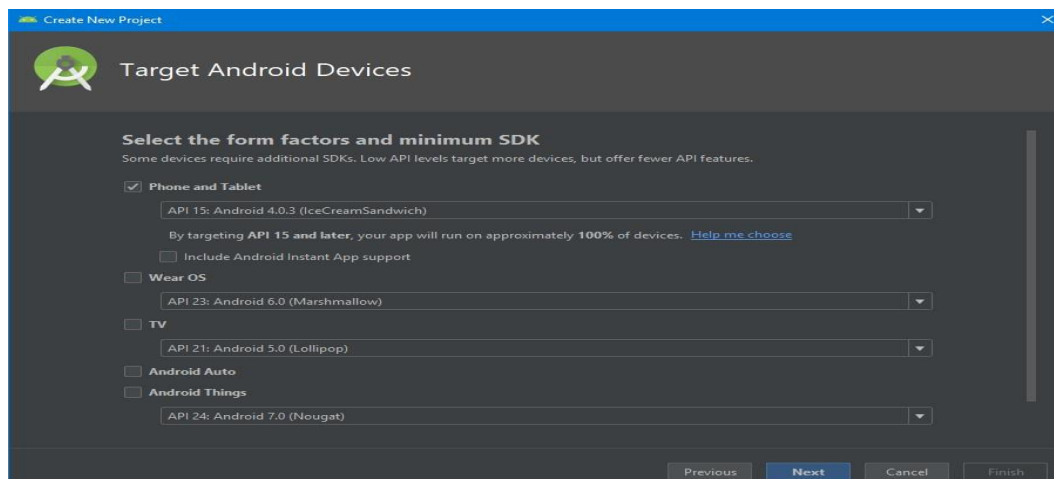
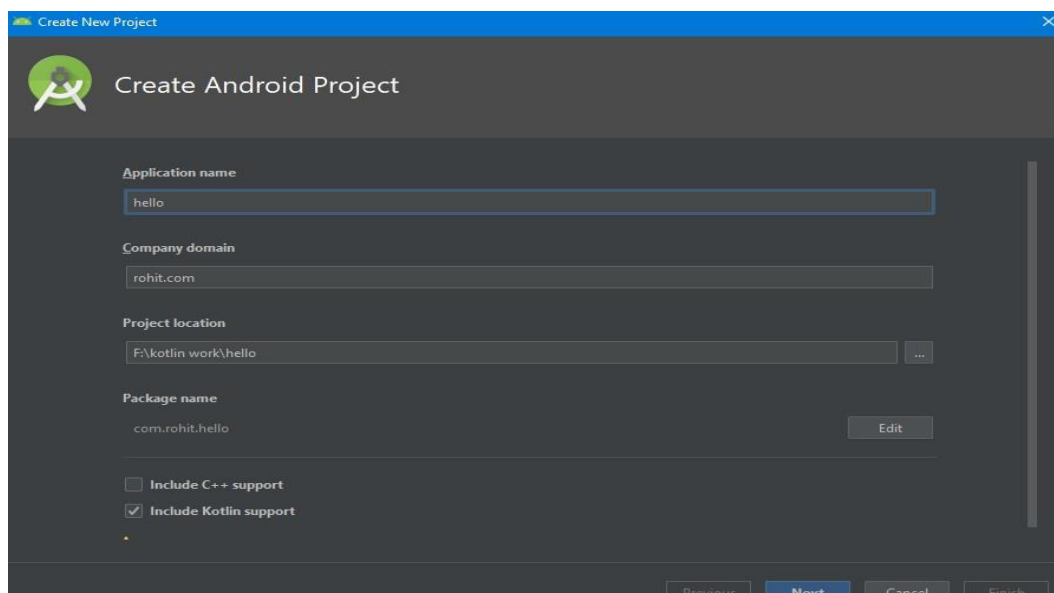


## Practical 1

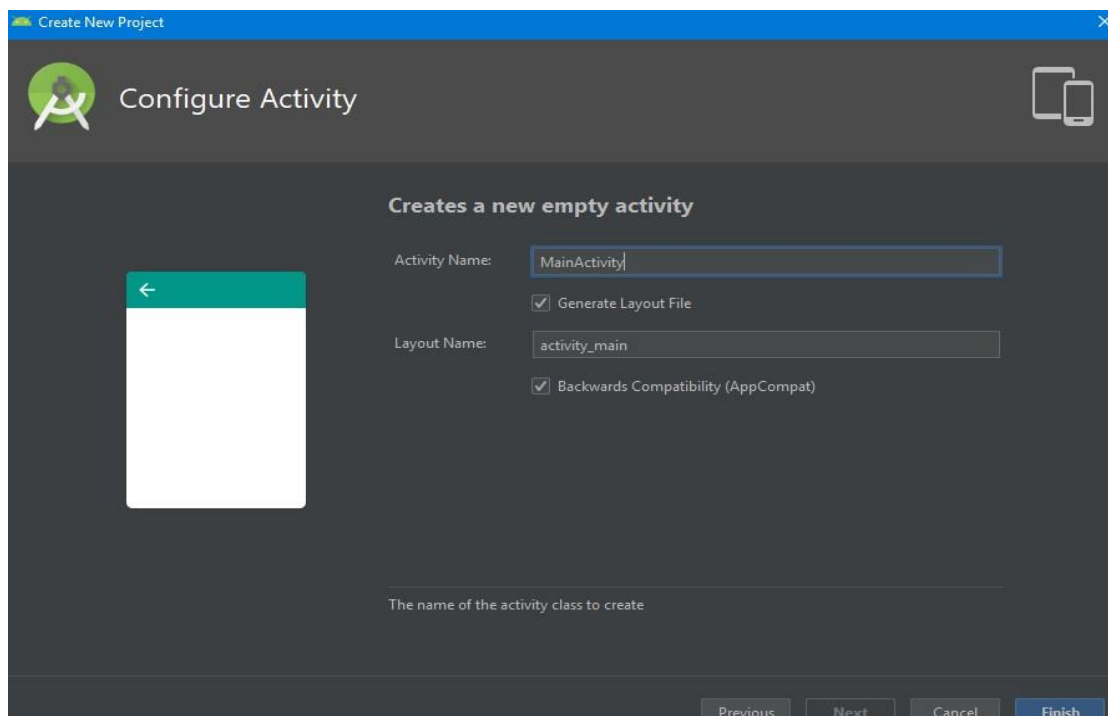
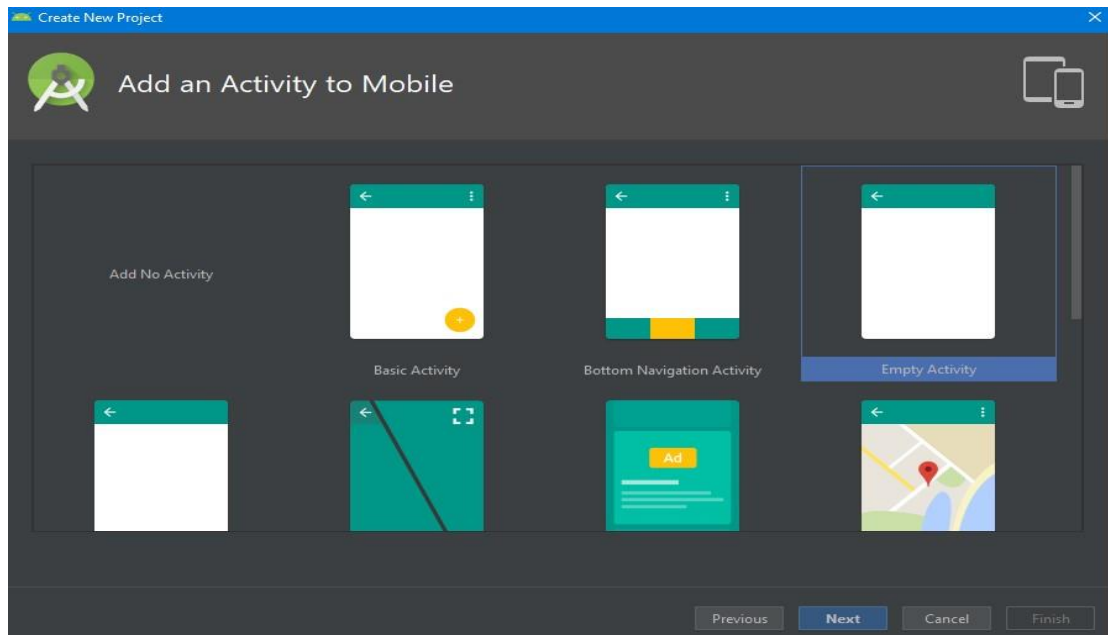
### **Aim: Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals:**

Creating a Project, Android Components, Activities, Services, Content Providers, Broadcast Receivers, Interface overview, Creating Android Virtual device, USB debugging mode, Android Application Overview. Simple “Hello World” program.

Creating a project:



## Android Programming Practical



### 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:id="@+id/main"
```

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

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

### MainActivity.java

```
package com.example.pract0;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

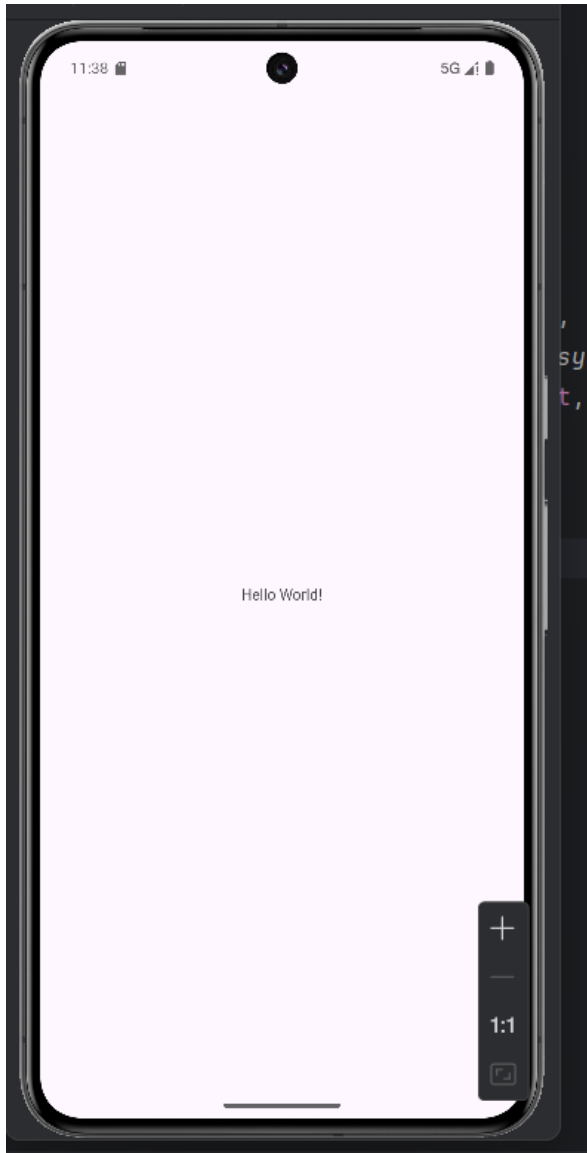
public class MainActivity extends AppCompatActivity {

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars =
            insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
            systemBars.bottom);
        });
    }
}
```

