

## **CO1: Set up the Android development environment and debug apps using emulator and Logcat.**

### **Experiment 1: First Run App**

**Objective:** To successfully set up a basic android project and run it on either an android emulator or a physical android device

**Question:** create a new empty view activity Android project in android studio without making any code changes, successfully run this default empty app on an android emulator. take a screenshot of the app running on the emulator. if you have a physical Android phone, also try to run the app on your phone and take a screenshot

### **MainActivity.java**

```
package com.example.application1;
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;
        });
    }
}
```

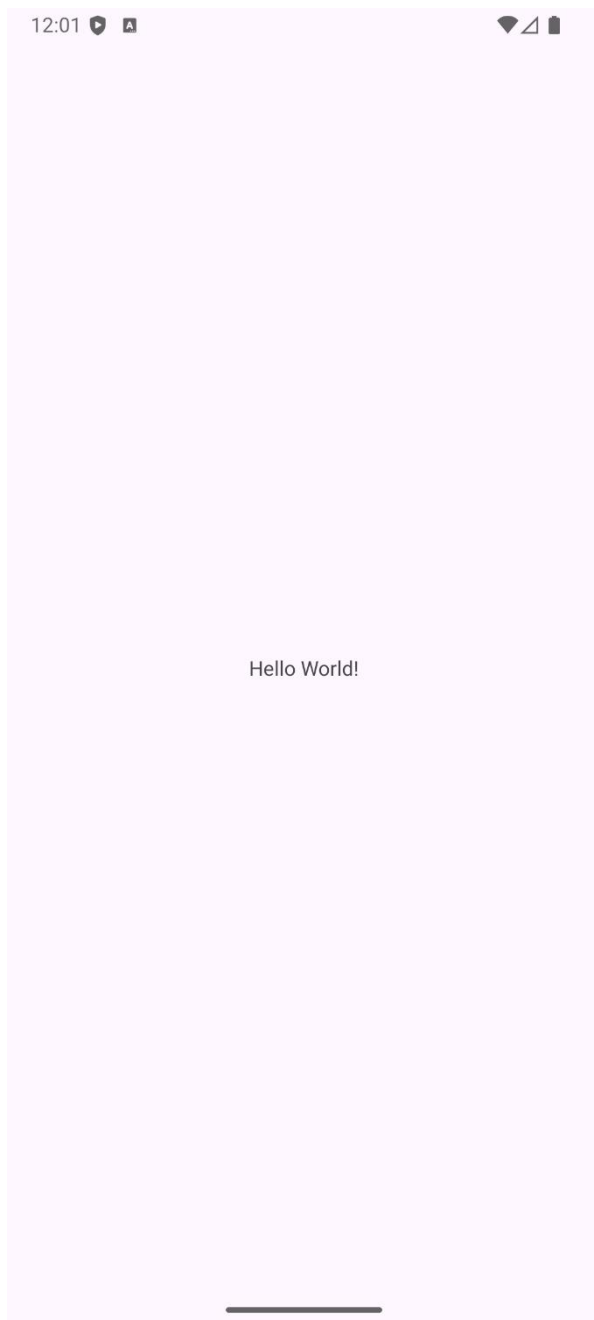
**activity\_main.xml**

```
<?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="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>
```

## Output



## **Experiment 2: "Logcat Debugging Basics"**

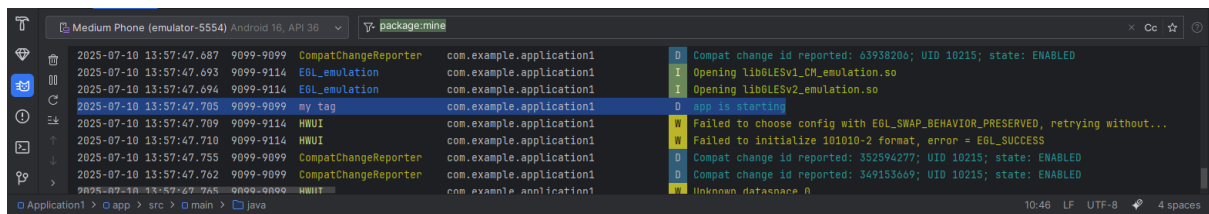
**Objective:** To learn how to use Logcat in Android Studio to view application messages and system logs for basic debugging.

