

1. Creating Hello World Application

Steps:

1. Click New Project, the New Project Dialog box appears.
2. Choose Empty Views Activity then click Next.
3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.
4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of 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!"
        android:textSize="40sp"
        android:textStyle="bold|italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

=>Coding part of MainActivity.java

```
package com.example.loginapplication;

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);
    }
}
```

2. Creating an Application that displays message based on screen orientation

Step 1: Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.

4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of 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:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Portrait"
        android:textSize="40sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.25" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Launch new activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.499" />

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

=>Coding part of MainActivity.java

```
package com.example.orientation;
```

```
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 onClick(View v){
        Intent intent=new Intent(MainActivity.this,Nextactivity.class);
        startActivity(intent);
    }

}

```

Step 2: Create another new empty views activity and give the name as Nextactivity (Go to app>>New>>Activity>>Empty Views Activity)

=>Coding part of Activity_nextactivity.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=".Nextactivity">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Landscape Orientation"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Step 3: Add the screen orientation values in AndroidManifest.xml

=>Coding part of AndroidManifest.xml

```

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

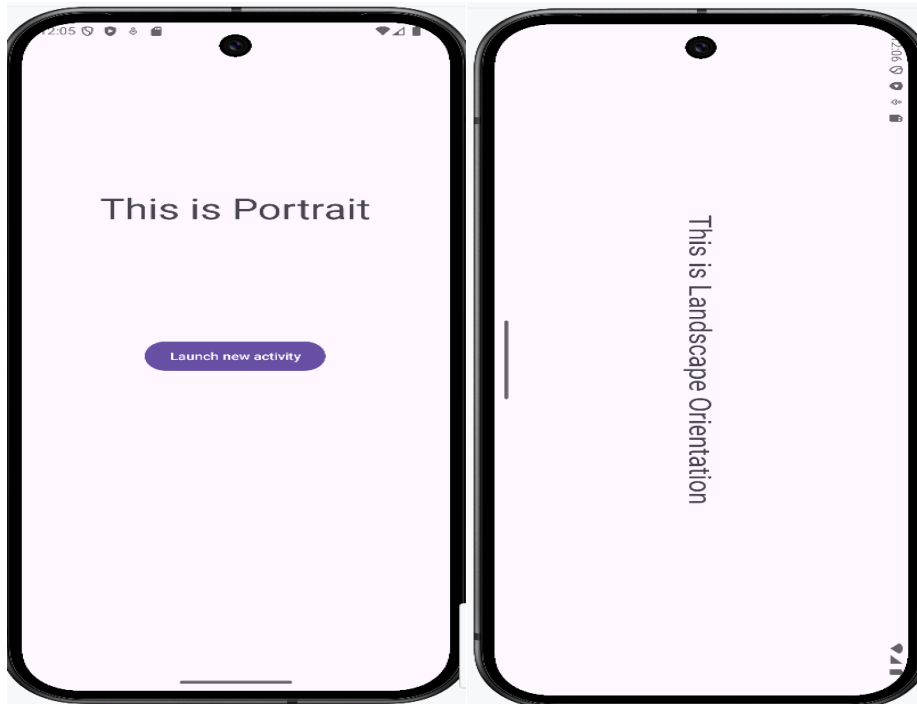
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Orientation"
        tools:targetApi="31">
        <activity
            android:name=".Nextactivity"
            android:exported="false"
            android:screenOrientation="landscape"

            />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:screenOrientation="portrait"
            >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```

=>Output of the program



3. Create and Application to develop Login window using UI controls

Step 1: Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.

4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of Activity_main.xml

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

    <TextView
        android:id="@+id/tvTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="200dp"
        android:text="Login Form"
        android:textSize="40sp" />
```

```

<TextView
    android:id="@+id/tvUserName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="User Name"
    android:textSize="24sp" />

<EditText
    android:id="@+id/etUserName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter the User Name"
    android:inputType="text"
    android:textSize="24sp" />

<TextView
    android:id="@+id/tvPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password"
    android:textSize="24sp" />

<EditText
    android:id="@+id/etPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter the Password"
    android:inputType="textPassword"
    android:textSize="24sp" />

<Button
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Login"
    android:layout_gravity="center" />

</LinearLayout>