```
        return insets;  
    });  
}  
}
```

**Output:**

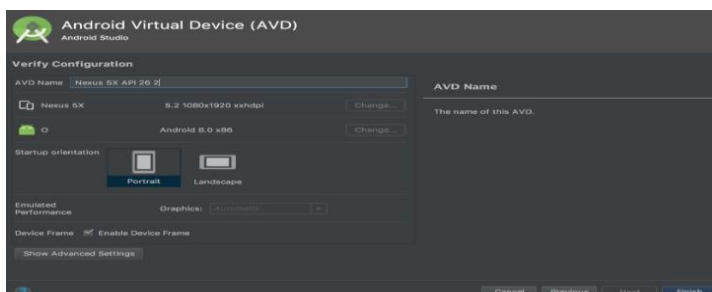
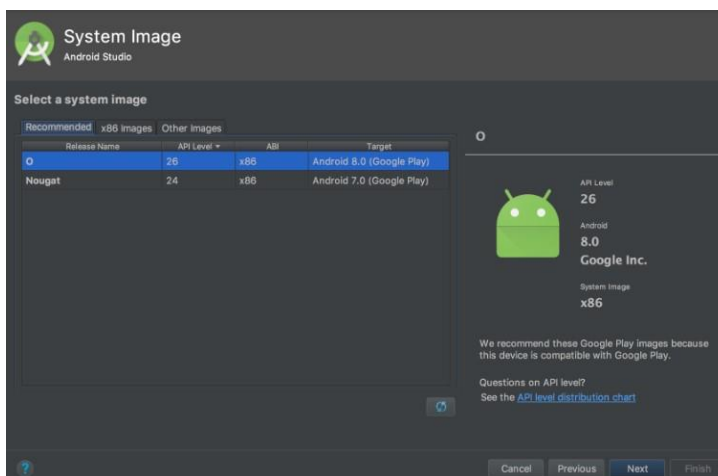
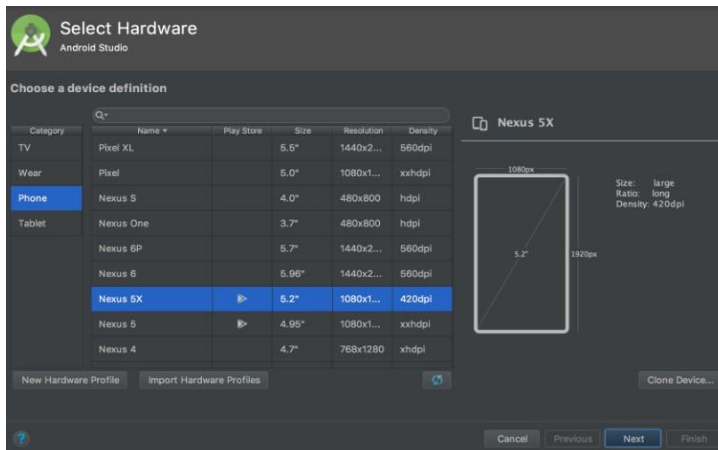
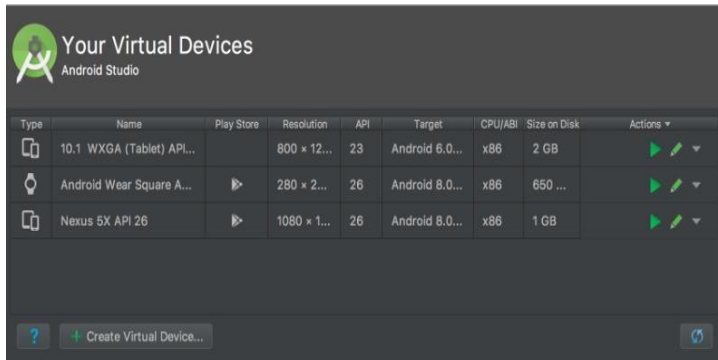


Create and manage virtual devices:

To open the AVD Manager, do one of the following:

- **Select Tools → AVD Manager.**
- **Click AVD Manager icon in the toolbar.**

## Android Programming Practical



Click **Finish** and **OK**

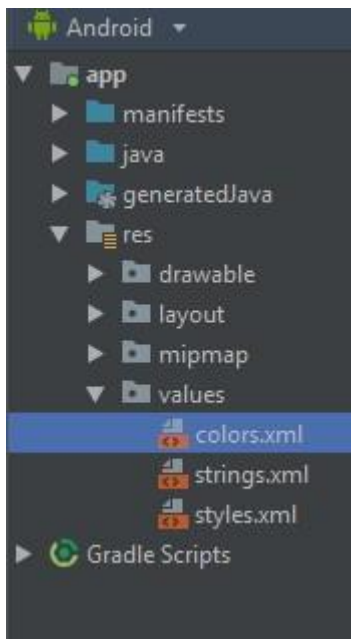
## Practical 2

### Aim : Programming Resources

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

- ✓ *Color: Defines app colors using XML or resource files for consistency across UI elements.*

### Color:

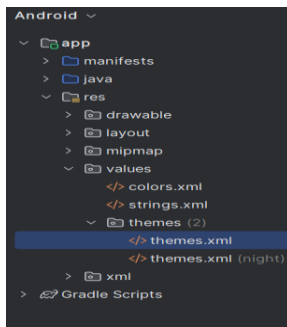


### **colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFF</color>
    <color name="red">#E42416</color>
    <color name="yellow">#FFEB3B</color>
</resources>
```

### Theme:

- ✓ *Theme: A collection of attributes that define the app's overall look, including colors, styles, and fonts.*

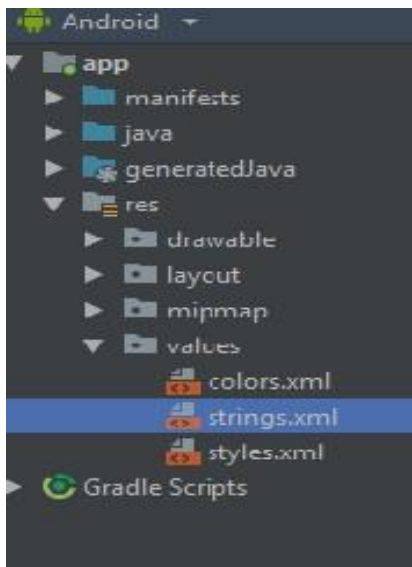


### themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <style name="Base.Theme.Pract1"
parent="Theme.AppCompat.Light.DarkActionBar">
  </style>
  <style name="Theme.Pract1" parent="Base.Theme.Pract1" />
</resources>
```

### String:

- ✓ String: Stores text values in strings.xml for localization and reusability.

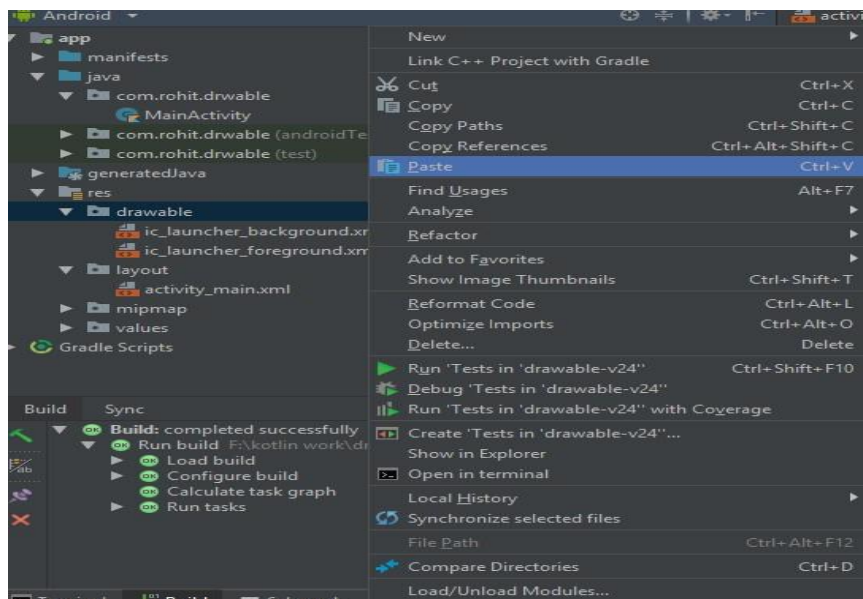


### String.xml

```
<resources>
  <string name="app_name">Pract1</string>
  <string name="welcome_msg">Hey Welcome to my App!</string>
</resources>
```

### Drawable:

- ✓ Drawable: XML or image-based graphics used for UI elements like icons and backgrounds.
1. Right click on drawable folder -> New -> Drawable Resource File if you want to create border
  2. Give File Name->border->OK
  3. Copy the image if you want to create image drawable
  4. Paste that image file inside the drawable folder



### Dimension, Image:

- ✓ Dimension: Defines size values (e.g., padding, margins, text size) for scalable UI design.
- ✓ Image: Bitmap or vector assets used for UI elements and animations.

### MainActivity.java

```
package com.example.pract1;  
import android.os.Bundle;  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;
```



```
public class MainActivity extends AppCompatActivity {

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
    }
}
```

### **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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/welcome_msg"
        android:textColor="@color/red"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

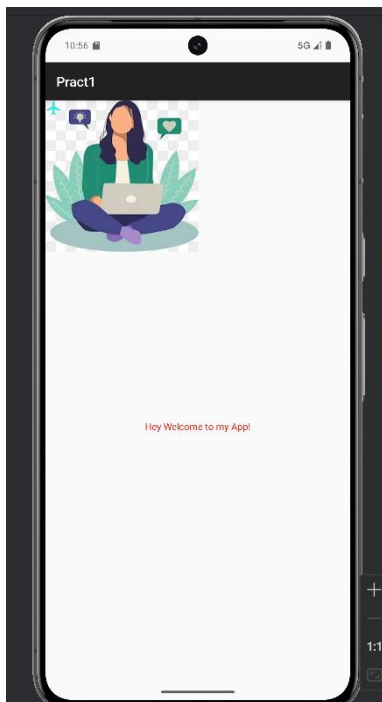
```
app:layout_constraintVertical_bias="0.56" />
```

```
<ImageView  
    android:id="@+id/imageView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:scaleType="matrix"  
    android:scrollbarAlwaysDrawHorizontalTrack="false"  
    android:scrollbarAlwaysDrawVerticalTrack="false"  
    app:srcCompat="@drawable/girl"  
    tools:layout_editor_absoluteX="93dp"  
    tools:layout_editor_absoluteY="116dp" />
```

```
<ImageView  
    android:id="@+id/imageView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:srcCompat="@drawable/airplanemode"  
    tools:layout_editor_absoluteX="371dp"  
    tools:layout_editor_absoluteY="16dp" />
```

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

### Output:

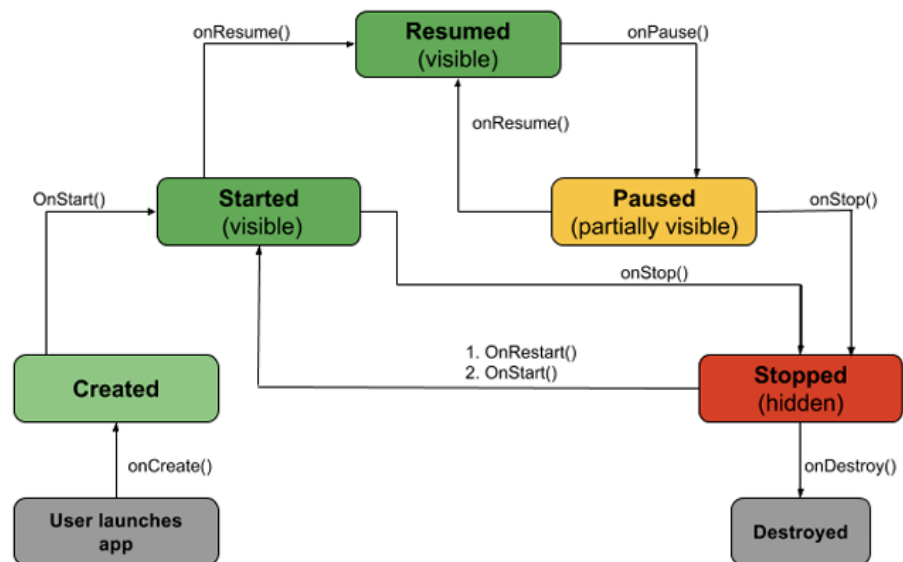


### Practical 3

**Aim : Programming Activities and fragments**

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

**Activity Lifecycle:**



- ***onCreate():*** Called by the OS when the activity is first created. This is where you initialize any UI elements or data objects. You also have the *savedInstanceState* of the activity that contains its previously saved state, and you can use it to recreate that state.
- ***onStart():*** Just before presenting the user with an activity, this method is called. It's always followed by *onResume()*. In here, you generally should start UI animations, audio based content or anything else that requires the activity's contents to be on screen.
- ***onResume():*** As an activity enters the foreground, this method is called. Here you have a good place to restart animations, update UI elements, restart camera previews, resume audio/video playback or initialize any components that you release during *onPause()*.
- ***onPause():*** This method is called before sliding into the background. Here you should stop any visuals or audio associated with the activity such as UI animations, music playback or the camera. This method is followed by *onResume()* if the activity returns to the foreground or by *onStop()* if it becomes hidden.

- ***onStop():*** This method is called right after *onPause()*, when the activity is no longer visible to the user, and it's a good place to save data that you want to commit to the disk. It's followed by either *onRestart()*, if this activity is coming back to the foreground, or *onDestroy()* if it's being released from memory.
- ***onRestart():*** Called after stopping an activity, but just before starting it again. It's always followed by *onStart()*.
- ***onDestroy():*** This is the final callback you'll receive from the OS before the activity is destroyed. You can trigger an activity's destruction by calling *finish()*, or it can be triggered by the system when the system needs to recoup memory. If your activity includes any background threads or other long-running resources, destruction could lead to a memory leak if they're not released, so you need to remember to stop these processes here as well.

### Themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Pract2B"
parent="Theme.AppCompat.DayNight.DarkActionBar">
    </style>
    <style name="Theme.Pract2B" parent="Base.Theme.Pract2B" />
</resources>
```