**Question:** Open your "Hello, Interactive World!" app from CO1. In your MainActivity.java file, inside the onCreate () method, add a simple log message using Log.d("MyTag", "App is starting!"); Run the app. Then, open the Logcat window in Android Studio and find your custom log message. Demonstrate how to filter Logcat to show only messages from your app using the tag "MyTag"

### **MainActivity.java**

```
package com.example.application1;
import android.os.Bundle;
import android.util.Log;
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;
        });
        Log.d("my tag", "app is starting");
    }
}
```

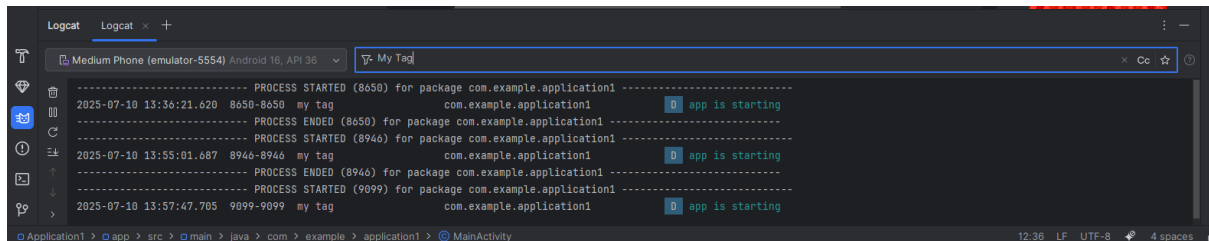
## Output



Medium Phone (emulator-5554) Android 16, API 36 package:mine

Time	Level	Tag	Package	Message
2025-07-10 13:57:47.687	DEBUG	CompatChangeReporter	com.example.application1	Compat change id reported: 63938206; UID 10215; state: ENABLED
2025-07-10 13:57:47.693	INFO	EGL_emulation	com.example.application1	Opening libGLESv1_CM.emulation.so
2025-07-10 13:57:47.694	INFO	EGL_emulation	com.example.application1	Opening libGLESv2_emulation.so
2025-07-10 13:57:47.705	DEBUG	my tag	com.example.application1	app is starting
2025-07-10 13:57:47.709	WARNING	HWUI	com.example.application1	Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
2025-07-10 13:57:47.710	WARNING	HWUI	com.example.application1	Failed to initialize 101010-2 format, error = EGL_SUCCESS
2025-07-10 13:57:47.755	DEBUG	CompatChangeReporter	com.example.application1	Compat change id reported: 352594277; UID 10215; state: ENABLED
2025-07-10 13:57:47.762	DEBUG	CompatChangeReporter	com.example.application1	Compat change id reported: 349153669; UID 10215; state: ENABLED
2025-07-10 13:57:47.765	UNKNOWN	HWUIT	com.example.application1	Unknown datasource 0

Application1 > app > src > main > java 10:46 LF UTF-8 4 spaces



Logcat Logcat + My Tag

Time	Level	Tag	Package	Message
2025-07-10 13:36:21.620	DEBUG	my tag	com.example.application1	app is starting
2025-07-10 13:36:21.620	DEBUG	my tag	com.example.application1	PROCESS STARTED (8650) for package com.example.application1
2025-07-10 13:36:21.620	DEBUG	my tag	com.example.application1	PROCESS ENDED (8650) for package com.example.application1
2025-07-10 13:55:01.687	DEBUG	my tag	com.example.application1	app is starting
2025-07-10 13:55:01.687	DEBUG	my tag	com.example.application1	PROCESS STARTED (8946) for package com.example.application1
2025-07-10 13:55:01.687	DEBUG	my tag	com.example.application1	PROCESS ENDED (8946) for package com.example.application1
2025-07-10 13:57:47.705	DEBUG	my tag	com.example.application1	app is starting
2025-07-10 13:57:47.705	DEBUG	my tag	com.example.application1	PROCESS STARTED (9099) for package com.example.application1