```

=>Coding part of MainActivity.java

```

package com.example.loginapplication;

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);
    }
}

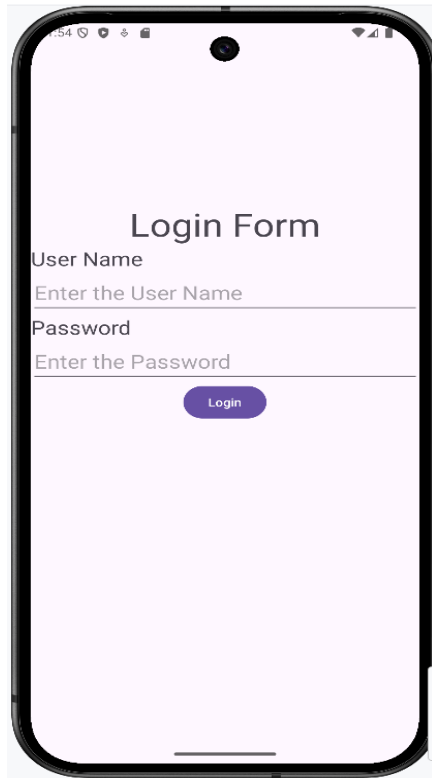
```

```

        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
}

```

=>Output of the program



4. Create and Application to implement new activity using explicit intent and implicit intent

Step 1: Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.

4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of Activity_main.xml

```

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

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Explicit Activity"
    android:textSize="30sp"
    android:layout_gravity="center"
    android:layout_marginTop="200dp"
/>

<Button
    android:id="@+id/btnExplicitContent"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="200dp"
    android:onClick="onClick"
    android:text="Explicit Activity" />

</LinearLayout>

```

=> Coding part of MainActivity.java

```

package com.example.implicitexplicit;
import android.content.Intent;
import android.view.View;
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;
        });
    }

    public void onClick(View view) {

```



```

        Intent intent = new Intent(MainActivity.this, NextActivity.class);
        startActivity(intent);
    }
}

```

Step 2: Create another new empty views activity and give the name as Nextactivity (Go to app>> New>>Activity>>Empty Views Activity)

Coding part of activity_next.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".NextActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Implicit Activity"
        android:textSize="30sp"
        android:layout_gravity="center"
        android:layout_marginTop="200dp"
        />

    <Button
        android:id="@+id/btnImplicitContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="200dp"
        android:onClick="onClick"
        android:text="Implicit Activity" />

</LinearLayout>

```

=>Coding part of NextActivity.java

```

package com.example.implicitexplicit;

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 android.content.Intent;
import android.net.Uri;
import android.view.View;

```