### MainActivity.java

```
package com.example.pract2b;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
}
```

### **Fragment\_Life.java**

```
package com.example.pract2b;
import android.content.Context;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment_Life extends Fragment {

    String tag="LifeCycle";

    public Fragment_Life() {
        // Required empty public constructor
    }

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

    @Override
    public void onAttach(@NonNull Context context) {
        super.onAttach(context);
        Log.d(tag,"in onAttach event");
    }
}
```

```
@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Log.d(tag,"in onCreate event");
}

@Override
public void onActivityCreated(@Nullable Bundle
savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    Log.d(tag,"in onActivityCreated event");
}

@Override
public void onViewCreated(@NonNull View view, @Nullable
Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    Log.d(tag,"in onViewCreated event");
}

@Override
public void onViewStateRestored(@Nullable Bundle
savedInstanceState) {
    super.onViewStateRestored(savedInstanceState);
    Log.d(tag,"in onViewStateRestored event");
}

@Override
public void onStart() {
    super.onStart();
    Log.d(tag,"onStart event");
}

@Override
public void onResume() {
    super.onResume();
    Log.d(tag,"onResume event");
}

@Override
```

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

@Override
public void onStop() {
    super.onStop();
    Log.d(tag,"onStop event");
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    Log.d(tag,"onDestroyView event");
}

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

### **Fragment\_life.xml**

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

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="Hello, welcome to fragment" />

</FrameLayout>
```

**activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

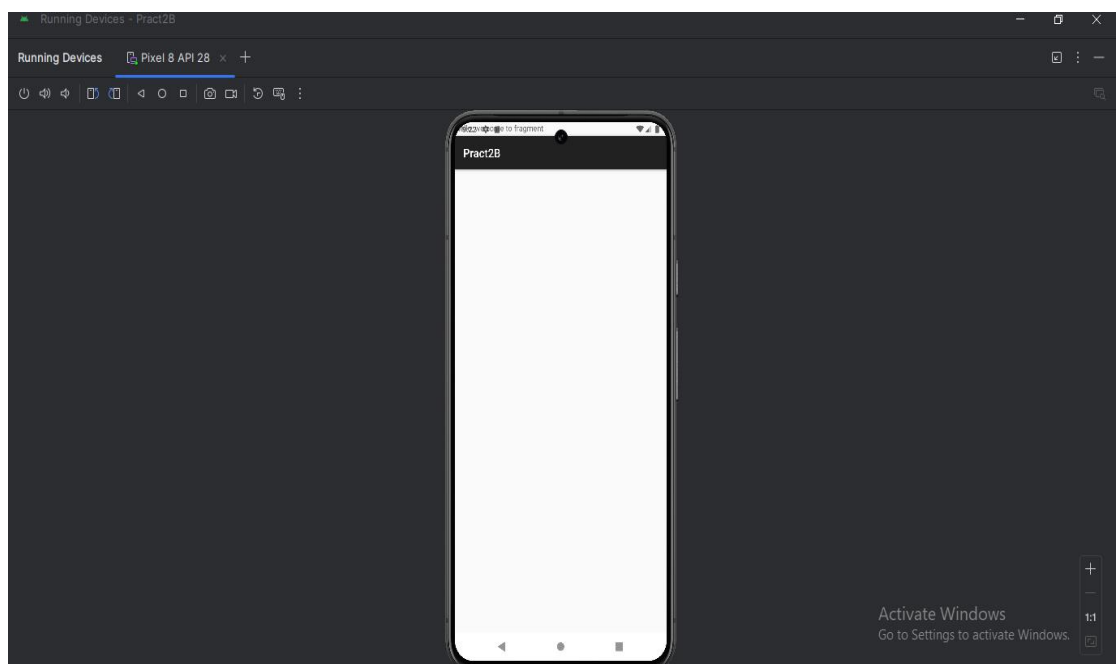
    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout="@layout/fragment__life"
        android:id="@+id/frg1"
        class="com.example.pract2b.Fragment_Life"/>
</RelativeLayout>
```



### Output:

The screenshot shows the Logcat window in Android Studio. The filter is set to 'package:mime lifecycle'. The log entries show the lifecycle events for the package 'com.example.pract2b' on a Pixel 8 API 28 emulator. The events are as follows:

Time	Process	Thread	Message
2025-03-11 21:16:44.940	6130-6130	LifeCycle	com.example.pract2b D in onAttach event
2025-03-11 21:16:44.940	6130-6130	LifeCycle	com.example.pract2b D in onCreate event
2025-03-11 21:16:45.000	6130-6130	LifeCycle	com.example.pract2b D in onViewCreated event
2025-03-11 21:16:45.009	6130-6130	LifeCycle	com.example.pract2b D in onActivityCreated event
2025-03-11 21:16:45.010	6130-6130	LifeCycle	com.example.pract2b D in onViewStateRestored event
2025-03-11 21:16:45.038	6130-6130	LifeCycle	com.example.pract2b D onStart event
2025-03-11 21:16:45.044	6130-6130	LifeCycle	com.example.pract2b D onResume event
2025-03-11 21:16:59.545	6130-6130	LifeCycle	com.example.pract2b D onPause event
2025-03-11 21:16:59.614	6130-6130	LifeCycle	com.example.pract2b D onStop event
2025-03-11 21:17:38.349	6130-6130	LifeCycle	com.example.pract2b D onStart event
2025-03-11 21:17:38.357	6130-6130	LifeCycle	com.example.pract2b D onResume event
2025-03-11 21:17:41.346	6130-6130	LifeCycle	com.example.pract2b D onPause event
2025-03-11 21:17:42.003	6130-6130	LifeCycle	com.example.pract2b D onStop event
2025-03-11 21:17:42.007	6130-6130	LifeCycle	com.example.pract2b D onDestroyView event
2025-03-11 21:17:42.015	6130-6130	LifeCycle	com.example.pract2b D onDestroy event
2025-03-11 21:17:55.010	6130-6130	LifeCycle	com.example.pract2b D in onAttach event
2025-03-11 21:17:55.011	6130-6130	LifeCycle	com.example.pract2b D in onCreate event
2025-03-11 21:17:55.018	6130-6130	LifeCycle	com.example.pract2b D in onViewCreated event
2025-03-11 21:17:55.026	6130-6130	LifeCycle	com.example.pract2b D in onActivityCreated event



## Practical 4

**Aim : Programs related to different Layouts**

**Coordinate, Linear, Relative, Table, Absolute, Frame, List View.**

### 1. Linear Layout

*LinearLayout is one of the most basic layouts in android studio, that arranges multiple sub-views (UI elements) sequentially in a single direction i.e. horizontal or vertical manner by specifying the android:orientation attribute.*

*If one applies android:orientation="vertical" then elements will be arranged one after another in a vertical manner (i.e. top to bottom) and If you apply android:orientation="horizontal" then elements will be arranged one after another in a horizontal manner (i.e. left to right).*

### **Themes.xml**

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Pract3A"
parent="Theme.AppCompat.DayNight.DarkActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item
name="colorPrimary">@color/my_light_primary</item> -->
    </style>

    <style name="Theme.Pract3A" parent="Base.Theme.Pract3A" />
</resources>
```

### **activity\_main.xml**

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >

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

        <EditText
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:textSize="30dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:textSize="30dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:textSize="30dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Reset"
        android:textSize="30dp"/>

</LinearLayout>
</LinearLayout>
```

## 2. Relative Layout

*Relative Layout in Android is a layout that arranges its child views in relation to each other. Unlike Linear Layout, which can make element arrangement complex, RelativeLayout simplifies the process by allowing flexible and dynamic positioning. It enables you to align views relative to each other or to the edges of the parent container, making it ideal for creating more complex and responsive UI designs.*

### **activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >

    <EditText
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:hint="Username"
        android:textSize="30dp"
        android:id="@+id/txtusername"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:textSize="30dp"
        android:layout_below="@id/txtusername"
        android:id="@+id/txtpassword"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:textSize="30dp"
        android:id="@+id/sub"
        android:layout_below="@id/txtpassword"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="Reset"
        android:layout_below="@id/txtpassword"
        android:layout_toRightOf="@id/sub"/>

</RelativeLayout>
```

### 3. Table Layout

*TableLayout is a ViewGroup that displays child View elements in rows and columns.*

#### **activity\_main.xml**

```
<TableLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TableRow>
        <EditText android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:hint="Username"
            android:textSize="30dp"/>
```

```

</TableRow>
<TableRow>
    <EditText android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:hint="Password"
        android:textSize="30dp"/>

</TableRow>
<TableRow>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="Submit"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="Reset"/>
</TableRow>

</TableLayout>

```

#### 4. Absolute Layout

*An Absolute Layout allows you to specify the exact location i.e. X and Y coordinates of its children with respect to the origin at the top left corner of the layout. The absolute layout is less flexible and harder to maintain for varying sizes of screens that's why it is not recommended. Although Absolute Layout is deprecated now.*

##### **activity\_main.xml**

```

<AbsoluteLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:textSize="30dp"/>

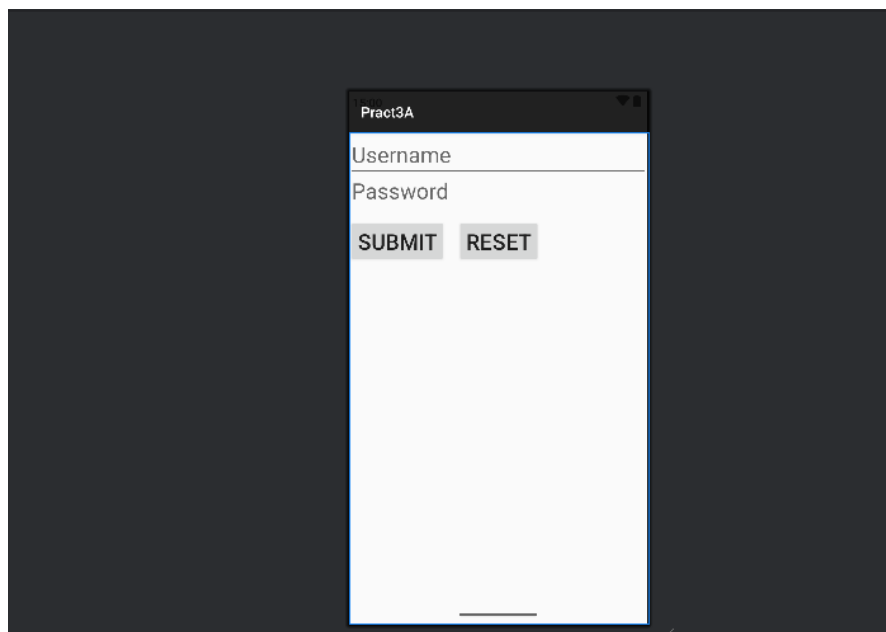
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:layout_y="50dp"
    android:textSize="30dp"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"
    android:textSize="30dp"
    android:layout_y="120dp"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reset"
    android:textSize="30dp"
    android:layout_x="150dp"
    android:layout_y="120dp"/>

</AbsoluteLayout>
```

**Output of 1,2,3,4:**



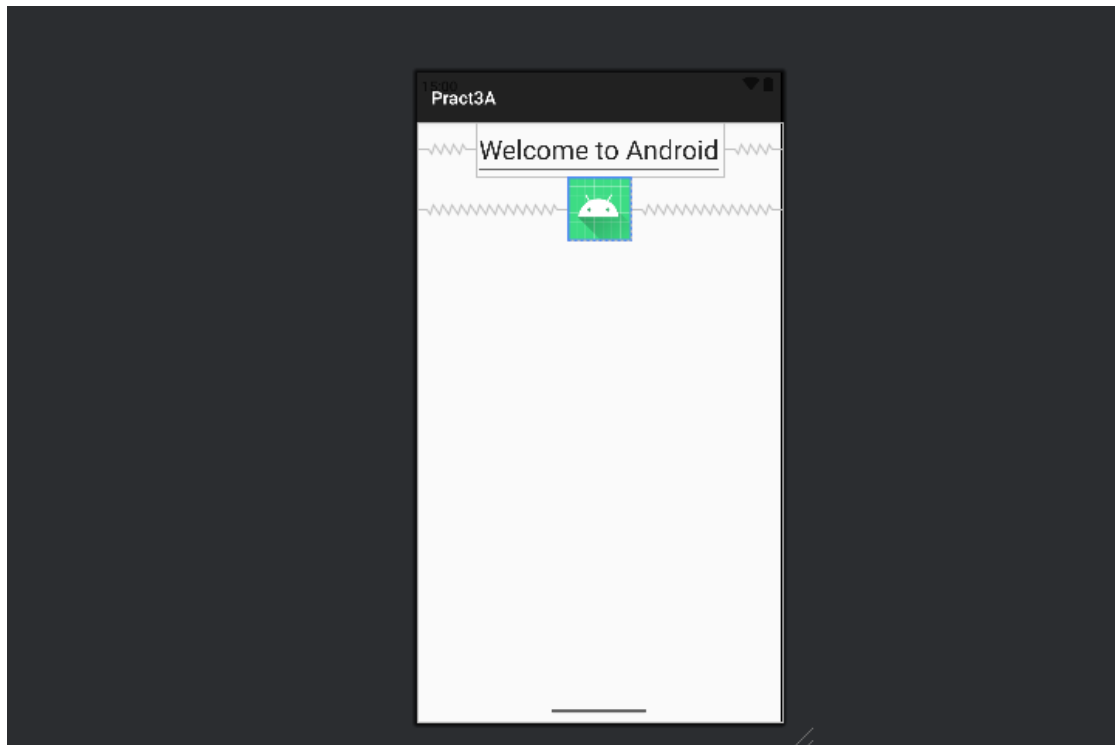
## 5. Frame Layout

*Android Framelayout is a ViewGroup subclass that is used to specify the position of multiple views placed on top of each other to represent a single view screen. Generally, we can say FrameLayout simply blocks a particular area on the screen to display a single view. Here, all the child views or elements are added in stack format means the most recently added child will be shown on the top of the screen. But, we can add multiple children's views and control their positions only by using gravity attributes in FrameLayout.*

### **activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Android"
        android:textSize="30dp"
        android:id="@+id/txt"
        android:layout_centerHorizontal="true"/>
    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/txt"
        android:layout_centerHorizontal="true">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@mipmap/ic_launcher_round"/>
    </FrameLayout>
</RelativeLayout>
```