Application1 > app > src > main > java > com > example > application1 > MainActivity 12:36 LF UTF-8 4 spaces

**CO2: Understand Android components, app structure, and activity lifecycle.****Experiment 3: "Hello, Interactive World!"**

**Objective:** To learn how to make a button change text on the screen and show a quick message.

**Question:** Make a simple Android app. Design its main screen (activity\_main.xml) to include a TextView that initially displays "Hello World!" and a Button labelled "Change Greeting". Write the code (MainActivity.java) so that when the "Change Greeting" button is clicked, the TextView updates its text to "Welcome to Android!"

**MainActivity.java**

```
package com.example.application1;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private TextView Message ;
    private Button Click;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Message = findViewById(R.id.textViewGreeting);
        Click = findViewById(R.id.button);
        Click.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Message.setText("Welcome to Android");
            }
        });
    }
}
```

**activity\_main.xml**

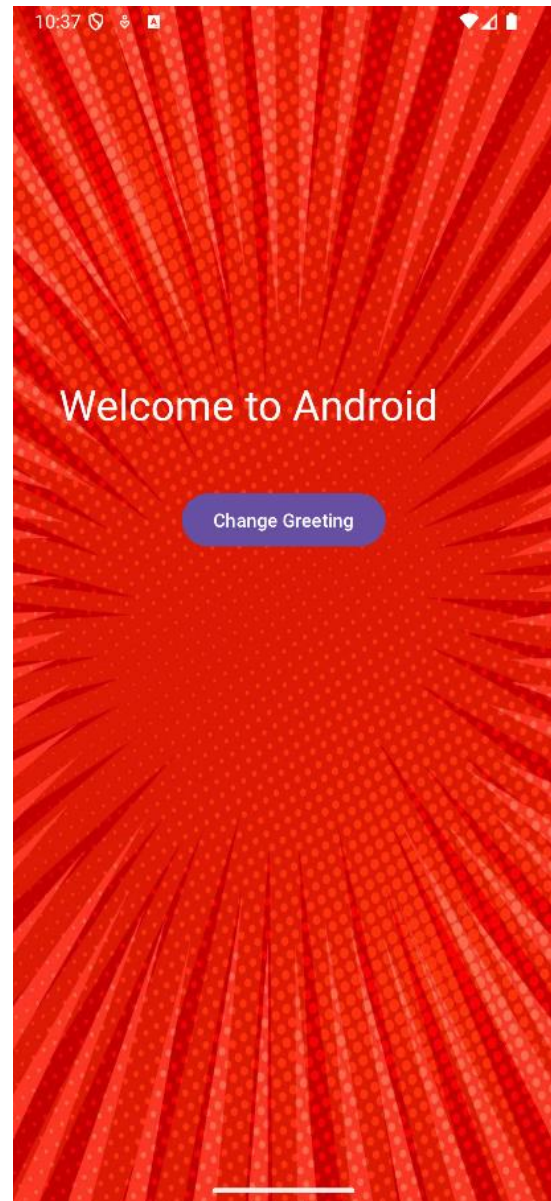
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/img2"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textViewGreeting"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="282dp"
        android:layout_marginEnd="88dp"
        android:text="Hello World!"
        android:textColor="@color/white"
        android:textSize="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="43dp"
        android:layout_marginEnd="40dp"
        android:text="Change Greeting"
        app:layout_constraintEnd_toEndOf="@+id/textViewGreeting"
        app:layout_constraintTop_toBottomOf="@+id/textViewGreeting" />

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

## Output





**Experiment 4: "Understanding Activity Lifecycle"**

**Objective:** To understand the various states an Android Activity goes through during its lifetime (its lifecycle) and to observe these transitions using Logcat.