```

public class NextActivity extends AppCompatActivity {

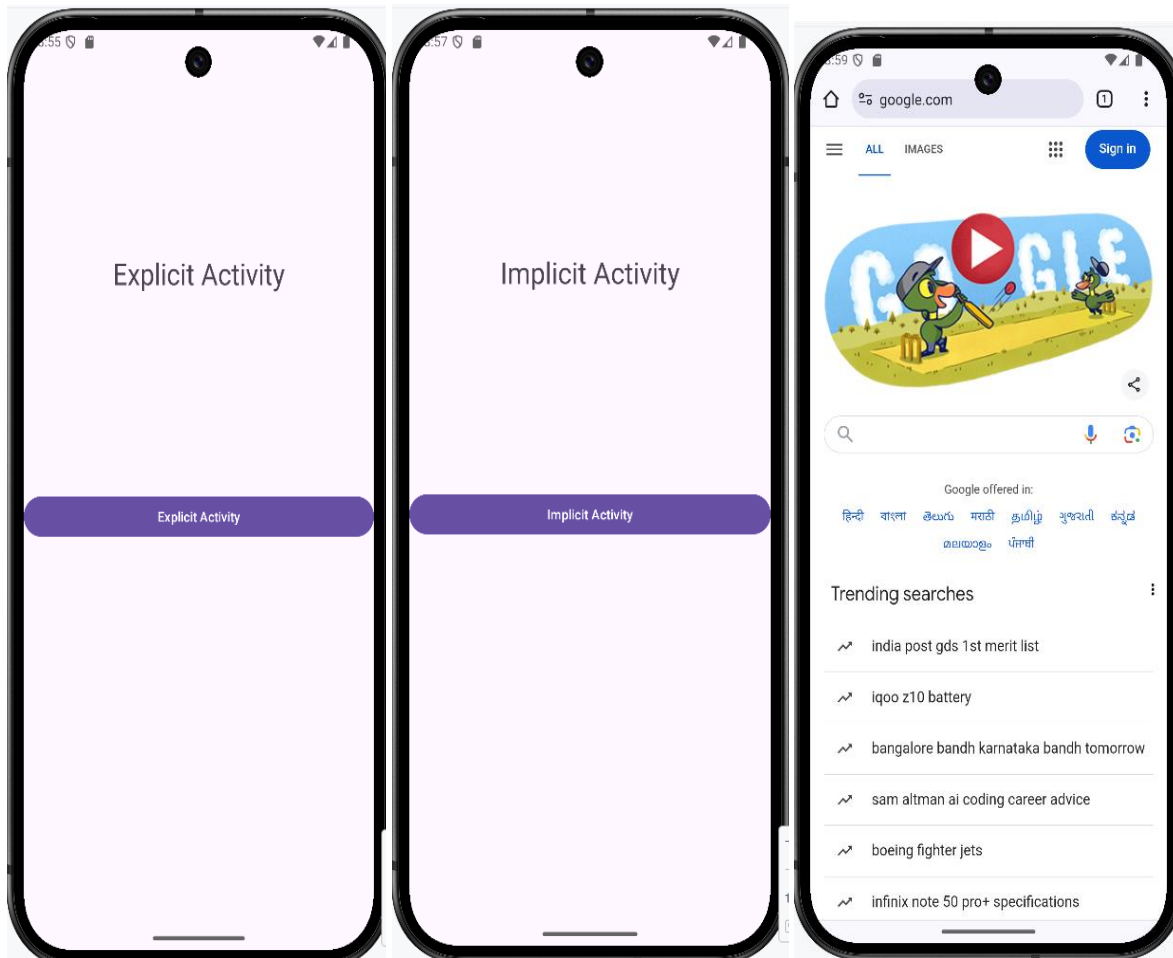
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_next);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }

    public void onClick(View view) {
        Uri webpage= Uri.parse("http://www.google.com");
        Intent intent = new Intent(Intent.ACTION_VIEW, webpage);
        startActivity(intent);
    }

}

```

=>Output of the program



6. Create and Application to show all views

Steps:

1. Click New Project, the New Project Dialog box appears.
2. Choose Empty Views Activity then click Next.
3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.
4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of Activity_main.xml

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

    <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a TextView"
    android:textSize="18sp"
    android:layout_marginBottom="16dp"/>
```

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your text here"
    android:layout_marginBottom="16dp" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me"
    android:layout_marginBottom="16dp" />
```

```
<CheckBox
    android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Check Me"
    android:layout_marginBottom="16dp" />
```

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Which Section You belong to"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>
```

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginBottom="16dp" />
```

```
<RadioButton
    android:id="@+id/radioButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="A section" />
```

```
<RadioButton
    android:id="@+id/radioButton2"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="B Section" />

<Switch
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Switch Testing"
    android:layout_marginBottom="16dp"/>
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select your favourite Programming Language"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>
<Spinner
    android:id="@+id/spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"/>
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Progress Bar"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:progress="25"
    android:layout_marginBottom="16dp" />
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select the number between 1 to 50"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>
<NumberPicker
    android:id="@+id/numberPicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"/>
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select your date of birth"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>

```

```

<DatePicker
    android:id="@+id/datePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"/>
<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Set the next class hour time"
    android:textSize="20sp"
    android:layout_marginBottom="16dp"/>
<TimePicker
    android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"/>

</LinearLayout>

</ScrollView>

```

=>Coding part of MainActivity.java

```

package com.example.mainactivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

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 java.util.ArrayList;
import android.widget.NumberPicker;