**Output: -**



### 6. List View

*A ListView in Android is a type of AdapterView that displays a vertically scrollable list of items, with each item positioned one below the other. Using an adapter, items are inserted into the list from an array or database efficiently. For displaying the items in the list method setAdaptor() is used. The setAdaptor() method binds the adapter with the ListView.*

#### **activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listv"/>

</RelativeLayout>
```



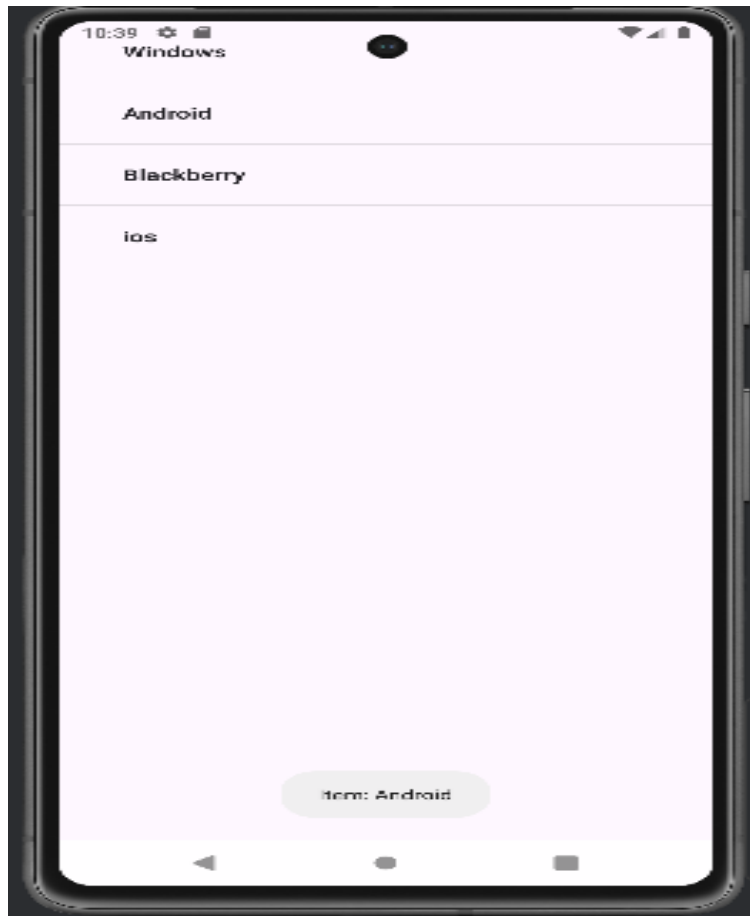
## MainActivity.java

```
package com.example.pract3new;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    ListView l;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        final String[] s={"Windows","Android","Blackberry","ios"};
        l=(ListView)findViewById(R.id.listv);
        ArrayAdapter ada=new ArrayAdapter(MainActivity.this,
        android.R.layout.simple_expandable_list_item_1,s);
        l.setAdapter(ada);
        l.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {
                Toast.makeText( MainActivity.this, "Item:
                "+s[position],Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

**Output:**



## Practical 5

**Aim : Programming UI elements**

**AppBar, Fragments, UI Components**

5a ) AppBar

*The **app bar**, also known as the action bar, is one of the most important design elements in your app's activities, because it provides a visual structure and interactive elements that are familiar to users.*

**Step-1**

**Res → Layout → New → Layout Resource File → toolbar\_layout**

**Remove inbulid code in the toolbar\_layout**

**toolbar\_layout.xml**

```
<Toolbar android:layout_width="match_parent"
    android:id="@+id/toolbar"
    android:layout_height="100dp"
    android:background="#787"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar"
    android:popupTheme="@style/Theme.AppCompat.Light"
    xmlns:android="http://schemas.android.com/apk/res/android" />
```

**Step -2**

**activity\_main.xml**

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="100dp"/>

    <include layout="@layout/toolbar_layout"/>
</LinearLayout>
```

**Go to browser → search for “android icon download”**

## MainActivity.java

```

package com.example.pract4a;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Switch;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    Toolbar tb;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        tb=findViewById(R.id.toolbar);
        setSupportActionBar(tb);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater mi=getMenuInflater();
        mi.inflate(R.menu.app_bar_menu,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if(item.getItemId()==R.id.a_Search)

```

```

        {
            Toast.makeText(this,"Search is Clicked",
Toast.LENGTH_LONG).show();
            return true;
        } else if (item.getItemId()==R.id.a_chat) {
            Toast.makeText(this,"Chat is
Clicked",Toast.LENGTH_LONG).show();
            return true;

            }else if (item.getItemId()==R.id.a_above) {
                Toast.makeText(this,"Chat is
Clicked",Toast.LENGTH_LONG).show();
                return true;

            }
        } else {
            return super.onOptionsItemSelected(item);
        }
    }
}

```

### app\_bar\_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
<item android:id="@+id/a_Search"
    android:title="search"
    android:icon="@mipmap/search"
    app:showAsAction="ifRoom"/>

<item android:id="@+id/a_above"
    android:title="About"
    android:icon="@mipmap/about"
    app:showAsAction="ifRoom"/>

<item android:id="@+id/a_chat"
    android:title="Chat"
    android:icon="@mipmap/chat"

```

```
        app:showAsAction="ifRoom"/>
</menu>
```

### 5b ) Fragments

*A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments can't live on their own. They must be hosted by an activity or another fragment.*

#### **activity\_main.xml**

```
<FrameLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/frag1"
    xmlns:android="http://schemas.android.com/apk/res/android" >

</FrameLayout>
```

**Step1 : App→ New→ fragment→ Blank fragment→  
“FirstFragment” → Finish.**

**Step2 : App→ New→ fragment→ Blank fragment→  
“SecondFragment” → Finish.**

**Step3 : App→ New→ fragment→ Blank fragment→  
“ThirdFragment” → Finish.**

**Step 4: Go to layout→ Open fragment\_first.xml**

#### **fragment\_first.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Fragment"
        android:textSize="40dp"
```

```

        android:layout_above="@+id/btnsfrag"
        android:layout_centerHorizontal="true"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Second Fragment"
    android:layout_centerInParent="true"
    android:id="@+id/btnsfrag"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/btnsfrag"
    android:text="Third Fragment"
    android:layout_centerInParent="true"
    android:id="@+id/btntfrag"/>
</RelativeLayout>

```

### MainActivity.java

```

package com.example.pract4b;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

public class MainActivity extends AppCompatActivity {
    public static FragmentManager fragmentManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        fragmentManager=getSupportFragmentManager();
        if(findViewById(R.id.fragI)!=null)

```

```

        {
            if(savedInstanceState !=null)
            {
                return;
            }
            FragmentTransaction
fragmentTransaction=fragmentManager.beginTransaction();
            FirstFragment ff=new FirstFragment();
            fragmentTransaction.add(R.id.frag1,ff,null);
            fragmentTransaction.commit();
        }
    }
}

```

### **FirstFragment.java**

```

package com.example.pract4b;
import android.annotation.SuppressLint;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class FirstFragment extends Fragment {
    private Button sbtn,tbtn;
    public FirstFragment() {

    }

    @SuppressWarnings("MissingInflatedId")
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                            Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_first,container,false);
        sbtn=view.findViewById(R.id.btnsfrag);
        sbtn.setOnClickListener(new View.OnClickListener() {

```



```
        @Override
        public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,ne
w SecondFragment(),null).addToBackStack(null).commit();
        }
    });
    tbtn=view.findViewById(R.id.btntfrag);
    tbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,ne
w ThirdFragment(),null).addToBackStack(null).commit();
        }
    });
    return view;
}
}
```

### **SecondFragment.java**

```
package com.example.pract4b;
import android.annotation.SuppressLint;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class SecondFragment extends Fragment {

    private Button fbtn,tbtn;

    public SecondFragment() {

    }
}
```

```

@SuppressLint("MissingInflatedId")
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    View
view=inflater.inflate(R.layout.fragment_second,container,false);
    fbtn=view.findViewById(R.id.btnffrag);
    fbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,ne
w FirstFragment(),null).addToBackStack(null).commit();
        }
    });
    tbtn=view.findViewById(R.id.btntfrag);
    tbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,
new ThirdFragment(),null).addToBackStack(null).commit();
        }
    });
    return view;
}
}

```

### ThirdFragment.java

```

package com.example.pract4b;
import android.annotation.SuppressLint;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```
import android.widget.Button;

public class ThirdFragment extends Fragment {
    private Button fbtn,sbtn;
    public ThirdFragment() {

    }

    @SuppressWarnings("MissingInflatedId")
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.fragment_third,container,false);
        fbtn=view.findViewById(R.id.btnffrag);
        fbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,ne
w FirstFragment(),null).addToBackStack(null).commit();
            }
        });
        sbtn=view.findViewById(R.id.btnsfrag);
        sbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

MainActivity.fragmentManager.beginTransaction().replace(R.id.frag1,ne
w SecondFragment(),null).addToBackStack(null).commit();
            }
        });
        return view;
    }
}
```

### 5c) UI Components

There are number of UI controls provided by Android that allow you to build the graphical user interface for your app.

<b>Sr.No.</b>	<b>UI Control &amp; Description</b>
1	<b><i>TextView</i></b> <i>This control is used to display text to the user.</i>
2	<b><i>EditText</i></b> <i>EditText is a predefined subclass of TextView that includes rich editing capabilities.</i>
3	<b><i>AutoCompleteTextView</i></b> <i>The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing.</i>
4	<b><i>Button</i></b> <i>A push-button that can be pressed, or clicked, by the user to perform an action.</i>
5	<b><i>ImageButton</i></b> <i>An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user.</i>
6	<b><i>CheckBox</i></b> <i>An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive.</i>
7	<b><i>ToggleButton</i></b> <i>An on/off button with a light indicator.</i>
8	<b><i>RadioButton</i></b> <i>The RadioButton has two states: either checked or unchecked.</i>
9	<b><i>RadioGroup</i></b> <i>A RadioGroup is used to group together one or more RadioButtons.</i>

10	<b><i>ProgressBar</i></b> <i>The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background.</i>
11	<b><i>Spinner</i></b> <i>A drop-down list that allows users to select one value from a set.</i>
12	<b><i>TimePicker</i></b> <i>The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode.</i>
13	<b><i>DatePicker</i></b> <i>The DatePicker view enables users to select a date of the day.</i>

**Step 1: Open activity\_main.xml write the below code in it.**

**activity\_main.xml**

```

<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Save"
        android:id="@+id/btnsave"
        android:textSize="30dp"/>
    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher_round"
        android:id="@+id/imgbtn"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:hint="Enter Your Lovly Name"
        android:id="@+id/txtname1"/>

```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="IT"
    android:textSize="30dp"
    android:id="@+id/chkit"/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CS"
    android:textSize="30dp"
    android:id="@+id/chkcs"/>
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/rg">
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male"
        android:textSize="30dp"
        android:id="@+id/rbmale"/>
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female"
        android:textSize="30dp"
        android:id="@+id/rbfemale"/>
</RadioGroup>

<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tb"/>
</LinearLayout>
```