**Question:** Write an Android program that demonstrates the Activity lifecycle

**MainActivity.java**

```
package com.example.application1;
import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private static final String TAG ="LifeCycle";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG,"OnCreate()-Activity is being created");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG,"OnStart()-Activity is visible");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG,"OnResume()-Activity is interactive");
    }

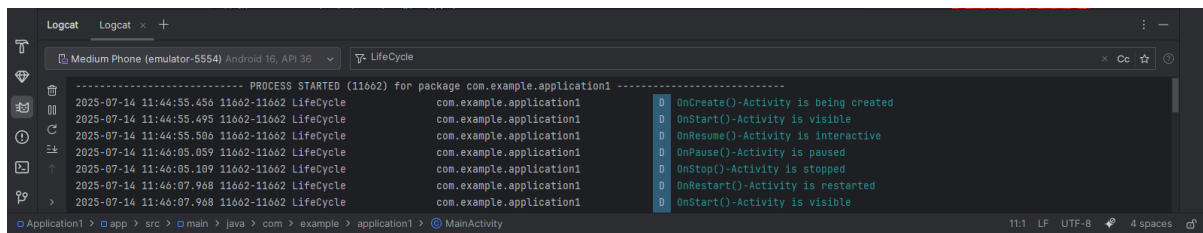
    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG,"OnPause()-Activity is paused");
    }
}
```

```
@Override
protected void onStop() {
    super.onStop();
    Log.d(TAG,"OnStop()-Activity is stopped");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.d(TAG,"OnDestroy()-Activity is destroyed");
}

@Override
protected void onRestart() {
    super.onRestart();
    Log.d(TAG,"OnRestart()-Activity is restarted");
}
}
```

## Output

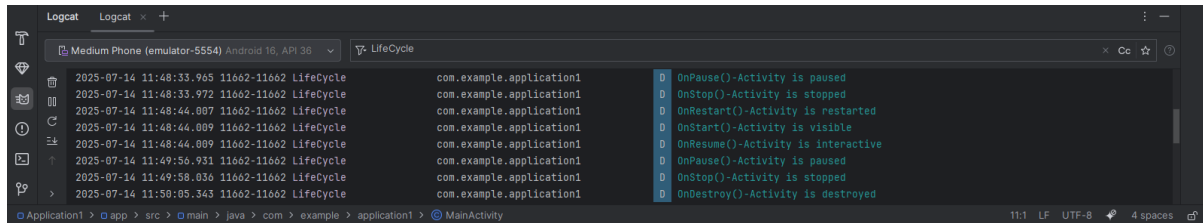


Logcat Logcat x +

Medium Phone (emulator-5554) Android 16, API 36 LifeCycle

```
----- PROCESS STARTED (11662) for package com.example.application1 -----
2025-07-14 11:44:55.456 11662-11662 LifeCycle com.example.application1 D onCreate()-Activity is being created
2025-07-14 11:44:55.495 11662-11662 LifeCycle com.example.application1 D onStart()-Activity is visible
2025-07-14 11:44:55.506 11662-11662 LifeCycle com.example.application1 D onResume()-Activity is interactive
2025-07-14 11:46:05.059 11662-11662 LifeCycle com.example.application1 D onPause()-Activity is paused
2025-07-14 11:46:05.109 11662-11662 LifeCycle com.example.application1 D onStop()-Activity is stopped
2025-07-14 11:46:07.968 11662-11662 LifeCycle com.example.application1 D onStart()-Activity is restarted
2025-07-14 11:46:07.968 11662-11662 LifeCycle com.example.application1 D onStart()-Activity is visible
```

Application1 > app > src > main > java > com > example > application1 > MainActivity 11:1 LF UTF-8 4 spaces