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;
        });
        Spinner spinner = findViewById(R.id.spinner);
        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("Select Language");
    }
}

```

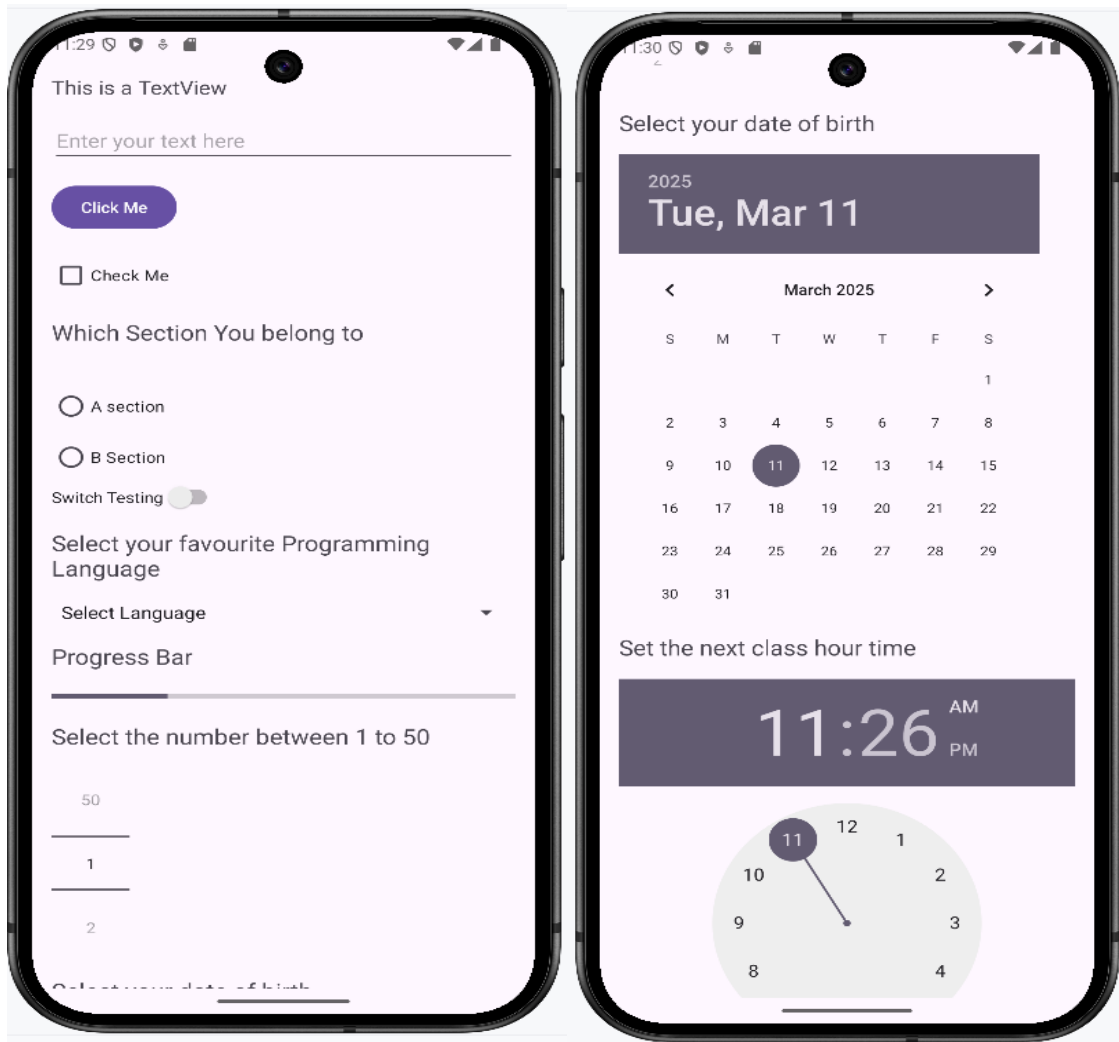
```
    arrayList.add("JAVA");
    arrayList.add("ANDROID");
    arrayList.add("C Language");
    arrayList.add("CPP Language");
    arrayList.add("Python Programming");

    ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, arrayList);
    arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(arrayAdapter);

    NumberPicker numberPicker1 = findViewById(R.id.numberPicker1);
    numberPicker1.setMinValue(1);
    numberPicker1.setMaxValue(50);

}
}
```

=> Output of the program



7 Create a menu in application

Steps:

1. Click New Project, the New Project Dialog box appears.
2. Choose Empty Views Activity then click Next.
3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.

Steps to be followed to create the menu

Step1. Create Android resource directory by clicking res>>new>>Android Resource directory, give the name as menu

Step2. Right click menu folder click new>> Menu Resource File, give the name of the file as menus.

Add the following code in menus.xml


```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/php" android:title="PHP"/>
  <item android:id="@+id/java" android:title="JAVA"/>
  <item android:id="@+id/csharp" android:title="C#"/>
</menu>
```

Step 3. =>Coding part of Activity_main.xml (Remove Helloworld textview control)

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

Step 4. =>Coding part of MainActivity.java