**Step2: Open MainActivity.java and write the code.**

### MainActivity.java

```
package com.example.pract4c;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import android.widget.ToggleButton;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    Button btnsave;
    ImageButton imgbtn;
    EditText txtname;
    CheckBox chkit,chkcs;
    RadioGroup rg;
    RadioButton rbmale,rbfemale;
    ToggleButton tb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        btnsave=(Button)findViewById(R.id.btnsave);
        imgbtn=(ImageButton) findViewById(R.id.imgbtn);
        txtname=(EditText)findViewById(R.id.txtname1);
        chkcs=(CheckBox)findViewById(R.id.chkcs);
        chkit=(CheckBox) findViewById(R.id.chkit);
        rg=(RadioGroup) findViewById(R.id.rg);
        rbfemale=(RadioButton) findViewById(R.id.rbfemale);
        rbmale=(RadioButton) findViewById(R.id.rbmale);
        tb=(ToggleButton) findViewById(R.id.tb);
        btnsave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        DisplayToast("Save Button CClick");
    }

});
imgbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DisplayToast("Image Button CClick");
    }
});
chkcs.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(chkcs.isChecked())
        {
            DisplayToast("Class cs is checked");
        }
        else {
            DisplayToast("Class CS is Unchecked");
        }
    }
});
chkkit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(chkit.isChecked())
        {
            DisplayToast("Class IT is checked");
        }
        else {
            DisplayToast("Class IT is Unchecked");
        }
    }
});
rbmale.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(rbmale.isChecked())
        {
            DisplayToast("Male is checked");
        }
    }
});

```

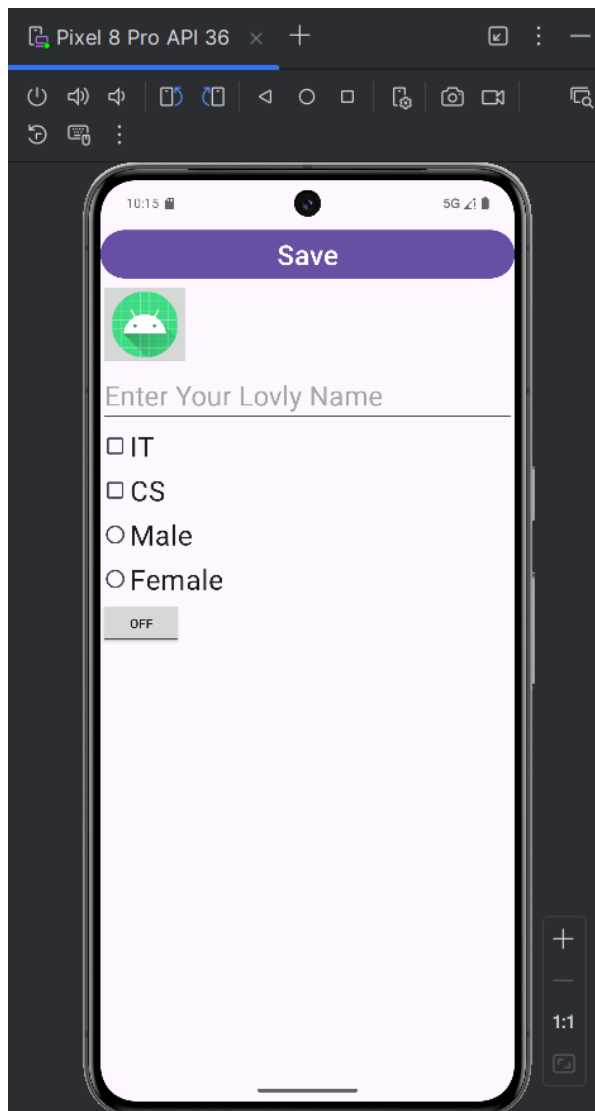


```
        else {
            DisplayToast("Male is Unchecked");
        }

    }
});
rbfemale.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(rbfemale.isChecked())
        {
            DisplayToast("Female is checked");
        }
        else {
            DisplayToast("Female is Unchecked");
        }
    }
});
tb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DisplayToast("Toggle Button CLicked");
    }
});

}
public void DisplayToast(String s)
{
    Toast.makeText(this,s,Toast.LENGTH_SHORT).show();
}
}
```

**Output:**



## Practical 6

### Aim : Programming menus, dialog, dialog fragments

#### 6a) Programming menus

*Menus in Android are a collection of options that provide users with different choices and functionalities within an application. They are typically used for actions that don't fit directly on the main user interface (UI).*

#### Step 1 : Change the Themes.xml

##### themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Pract5a"
parent="Theme.AppCompat.DayNight.DarkActionBar">
        </style>

    <style name="Theme.Pract5a" parent="Base.Theme.Pract5a" />
</resources>
```

Step2: res->Right click on it->new->Android Resource File

->my\_menu-> write the code.

##### my\_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/search"
        android:title="Search"/>
    <item android:id="@+id/about"
        android:title="About"/>
    <item android:id="@+id/exit"
        android:title="Exit"/>
</menu>
```

##### 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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

**Step3: Open MainActivity.java write code in it.**

### MainActivity.java

```
package com.example.pract5a;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

```

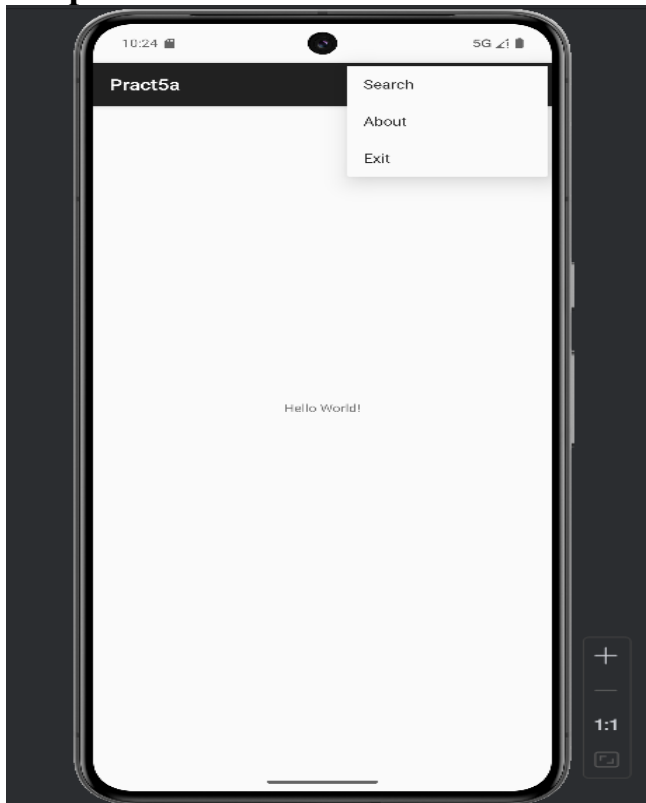
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater=getMenuInflater();
        inflater.inflate(R.menu.my_menu,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if(item.getItemId()==R.id.search)
        {
            Toast.makeText(this, "Search Selected",
Toast.LENGTH_SHORT).show();
        }
        else if(item.getItemId()==R.id.about)
        {
            Toast.makeText(this, "About Selected",
Toast.LENGTH_SHORT).show();
        }
        else if(item.getItemId()==R.id.exit)
        {
            Toast.makeText(this, "Exit Selected",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(this, "No option Selected",
Toast.LENGTH_SHORT).show();
        }
        return super.onOptionsItemSelected(item);
    }
}

```

### Output:



### 6b) Dialog

*A dialog is a small window that prompts the user to make a decision or enter additional information. A dialog doesn't fill the screen and is normally used for modal events that require users to take an action before they can proceed.*

**Firstly, Change the theme.**

#### **activity\_main.xml**

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:layout_marginTop="100dp"
        android:onClick="onClickDialog"
        android:text="Click to display an AlertDialog...!!!"
        android:id="@+id/btnalert"/>
    <Button
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClickProgress"
        android:textSize="20dp"
        android:text="Click to display an ProgressDialog !"
        android:id="@+id/btnprog"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="19dp"
        android:onClick="onClickDatePicker"
        android:text="Click to display an DatePickerDialog!"
        android:id="@+id/btndate"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="19dp"
        android:onClick="onClickTimeDialog"
        android:text="Click to display an TimePickerDialog...!!!"
        android:id="@+id/btntime"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txtdate"
        android:textSize="30dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txttime"
        android:textSize="30dp"/>
</LinearLayout>
```

### MainActivity.java

```
package com.example.pract5b;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.app.TimePickerDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
```

```
import android.widget.DatePicker;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {
    CharSequence[] items={"Android","Security","Cloud"};
    boolean[] itemsChecked=new boolean[items.length];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
    public void onClickDialog(View v)
    {
        AlertDialog.Builder builder=new AlertDialog.Builder(this);
        builder.setTitle("This is dialog with simple Text....");
        builder.setPositiveButton("Enable", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getApplicationContext(),"Enable Clicked",
Toast.LENGTH_SHORT).show();
            }
        });
        builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getApplicationContext(),"Cancel Clicked",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```



```

    });
    builder.setMultiChoiceItems(items, itemsChecked, new
DialogInterface.OnMultiChoiceClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which, boolean
isChecked) {
            Toast.makeText(getBaseContext(),items[which]+(isChecked ?
"Checked" : "UnChecked"),Toast.LENGTH_SHORT).show();
        }
    });
    AlertDialog dialog=builder.create();
    dialog.show();
}
public void onClickProgress(View v)
{
    final ProgressDialog progressDialog=new ProgressDialog(this);

progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTA
L);
    progressDialog.setMessage("Please Wait.....");
    progressDialog.incrementProgressBy(20);
    progressDialog.setButton(Dialog.BUTTON_POSITIVE, "loading..",
new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            progressDialog.dismiss();
        }
    });
    progressDialog.show();
}
public void onClickDatePicker(View v)
{
    final int y,m,d;
    TextView txtdate=(TextView) findViewById(R.id.txtdate);
    DatePickerDialog datePickerDialog;
    Calendar c=Calendar.getInstance();
    y=c.get(Calendar.YEAR);
    m=c.get(Calendar.MONTH);
    d=c.get(Calendar.DAY_OF_MONTH);
    final DatePickerDialog.OnDateSetListener mdateset=new
DatePickerDialog.OnDateSetListener() {
        @Override