Logcat Logcat x +

Medium Phone (emulator-5554) Android 16, API 36 LifeCycle

```
2025-07-14 11:48:33.965 11662-11662 LifeCycle com.example.application1 D onPause()-Activity is paused
2025-07-14 11:48:33.972 11662-11662 LifeCycle com.example.application1 D onStop()-Activity is stopped
2025-07-14 11:48:44.007 11662-11662 LifeCycle com.example.application1 D onStart()-Activity is restarted
2025-07-14 11:48:44.009 11662-11662 LifeCycle com.example.application1 D onStart()-Activity is visible
2025-07-14 11:48:44.009 11662-11662 LifeCycle com.example.application1 D onResume()-Activity is interactive
2025-07-14 11:49:56.931 11662-11662 LifeCycle com.example.application1 D onPause()-Activity is paused
2025-07-14 11:49:58.036 11662-11662 LifeCycle com.example.application1 D onStop()-Activity is stopped
2025-07-14 11:58:05.343 11662-11662 LifeCycle com.example.application1 D onDestroy()-Activity is destroyed
```

Application1 > app > src > main > java > com > example > application1 > MainActivity 11:1 LF UTF-8 4 spaces

## **CO3: Design basic and advanced user interfaces using layouts, widgets, dialogs, and fragments**

### **Experiment 5: "TextView vs. Toast: Message Display"**

**Objective:** To clearly demonstrate and differentiate between the characteristics and appropriate use cases of a TextView (for persistent UI display) and a Toast message (for transient, non-intrusive feedback) in an Android application.

**Question:** Design an Android application to demonstrate the differences between TextView and Toast messages in terms of how they display information.

#### **MainActivity.java**

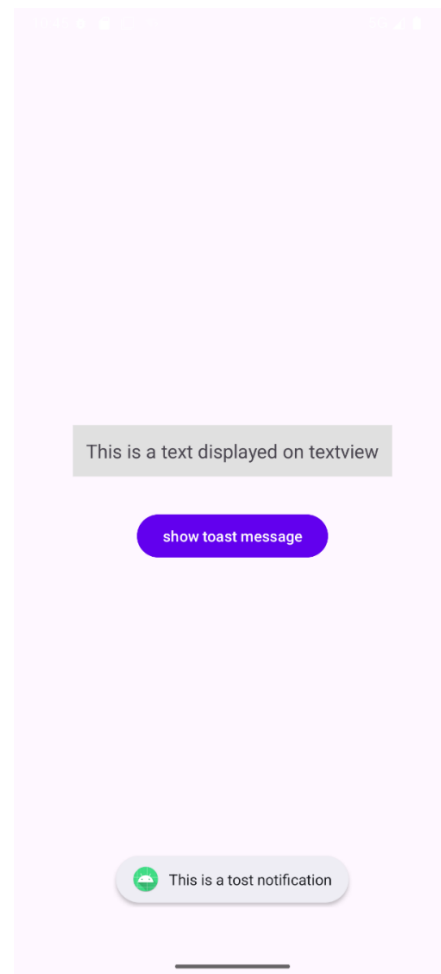
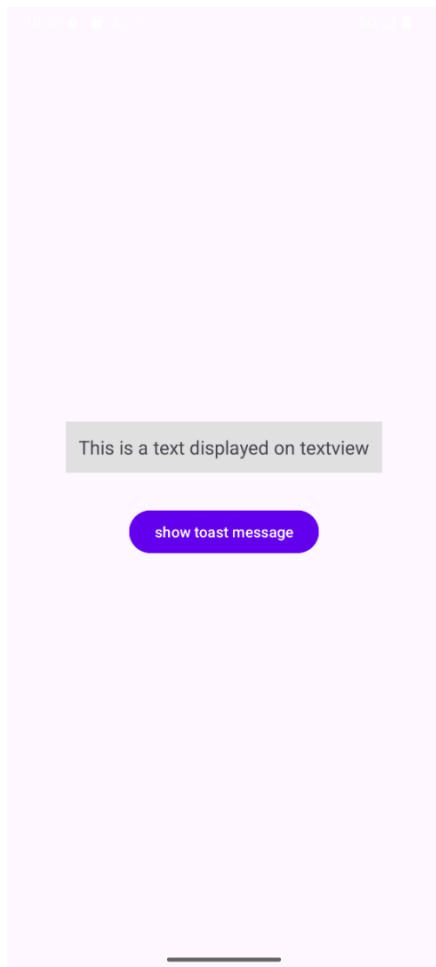
```
package com.example.experiment5;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView textView = findViewById(R.id.textview);
        textView.setText("This is a text displayed on textview");
    }
    // Button click handler
    public void showToastMessage(View view) {
        Toast.makeText(this, "This is a tost notification",
            Toast.LENGTH_LONG).show();
    }
}
```