```
package com.example.menuapp;
```

```
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 android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
```

```
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 onCreatePanelMenu(int featureId, Menu menu)
  {
    MenuInflater inflater=getMenuInflater();
    inflater.inflate(R.menu.menus,menu);
  }
}
```

```

        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        if(item.getItemId()==R.id.php) {
            Toast.makeText(this, "Php Page", Toast.LENGTH_SHORT).show();
        }
        if(item.getItemId()==R.id.java) {
            Toast.makeText(this, "Java Page", Toast.LENGTH_SHORT).show();
        }
        if(item.getItemId()==R.id.csharp) {
            Toast.makeText(this, "C# Page", Toast.LENGTH_SHORT).show();
        }
        return true;
    }
}

```

Step 5. Change the theme value AndroidManifest.xml file

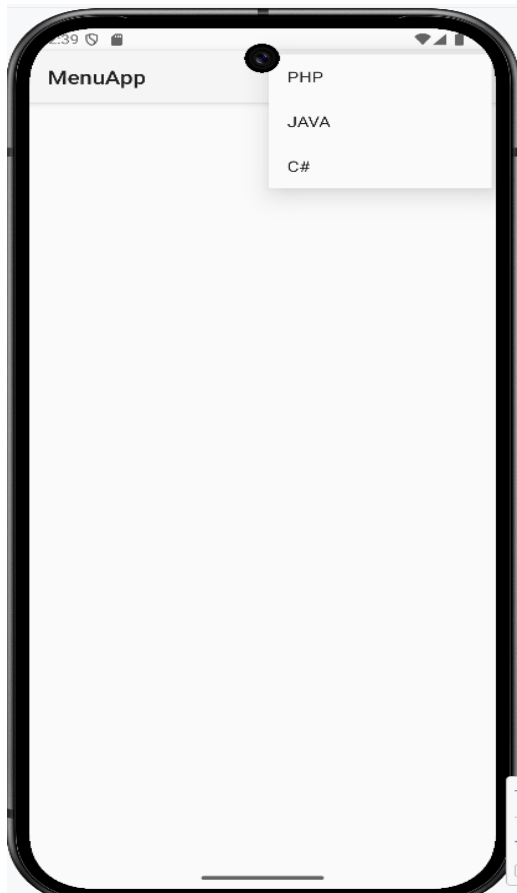
```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```

=>Output of the program



11. Create an application to send Email

Steps:

1. Click New Project, the New Project Dialog box appears.
2. Choose Empty Views Activity then click Next.
3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.
4. Update the following code in activity_main.xml and MainActivity.java

=>Coding part of Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/etTo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To"/>
```

```

<EditText
    android:id="@+id/etSubject"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/etTo"
    android:hint="Subject"/>
<EditText
    android:id="@+id/etMessage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/etSubject"
    android:hint="Message"/>
<Button
    android:id="@+id/btnSend"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClick"
    android:layout_below="@id/etMessage"
    android:text="Send"/>
</LinearLayout>

```

=>Coding part of MainActivity.java