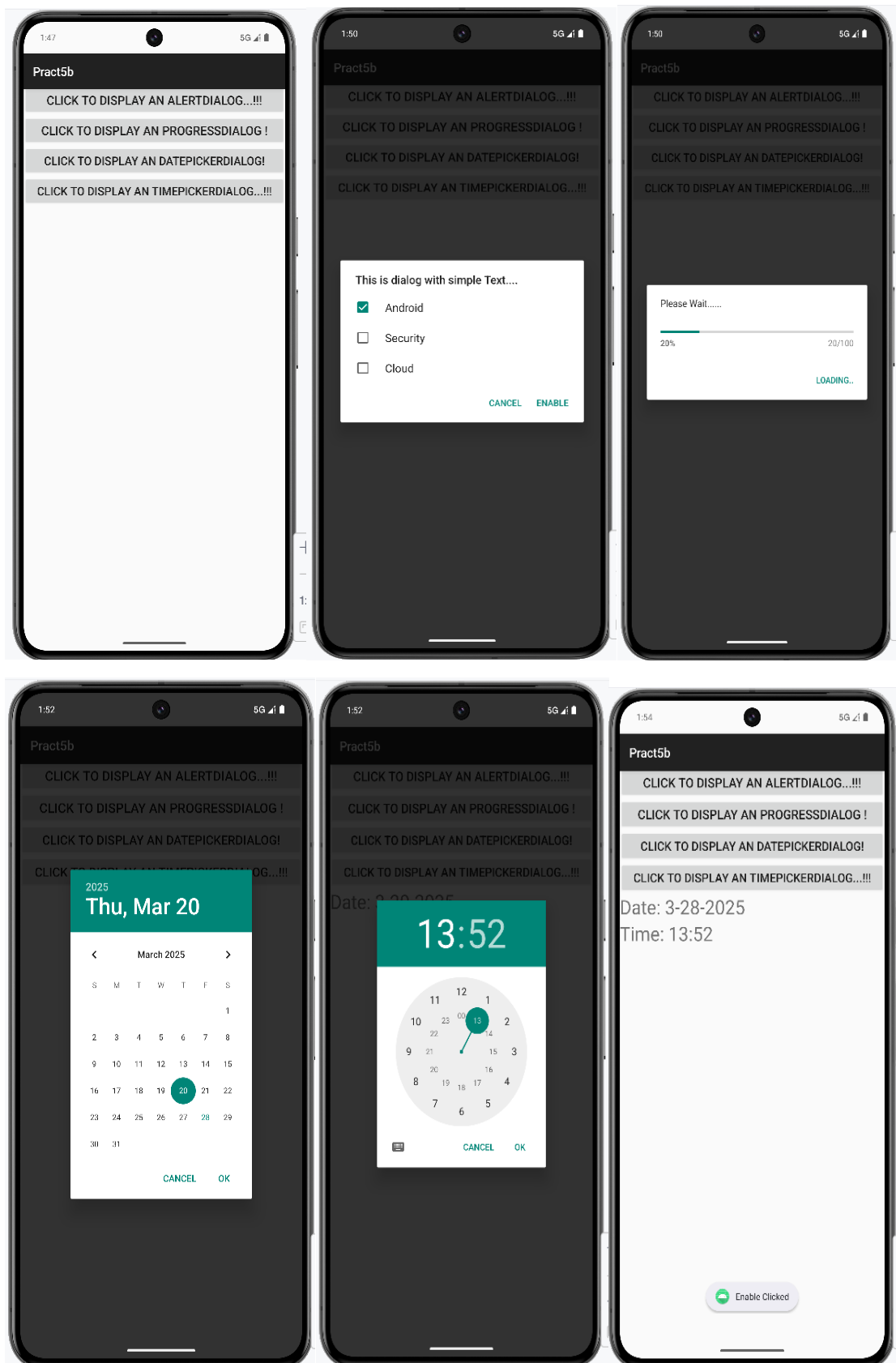
```

```

        public void onDateSet(DatePicker view, int year, int month, int
dayOfMonth) {
            txtdate.setText(new StringBuilder().append("Date:
").append(month+1).append("-").append(dayOfMonth).append("-
").append(year));
        }
    };
    datePickerDialog=new DatePickerDialog(this,mdateset,y,m,d);
    datePickerDialog.show();
}
public void onClickTimeDialog(View v)
{
    int h,m;
    final TextView txtdate=(TextView)findViewById(R.id.txttime);
    TimePickerDialog timePickerDialog;
    Calendar c=Calendar.getInstance();
    h=c.get(Calendar.HOUR_OF_DAY);
    m=c.get(Calendar.MINUTE);
    TimePickerDialog.OnTimeSetListener mtimeset=new
TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {
            txtdate.setText(new StringBuilder().append("Time:
").append(hourOfDay).append(":").append(minute));
        }
    };
    timePickerDialog=new TimePickerDialog(this,mtimeset,h,m,true);
    timePickerDialog.show();
}
}

```

### Output:



### 6c) Dialog fragments

*A DialogFragment is a special fragment subclass that is designed for creating and hosting dialogs. Although you don't need to host your dialog within a fragment, doing so lets the FragmentManager manage the state of the dialog and automatically restore the dialog when a configuration change occurs.*

#### **activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dialog Fragment"
        android:textSize="30dp"
        android:id="@+id/btndfrag"
        android:layout_centerInParent="true"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alert Fragment"
        android:textSize="30dp"
        android:layout_below="@+id/btndfrag"
        android:id="@+id/btnafrag"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

**Step1: res→new→ Resource File →"dfragment" →menu:Layout  
→Root element→ Relative Layout→ OK**

#### **dfragment.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:text="This is Dialog Fragment"
        android:textSize="30dp"/>
</RelativeLayout>
```

**Step2: Java->right click on example->new java class->**

**“DFragment”**

**DFragment.java**

```
package com.example.pract5c;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.DialogFragment;

public class DFragment extends DialogFragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
        @Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
    {
        View rootView=inflater.inflate(R.layout.dfragment,container,false);
        getDialog().setTitle("DialogFragment Test");
        return rootView;
    }
}
```

**Step3: Java->right click on example->new java file->**

**“AlertDialogFragment”**

### AlertDialogFragment.java

```
package com.example.pract5c;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.DialogFragment;

public class AlertDialogFragment extends DialogFragment {
    @NonNull
    @Override
    public Dialog onCreateDialog(@Nullable Bundle savedInstanceState)
    {
        return new
        AlertDialog.Builder(getActivity()).setIcon(android.R.mipmap.sym_def_a
pp_icon).setTitle("Alert Dialog Fragment").setMessage("Alert Dialog
Example....!").setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getContext(),"onCkicked",
Toast.LENGTH_SHORT).show();
            }
        })
        .setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(getContext(),"Cancel
Clicked",Toast.LENGTH_SHORT).show();
            }
        })
        .create();
    }
}
```

## MainActivity.java

```
package com.example.pract5c;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.fragment.app.FragmentManager;

public class MainActivity extends AppCompatActivity {
    Button btndfrag, btnafrag;
    FragmentManager fm=getSupportFragmentManager();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        btndfrag=(Button)findViewById(R.id.btndfrag);
        btnafrag=(Button)findViewById(R.id.btnafrag);

        btndfrag.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                DFragment df=new DFragment();
                df.show(fm,"Dialog Fragment");
            }
        });

        btnafrag.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialogFragment adf=new AlertDialogFragment();
                adf.show(fm,"Alert Dialod Fragment");
            }
        });
    }
}
```

Output:





## Practical 7

### Aim : Programs on Intents, Events, Listeners and Adapters

#### The Android Intent Class, Using Events and Event Listeners

- ✓ *Android Intent Class: Used for communication between components (activities, services, etc.).*
- ✓ *Events and Event Listeners: Handling user interactions like clicks, gestures, and touch events.*
- ✓ *Adapters: Bridging data sources (arrays, databases, etc.) with UI components like ListView and RecyclerView.*

#### activity\_main.xml

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter a Text"
        android:textSize="30dp"
        android:layout_marginTop="100dp"
        android:id="@+id/txt"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:id="@+id/btnsubmit"
        android:onClick="show"
        android:textSize="30dp"
        android:layout_below="@+id/txt"/>
</RelativeLayout>
```

**Step1: Java->new->Activity->Empty view activity->"MainActivity2"**

#### activity\_main2.xml

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:textSize="30dp"
android:layout_marginTop="100dp"
android:id="@+id/txt2"/>
```

</RelativeLayout>

### **MainActivity.java**

```
package com.example.pract6;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

    }
    public void show(View v)
    {
        EditText ed=(EditText) findViewById(R.id.txt);
        String s=ed.getText().toString();
        Intent in=new Intent(this, MainActivity2.class);
        in.putExtra("My key",s);
        startActivity(in);
    }
}
```

### **MainActivity2.java**

```
package com.example.pract6;
import android.os.Bundle;
import android.widget.TextView;
```

```
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main2);
        TextView tv=(TextView) findViewById(R.id.txt2);
        String s1=getIntent().getExtras().getString("My key");
        tv.setText("Value: "+s1);
    }
}
```

### Output:



## Practical 8

### Aim: Programs on Services, notification and broadcast receivers

#### 8a) Programs on Services.

*Services in Android are a special component that facilitates an application to run in the background in order to perform long-running operation tasks. The prime aim of a service is to ensure that the application remains active in the background so that the user can operate multiple applications at the same time.*

#### activity\_main.xml

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="Service Start"
        android:id="@+id/btnstart"
        android:layout_marginTop="100dp"
        android:onClick="startService"
        android:layout_centerHorizontal="true"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="Service Stopped"
        android:id="@+id/btnstop"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:onClick="stopService"
        android:layout_below="@+id/btnstart"/>
</RelativeLayout>
```

#### MainActivity.java

```
package com.example.pract7a;
import android.content.Intent;
import android.os.Bundle;
```

```
import android.view.View;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

    public void startService(View v)
    {
        Intent in=new Intent(this, MyService.class);
        startService(in);
    }

    public void stopService(View v)
    {
        Intent in=new Intent(this, MyService.class);
        stopService(in);
    }
}
```

### **MyService.java**

```
package com.example.pract7a;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.widget.Toast;
import androidx.annotation.Nullable;

public class MyService extends Service {
    @Override
    public void onCreate() {
```

```

        super.onCreate();
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Toast.makeText(this, "Service Started",
            Toast.LENGTH_SHORT).show();
        return START_STICKY;
    }
    @Override
    public void onDestroy() {
        Toast.makeText(this, "Service Stopped",
            Toast.LENGTH_SHORT).show();
    }
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}

```

## AndroidManifest.xml

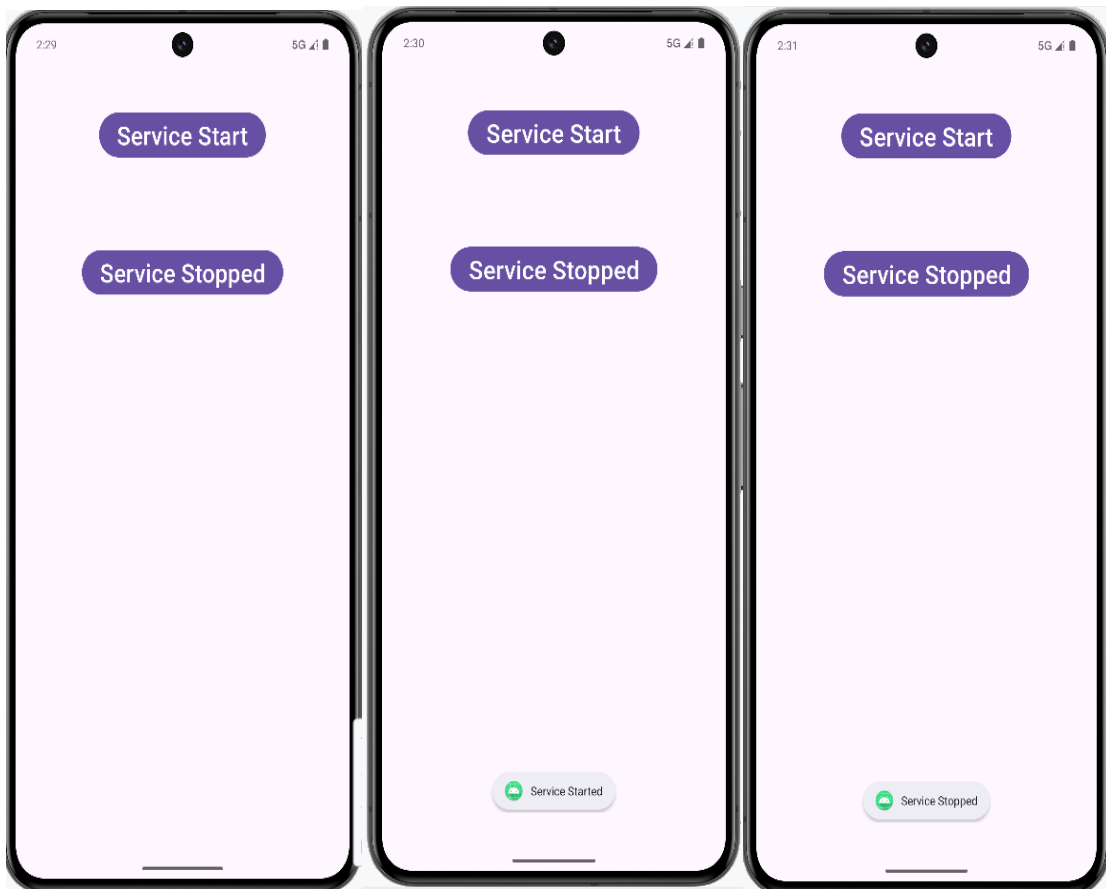
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:tools="http://schemas.android.com/tools">
4
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_rules"
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="Pract7A"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportsRtl="true"
13         android:theme="@style/Theme.Pract7A"
14         tools:targetApi="31">
15         <activity
16             android:name=".MainActivity"
17             android:exported="true">
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN" />
20
21                 <category android:name="android.intent.category.LAUNCHER" />
22             </intent-filter>
23         </activity>
24         <service android:name=".MyService" android:exported="false"/>
25     </application>
26
27 </manifest>

```

In "AndroidManifest.xml" file add the selected part in the already existing code.

