user from the 'gfg_users' table by their ID.

database_helper.dart

```
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'user.dart';

class DatabaseHelper {
  static final DatabaseHelper instance = DatabaseHelper._instance();
  static Database? _database;

  DatabaseHelper._instance();

  Future<Database> get db async {
    _database ??= await initDb();
    return _database!;
  }

  Future<Database> initDb() async {
    String databasesPath = await getDatabasesPath();
```

```
    String path = join(databasesPath, 'geeksforgeeks.db');

    return await openDatabase(path, version: 1, onCreate: _onCreate);
  }

  Future _onCreate(Database db, int version) async {
    await db.execute('''
      CREATE TABLE gfg_users (
        id INTEGER PRIMARY KEY,
        username TEXT,
        email TEXT
      )
    ''');
  }

  Future<int> insertUser(User user) async {
    Database db = await instance.db;
    return await db.insert('gfg_users', user.toMap());
  }

  Future<List<Map<String, dynamic>>> queryAllUsers() async {
    Database db = await instance.db;
    return await db.query('gfg_users');
  }

  Future<int> updateUser(User user) async {
    Database db = await instance.db;
    return await db.update('gfg_users', user.toMap(), where: 'id = ?', whereArgs:
[user.id]);
  }

  Future<int> deleteUser(int id) async {
    Database db = await instance.db;
    return await db.delete('gfg_users', where: 'id = ?', whereArgs: [id]);
  }

  Future<void> initializeUsers() async {
    List<User> usersToAdd = [
      User(username: 'John', email: 'john@example.com'),
      User(username: 'Jane', email: 'jane@example.com'),
      User(username: 'Alice', email: 'alice@example.com'),
      User(username: 'Bob', email: 'bob@example.com'),
    ];

    for (User user in usersToAdd) {
      await insertUser(user);
```

```
    }
  }
}
```

Step 5 : Defining the main file

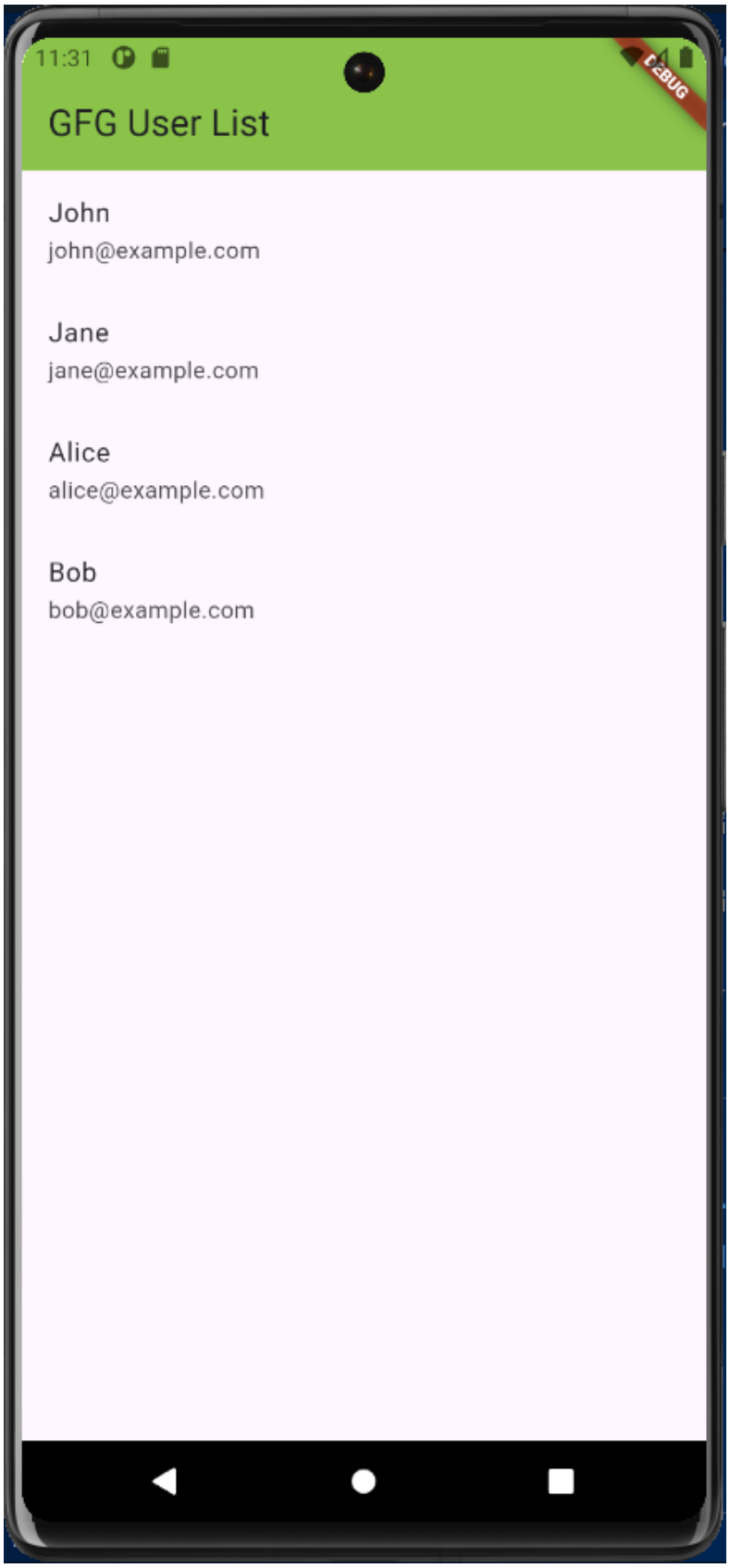
- **Initialization**: We make sure to initialize the database and insert the users ,before running the **MyApp** widget.
- **User List Display**: The UserList widget is responsible for displaying the list of users that we fetch from the database using the DatabaseHelper class.
- **State Management:** To keep things organized, we use the initState method to fetch the users when the widget is first initialized and update the user interface accordingly.
- **User Display**: We show each user's username and email using a **ListTile** widget within a **ListView.builder.**

main.dart

```
import 'package:flutter/material.dart';
import 'database_helper.dart'; // Import the DatabaseHelper class
import 'user.dart'; // Import the User class

void main() async {
  // Initialize the database and insert users
  WidgetsFlutterBinding.ensureInitialized();
  await DatabaseHelper.instance.initDb();
  await DatabaseHelper.instance.initializeUsers();

  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'User Management',
      home: UserList(),
    );
  }
}

class UserList extends StatefulWidget {
  @override
  _UserListState createState() => _UserListState();
}

class _UserListState extends State<UserList> {
  List<User> _users = [];
```

```
  @override
  void initState() {
    super.initState();
    _fetchUsers();
  }

  Future<void> _fetchUsers() async {
    final userMaps = await DatabaseHelper.instance.queryAllUsers();
    setState(() {
      _users = userMaps.map((userMap) => User.fromMap(userMap)).toList();
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('GFG User List'),
        backgroundColor: Colors.lightGreen,
      ),
      body: ListView.builder(
        itemCount: _users.length,
        itemBuilder: (context, index) {
          return ListTile(
            title: Text(_users[index].username),
            subtitle: Text(_users[index].email),
          );
        },
      ),
    );
  }
}
```

Output:


11:31
DEBUG
GFG User List
John
john@example.com
Jane
jane@example.com
Alice
alice@example.com
Bob
bob@example.com