```

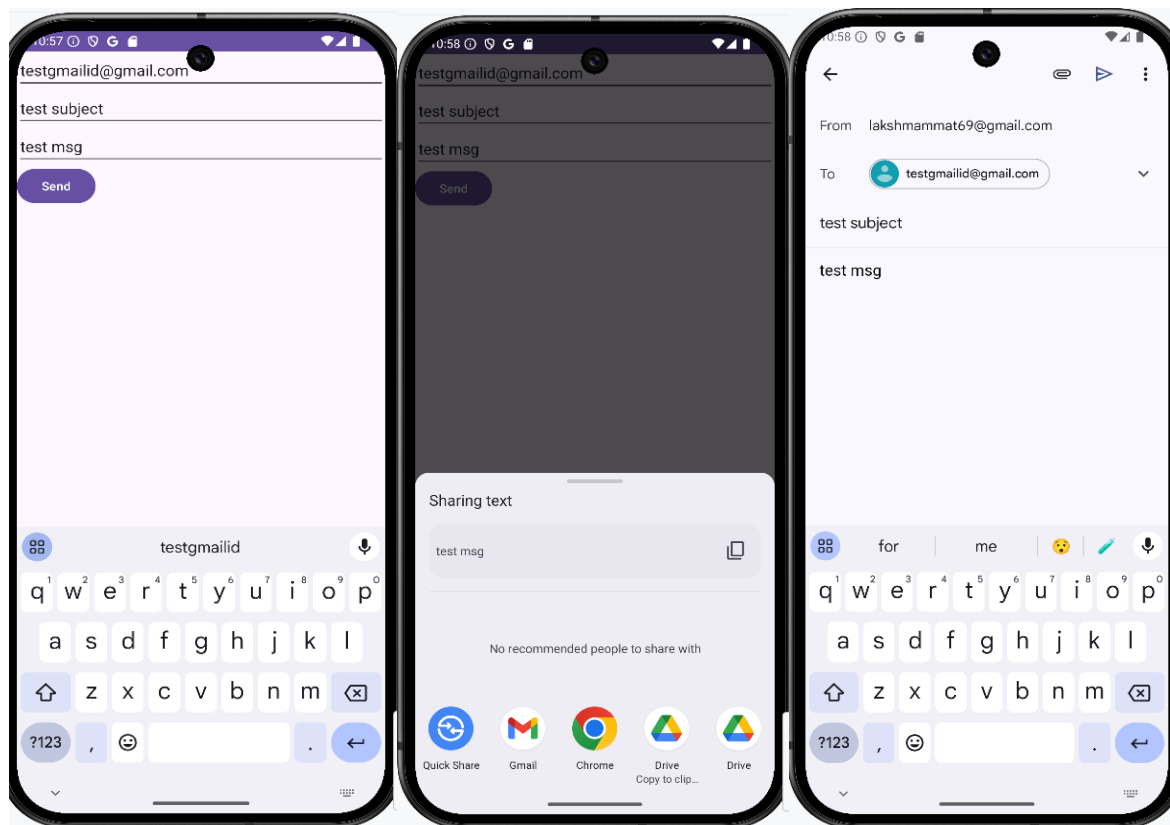
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    EditText etTo, etSubject, etMessage;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etTo = findViewById(R.id.etTo);
        etSubject = findViewById(R.id.etSubject);
        etMessage = findViewById(R.id.etMessage);
    }

    public void onClick(View v){
        String strTo = etTo.getText().toString().trim();
        String strSubject = etSubject.getText().toString().trim();
        String strMessage = etMessage.getText().toString().trim();
        Intent intent = new Intent(Intent.ACTION_SEND);
        intent.setType("text/plain");
        intent.putExtra(Intent.EXTRA_EMAIL, new String[]{strTo});
        intent.putExtra(Intent.EXTRA_SUBJECT, strSubject);
        intent.putExtra(Intent.EXTRA_TEXT, strMessage);
        if (intent.resolveActivity(getPackageManager()) != null)
        {
            startActivity(Intent.createChooser(intent, "Choose an email client"));
        }
    }
}

```

}
}
}



13. Create an Application with Login module. Check User name and password. On successful login change textview "Login Successful". On Login fail alert using Toast "Login Fail"

Steps:

- 1. Click New Project, the New Project Dialog box appears.**
- 2. Choose Empty Views Activity then click Next.**
- 3. Specify the Name of your project, Select the Language as Java, and Select the SDK as API 24("Nougat",Android 7.0).Click Finish Button.**
- 4. Update the following code in activity_main.xml and MainActivity.java**

=>Coding part of Activity_main.xml

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

```
<TextView
    android:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="200dp"
    android:text="Login Form"
    android:textSize="40sp" />
```

```
<TextView
    android:id="@+id/tvUserName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="User Name"
    android:textSize="24sp" />
```

```
<EditText
    android:id="@+id/etUserName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter the User Name"
    android:inputType="text"
    android:textSize="24sp" />
```

```
<TextView
    android:id="@+id/tvPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password"
    android:textSize="24sp" />
```

```
<EditText
    android:id="@+id/etPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter the Password"
    android:inputType="textPassword"
    android:textSize="24sp" />
```

```
<Button
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:onClick="onClick"
    android:text="Login" />
```

```
<TextView
    android:id="@+id/tvMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Welcome!!!"
    android:textColor="#E91E63"
    android:textSize="20sp" />
```

```
</LinearLayout>
```

=> Coding part of MainActivity.java

```
package com.example.loginapplication;
```

```
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
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 {
```