**Output:**



### 8b) Programs on notification receivers.

*A Notification Receiver in Android is typically implemented using a BroadcastReceiver to listen for system or app-specific notifications and trigger actions accordingly*

#### **activity\_main.xml**

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notify Me"
        android:textSize="30dp"
        android:onClick="displayNotification"
```

```
        android:layout_centerHorizontal="true"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

### MainActivity.java

```
package com.example.pract8a;
import android.Manifest;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.RequiresPermission;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    public final String CHANNEL_ID="Personal_NOTIFICATION";
    public final int id=001;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
    @RequiresPermission(Manifest.permission.POST_NOTIFICATIONS)
    public void displayNotification(View v) {
        createNotificationchannel();
        Toast.makeText(getApplicationContext(), "Hiee",
        Toast.LENGTH_SHORT).show();
        NotificationCompat.Builder builder = new
        NotificationCompat.Builder(this, CHANNEL_ID);
```

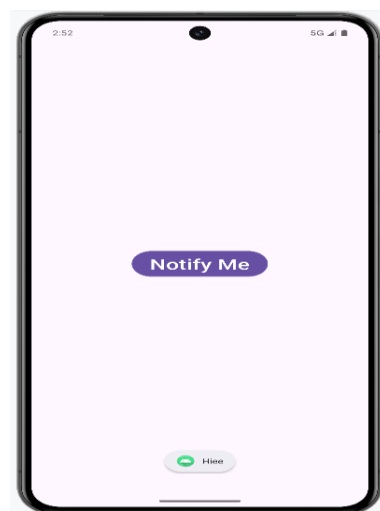
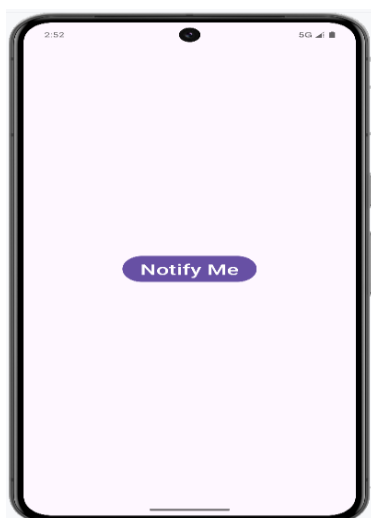


```

        builder.setSmallIcon(R.mipmap.ic_launcher);
        builder.setContentTitle("Simple Notification");
        builder.setContentText("This is a Text Notification");
        builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
        NotificationManagerCompat
notificationManagerCompat=NotificationManagerCompat.from(this);
        notificationManagerCompat.notify(id,builder.build());
    }
    private void createNotificationchannel()
    {
        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.O)
        {
            CharSequence name="Personal_Notification";
            String dis="This is a Text Description";
            int importance=NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel notificationChannel=new
NotificationChannel(CHANNEL_ID,name,importance);
            notificationChannel.setDescription(dis);
            NotificationManager notificationManager=(NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
            notificationManager.createNotificationChannel
(notificationChannel);
        }
    }
}

```

### Output:



### 8c) Programs on broadcast receivers.

*A BroadcastReceiver in Android is a component that listens for broadcasted messages (intents) from the system or other apps and reacts accordingly.*

#### **activity\_main.xml**

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Call Log"
        android:textSize="30dp"
        android:layout_marginTop="50dp"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:id="@+id/txtnum"/>
</LinearLayout>
```

#### **MainActivity.java**

```
package com.example.pract7c;
import android.os.Bundle;
import android.widget.TextView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    TextView txtnum;
    private static MainActivity instance;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

EdgeToEdge.enable(this);
setContentView(R.layout.activity_main);
instance = this;
}
public static MainActivity getInstance() {
    return instance;
}
public void myMethod(String s)
{
    txtnum=(TextView) findViewById(R.id.txtnum);
    txtnum.setText(s);
}
}

```

### NumberReceiver.java

```

package com.example.pract7c;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telecom.TelecomManager;
import android.telephony.TelephonyManager;
public class NumberReceiver extends BroadcastReceiver {
    static String allnum="";
    @Override
    public void onReceive(Context context, Intent intent) {
        String
state=intent.getStringExtra(TelephonyManager.EXTRA_STATE);
        if(state.equals(TelephonyManager.EXTRA_STATE_RINGING))
        {
            String
number=intent.getExtras().getString(TelephonyManager.EXTRA_INCOM
ING_NUMBER);
            allnum=allnum+"\n"+number;
            MainActivity.getInstance().myMethod(allnum);
        }
    }
}

```

## AndroidManifest.xml

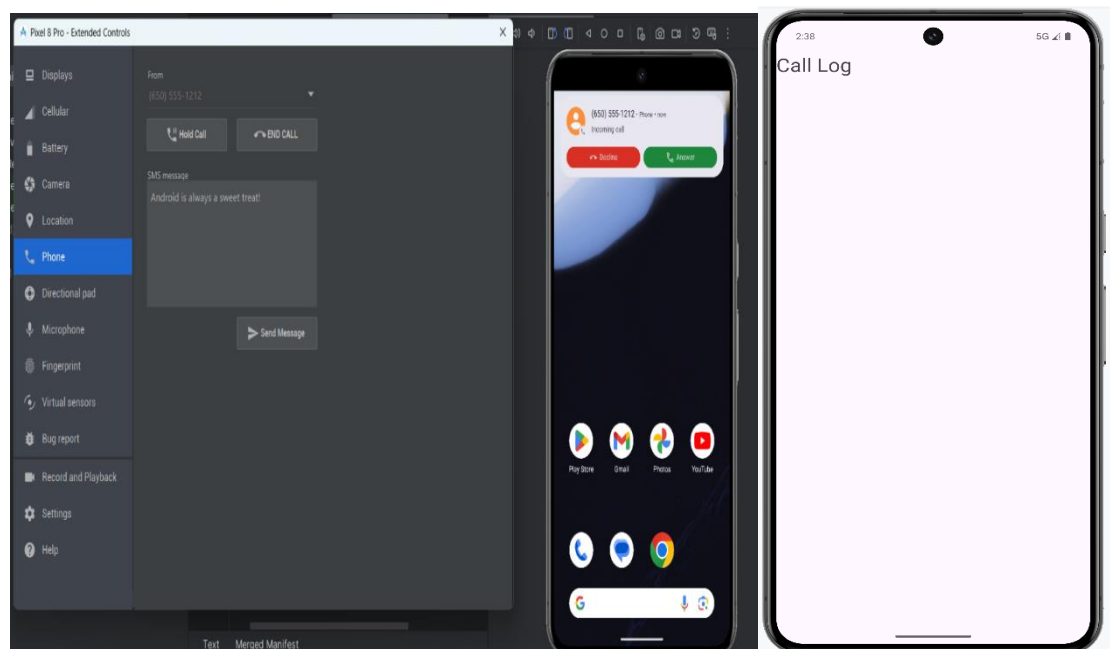
Include 2<sup>nd</sup> and 26<sup>th</sup> line in the existing code.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4      <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_rules"
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="@string/app_name"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportsRtl="true"
13         android:theme="@style/Theme.Pract7c"
14         tools:targetApi="31">
15         <activity
16             android:name=".MainActivity"
17             android:exported="true">
18             <intent-filter>
19                 <action android:name="android.intent.action.MAIN" />
20                 <category android:name="android.intent.category.LAUNCHER" />
21             </intent-filter>
22         </activity>
23         <receiver android:name=".NumberReceiver"
24             android:exported="true">
25             <intent-filter>
26                 <action android:name="android.intent.action.PHONE_STATE"/>
27             </intent-filter>
28         </receiver>
29     </application>
30 </manifest>

```

## Output:



## Practical 9

**Aim: a) Database Programming with SQLite**

**Steps to Download SQLite:**

**1: Open Browser ->SQLite Download for Windows**

**2:**

### Downloads

(Please consider sponsoring us on Patreon 😊)

### Windows

Our latest release (3.13.1) for Windows:

- [DB Browser for SQLite - Standard installer for 32-bit Windows](#)
- [DB Browser for SQLite - .zip \(no installer\) for 32-bit Windows](#)
- [DB Browser for SQLite - Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite - .zip \(no installer\) for 64-bit Windows](#)

**Download DB Browser for SQLite- Standard installer for 64-bit Windows**

**3: And install it.**

**activity\_main.xml**

```
<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Log In"
        android:textSize="30dp"
        android:layout_marginTop="100dp"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:hint="Enter username"
        android:id="@+id/txtusername"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="30dp"
    android:hint="Enter Password"
    android:inputType="textShortMessage"
    android:id="@+id/txtpswd"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        android:textSize="20dp"
        android:onClick="register"
        android:id="@+id/btnreg"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:textSize="20dp"
        android:onClick="Login"
        android:id="@+id/btnlog"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Drop"
        android:textSize="20dp"
        android:onClick="drop"
        android:id="@+id/btndrop"/>
</LinearLayout>

</LinearLayout>
```

### MainActivity.java

```
package com.example.pract8b;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
```

```

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    EditText txtuname,txtpaswd;
    Button btnreg,btnlog,btndrop;
    myDB mdb;
    SQLiteDatabase db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        txtuname=(EditText) findViewById(R.id.txtusname);
        txtpaswd=(EditText) findViewById(R.id.txtpswd);
        btnreg=(Button) findViewById(R.id.btnreg);
        btnlog=(Button) findViewById(R.id.btnlog);
        btndrop=(Button) findViewById(R.id.btndrop);
        mdb=new myDB(this);
    }
    public void register(View v)
    {
        String un=String.valueOf(txtuname.getText());
        String ps=String.valueOf(txtpaswd.getText());
        db=mdb.getWritableDatabase();
        String q="insert into Login values('"+un+"','"+ps+"')";
        db.execSQL(q);
        Toast.makeText(getApplicationContext(),"You have Registered
        Successfully",Toast.LENGTH_SHORT).show();
    }
    public void Login(View v)

```

```

{
    String un=String.valueOf(txtuname.getText());
    String ps=String.valueOf(txtpaswd.getText());
    db=mdb.getReadableDatabase();
    String q="select * from Login where username='"+un+"' and
password='"+ps+"'";
    try{
        Cursor c=db.rawQuery(q,null);
        if(c.getCount()==0)
        {
            Toast.makeText(getApplicationContext(), "Username and
password is Invalid, Do check again!", Toast.LENGTH_SHORT).show();
        }
        else{
            while(c.moveToNext())
            {
                String uname = c.getString(0);
                String passwd = c.getString(1);
                Toast.makeText(getApplicationContext(), "Username: "
+ uname + "Password: " + passwd, Toast.LENGTH_SHORT).show();
                if (un.contentEquals(uname) &&
ps.contentEquals(passwd)) {
                    Toast.makeText(getApplicationContext(), "Welcome
User: " + un, Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
    catch(SQLiteException ex)
    {
        ex.printStackTrace();
    }
}

public void drop(View v)
{
    db=mdb.getWritableDatabase();
    mdb.onUpgrade(db,1,2);
    Toast.makeText(getApplicationContext(),"All User
delete",Toast.LENGTH_SHORT).show();
}

}