```
    EditText etUserName,etPassword;
    TextView tvMessage;
```

```
    @Override
```

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

```
        etUserName=(EditText)findViewById(R.id.etUserName);
        etPassword=(EditText)findViewById(R.id.etPassword);
        tvMessage=(TextView)findViewById(R.id.tvMessage);
    }
```

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

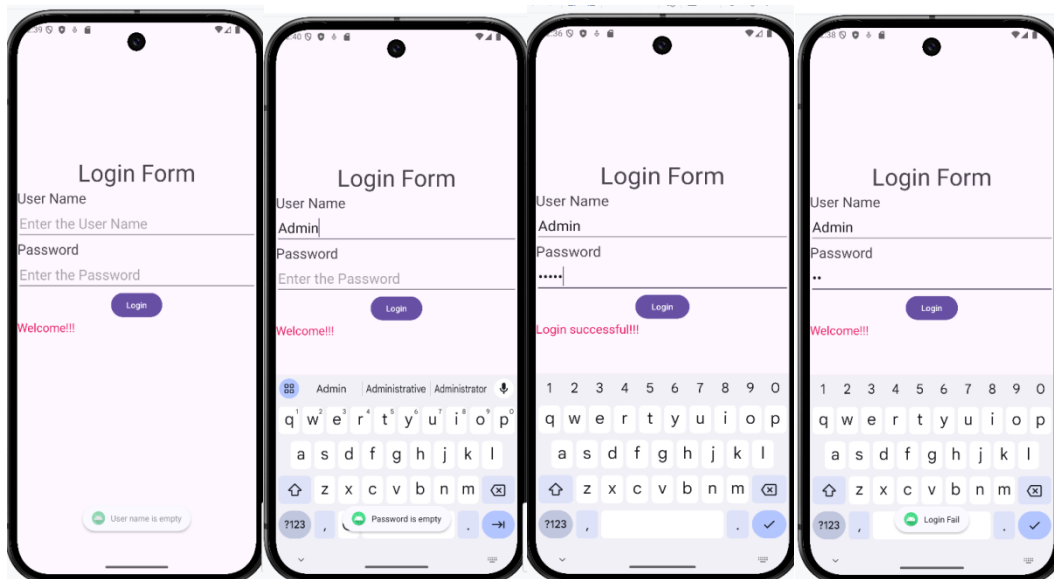
```
        tvMessage.setText("Welcome!!!");
        if(etUserName.getText().toString().isEmpty())
        {
            Toast.makeText(this,"User name is empty", Toast.LENGTH_LONG).show();
            return;
        }
        if(etPassword.getText().toString().isEmpty())
        {
            Toast.makeText(this,"Password is empty", Toast.LENGTH_LONG).show();
            return;
        }
        if(etUserName.getText().toString().equals("Admin") && etPassword.getText().toString().equals("Admin"))
        {
            tvMessage.setText("Login successful!!!");
        }
        else
        {
```

```

    Toast.makeText(this, "Login Fail", Toast.LENGTH_LONG).show();
}
}
}

```

=>OUTPUT OF THE PROGRAM



14. Learn to deploy Android applications

Steps to Deploy an Android Application

1. Prepare App (use Program 1 Hello world for this program)

2. Generate Signed APK (Android Package Kit):

- In Android Studio, navigate to Build > Generate Signed Bundle/APK.
- Follow the prompts to create a new keystore or use an existing one. A keystore is a binary file that contains a set of private keys.
- Configure the build type (release) and signing configuration.
- Generate the signed APK file.

3. Test your signed APK:

- Before distributing your app, test the signed APK to ensure that the signing process didn't introduce any issues.

- b. Install the APK on various devices and perform thorough testing.
- c. Release on Google Play Console:
- d. Sign in to the Google Play Console (<https://play.google.com/apps/publish>).
- e. Create a new app entry if this is your first release or select an existing app.
- f. Complete all the required information for the app listing, including the title, description, screenshots, and categorization
- g. Upload your signed APK file.
- h. Set pricing and distribution options.
- i. Optimize your store listing for search and conversion.
Once everything is set, click the "Publish" button to release your app to the Google Play Store.

4. Other Distribution Channels (Optional):

- Besides Google Play, you can distribute your app through other channels such as Amazon Appstore, Samsung Galaxy Store, or third party app marketplaces.
- Each distribution channel may have its own requirements and submission process, so be sure to follow their guidelines.

5. Monitor and Update:

- Keep an eye on user feedback and app performance metrics through the Google Play Console.
- Regularly update your app to fix bugs, add new features, and improve user experience based on feedback.