```



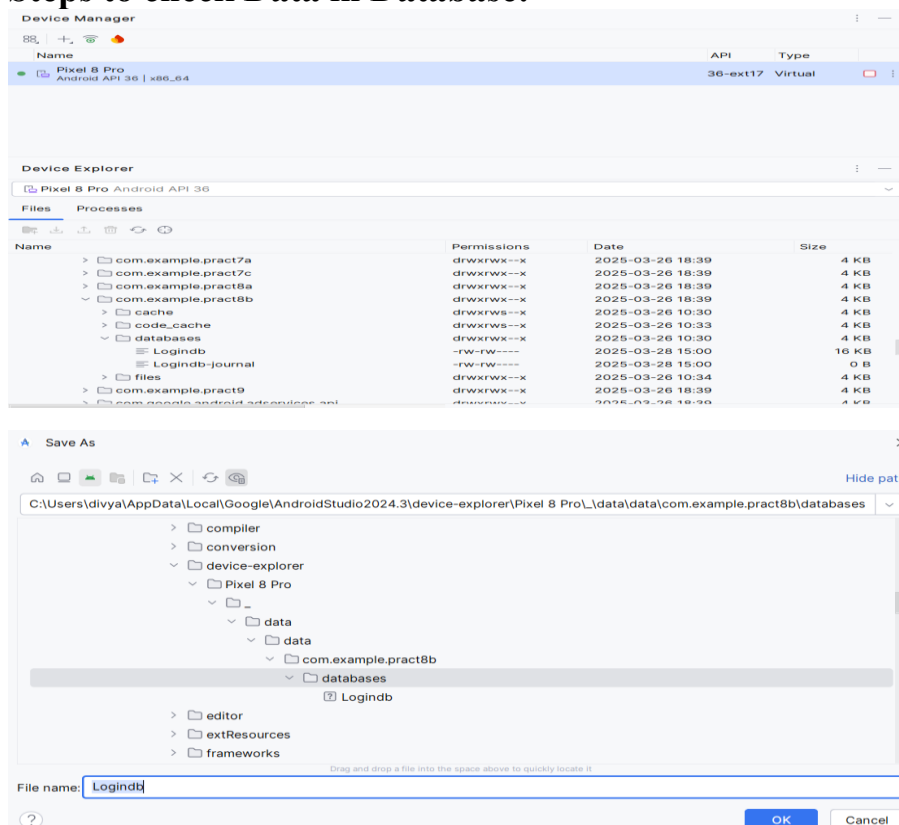
```

class myDB extends SQLiteOpenHelper
{
    myDB(Context c)
    {
        super(c,"Logindb",null,1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String str="create table Login(username text,password text)";
        db.execSQL(str);

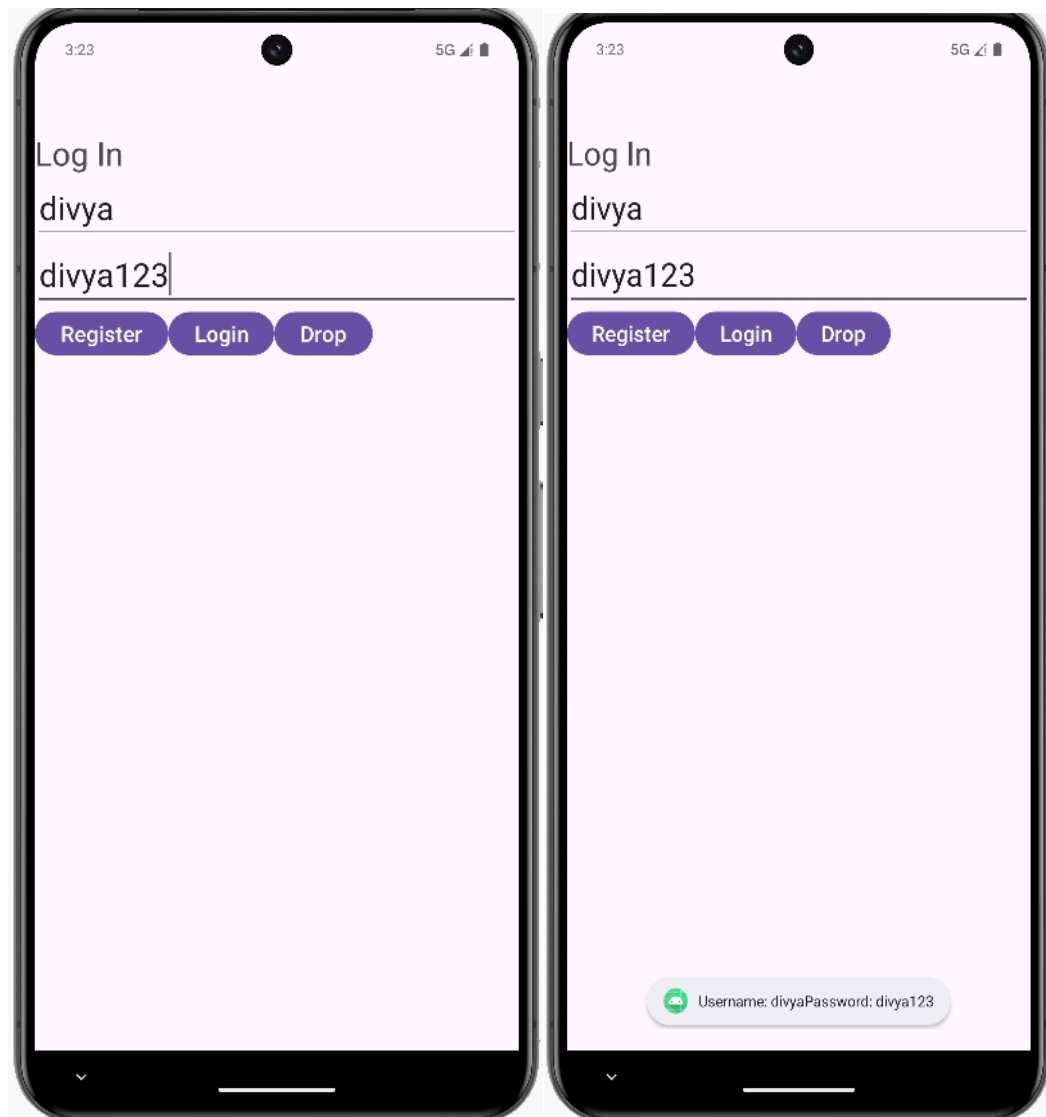
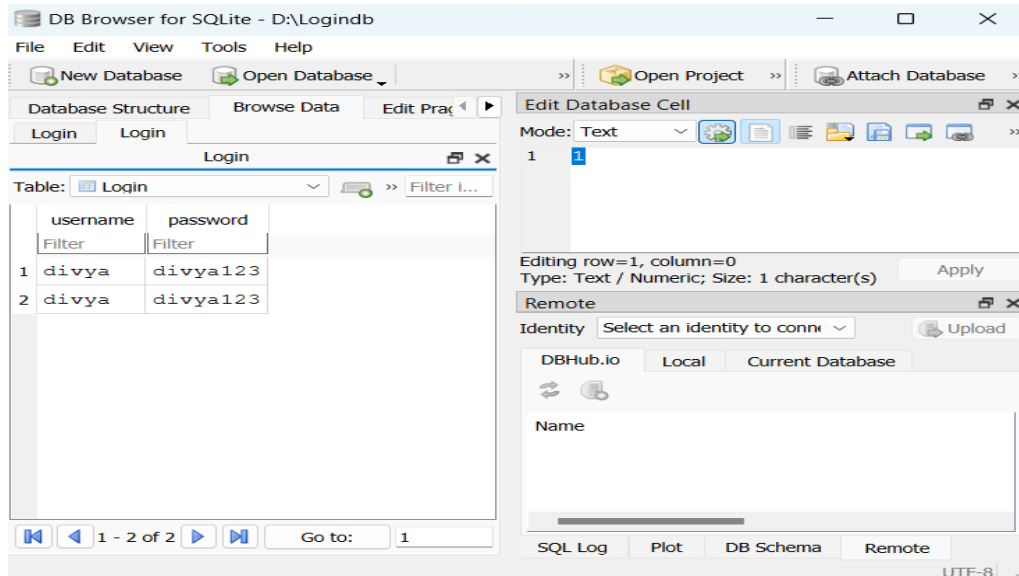
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
        String dr="drop table if exists Login";
        db.execSQL(dr);
        onCreate(db);

    }
}
    
```

### Steps to check Data in Database:



## Android Programming Practical



## Practical 10

### Aim: Programming Media API and Telephone API

#### Programming Media API

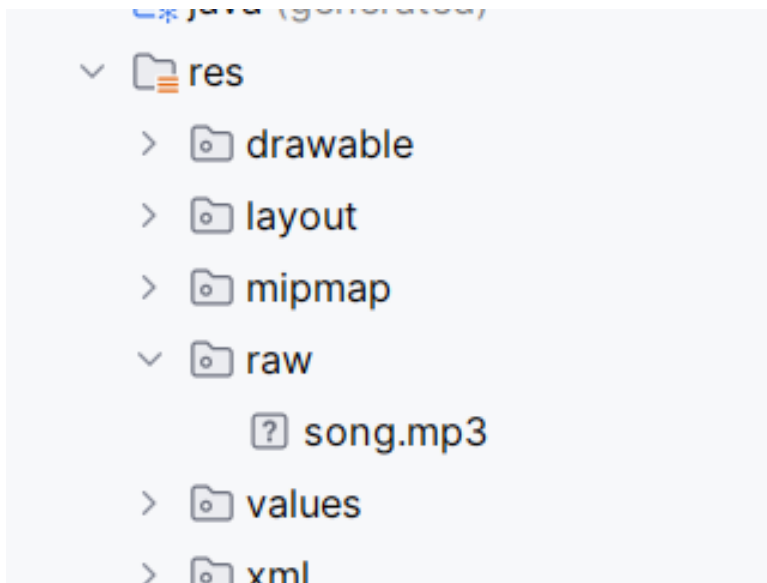
- Type I

##### activity\_main.xml

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"
        android:textSize="30dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:id="@+id/btnstart"/>
</RelativeLayout>
```

**Step1 : Download an MP3 audio or use your audio file from the respected System.**

**Step2 :Copy the .mp3 file and paste in res ->raw folder**



## MainActivity.java

```
package com.example.pract9;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    Button btnstart;
    MediaPlayer mediaPlayer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        btnstart=(Button)findViewById(R.id.btnstart);
        btnstart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

mediaPlayer=MediaPlayer.create(getApplicationContext(),R.raw.song);
                mediaPlayer.start();
                btnstart.setEnabled(false);
                Toast.makeText(getApplicationContext(), "Playing.....",
Toast.LENGTH_SHORT).show();
                mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                    @Override
                    public void onCompletion(MediaPlayer mp) {
```

```

        mediaPlayer.release();
        mediaPlayer=null;
        Toast.makeText(getApplicationContext(), "Song Stopped
Playing...", Toast.LENGTH_SHORT).show();
        btnstart.setEnabled(true);
    }
});

}

});

}

}

```

- Type II

### activity\_main.xml

```

<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"
        android:textSize="30dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:id="@+id/btnstart"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop"
        android:textSize="30dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:layout_below="@+id/btnstart"
        android:id="@+id/btnstop"/>
</RelativeLayout>

```

### MainActivity.java

```

package com.example.pract9;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    Button btnstart,btnstop;
    MediaPlayer mediaPlayer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        btnstart=(Button)findViewById(R.id.btnstart);
        btnstop=(Button)findViewById(R.id.btnstop);
        btnstart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

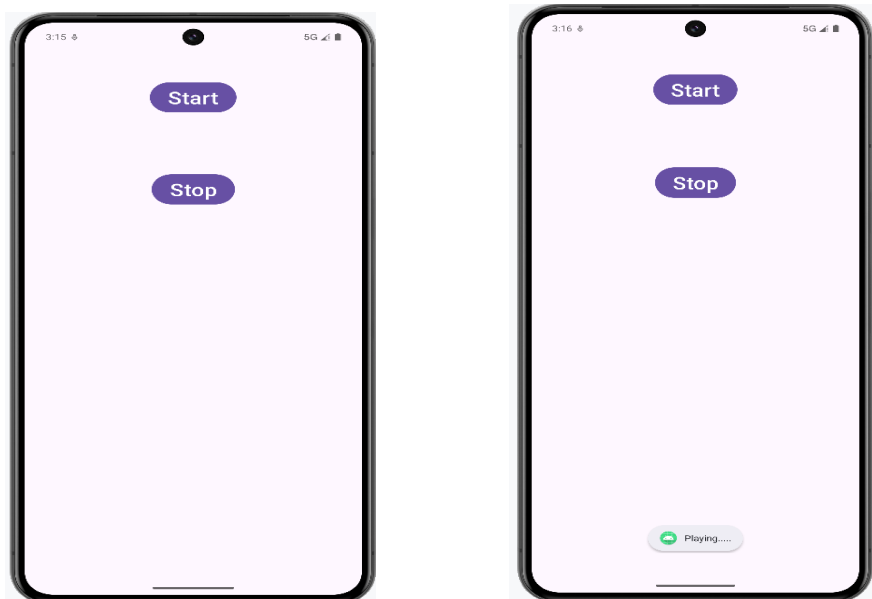
mediaPlayer=MediaPlayer.create(getApplicationContext(),R.raw.song);
                mediaPlayer.start();
                Toast.makeText(getApplicationContext(), "Playing.....",
Toast.LENGTH_SHORT).show();
            }
        });
        btnstop.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                    @Override
                    public void onCompletion(MediaPlayer mp) {
                        mediaPlayer.stop();
                        mediaPlayer.release();

```

```
        mediaPlayer=null;
        Toast.makeText(getApplicationContext(), "Playing
Stopped....", Toast.LENGTH_SHORT).show();
    }
    });
}
});
}
```

Output:



### Programming Telephone API

#### activity\_main.xml

```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnncall"
        android:text="Call"
        android:textSize="20dp"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

## MainActivity.java

```
package com.example.pract9b;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        btn=(Button) findViewById(R.id.btncall);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intent.ACTION_CALL);
                intent.setData(Uri.parse("tel:1212"));
                if (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CALL_PHONE)!=PackageManager.PERMISSION_
GRANTED)
                {
                    return;
                }
            }
        });
    }
}
```



```
startActivity(callIntent);
    }

    });

}
}
```

### AndroidManifest.xml

#### Add 7<sup>th</sup> line in “AndroidManifest.xml”

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4      <uses-feature
5          android:name="android.hardware.telephony"
6          android:required="false" />
7      <uses-permission android:name="android.permission.CALL_PHONE"/>
8      <application
9          android:allowBackup="true"
10         android:dataExtractionRules="@xml/data_extraction_rules"
11         android:fullBackupContent="@xml/backup_rules"
12         android:icon="@mipmap/ic_launcher"
13         android:label="Pract9b"
14         android:roundIcon="@mipmap/ic_launcher_round"
15         android:supportsRtl="true"
16         android:theme="@style/Theme.Pract9b"
17         tools:targetApi="31">
18         <activity
19             android:name=".MainActivity"
20             android:exported="true">
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23
24                 <category android:name="android.intent.category.LAUNCHER" />
25             </intent-filter>
26         </activity>
27     </application>
28 </manifest>
```

**To run the application we need two AVDs, so Install !**

**Either Enter full mobile number or last 4 digit to make a call. We are using full mobile number .**

**To get a full mobile number . Go to Setting→ System→ About emulated device→ Status→ SIM→ status**

**Enable Phone Permissions:**

**Setting→ App & notifications → TelephonyTest(Name of your App) → Permission→ Enable Phone Permission.**

**Start two emulators. Press Call Button.**

**Output:**