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

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

```
        android:padding="16dp"
        android:gravity="center">
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:padding="12dp"
    android:background="#E0E0E0"
    android:layout_marginBottom="32dp"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="show toast message"
    android:onClick="showToastMessage"
    android:backgroundTint="#6200EE"
    android:textColor="#FFFFFF"/>
</LinearLayout>
```

## Output



### **Experiment 6: "Basic Login and Input Display"**

**Objective:** To learn how to use EditText for user input, arrange UI elements using LinearLayout, and retrieve input data to display in a Toast message.

**Question:** Design a basic Login Form with places for username and password. Use a Linear Layout to arrange them, and show a message using a Toast.

#### **MainActivity.java**

```
package com.example.experiment6;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText username;
    EditText password;
    Button loginButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username=findViewById(R.id.username);
        password=findViewById(R.id.password);
        loginButton=findViewById(R.id.loginButton);
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(username.getText().toString().equals("user")&&
password.getText().toString().equals("1234"))
                {
                    Toast.makeText(MainActivity.this,"Login
Success",Toast.LENGTH_SHORT).show();
                }
                else {
                    Toast.makeText(MainActivity.this,"Login
Failed",Toast.LENGTH_SHORT).show();
                }
            }
        })
    }
}
```

```

    });
}
}

```

### **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"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/loginbkg"
    tools:context=".MainActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp"
        android:background="@drawable/custom_edittext">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"
            android:padding="24dp">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Login"
                android:id="@+id/logintext"
                android:textSize="36sp"

```



```
android:textAlignment="center"  
android:textStyle="bold"  
android:textColor="@color/purple"/>
```

<EditText

```
android:layout_width="match_parent"  
android:layout_height="50dp"  
android:id="@+id/username"  
android:background="@drawable/custom_edittext"  
android:drawableLeft="@drawable/baseline_person_24"  
android:drawablePadding="8dp"  
android:hint="Username"  
android:padding="8dp"  
android:textColor="@color/black"  
android:textColorHighlight="@color/cardview_dark_background"  
android:layout_marginTop="40dp"/>
```

<EditText

```
android:layout_width="match_parent"  
android:layout_height="50dp"  
android:id="@+id/password"  
android:background="@drawable/custom_edittext"  
android:drawableLeft="@drawable/baseline_person_24"  
android:drawablePadding="8dp"  
android:hint="Password"  
android:padding="8dp"  
android:inputType="textPassword"  
android:textColor="@color/black"  
android:textColorHighlight="@color/cardview_dark_background"  
android:layout_marginTop="20dp"/>
```

<Button

```
android:layout_width="match_parent"  
android:layout_height="60dp"  
android:id="@+id/loginButton"  
android:text="Login"  
android:textSize="18sp"  
android:layout_marginTop="30dp"  
android:backgroundTint="@color/purple"  
app:cornerRadius="20dp"/>
```

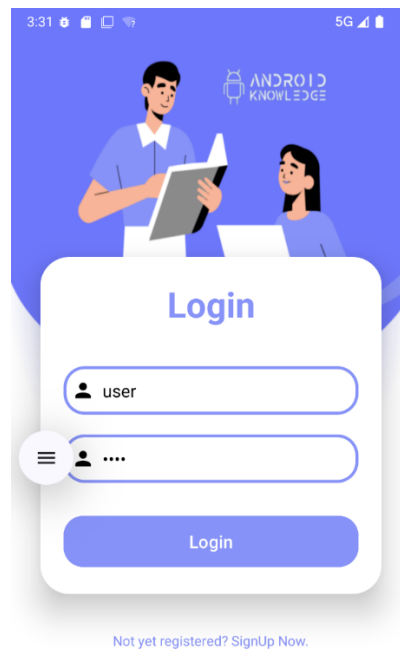
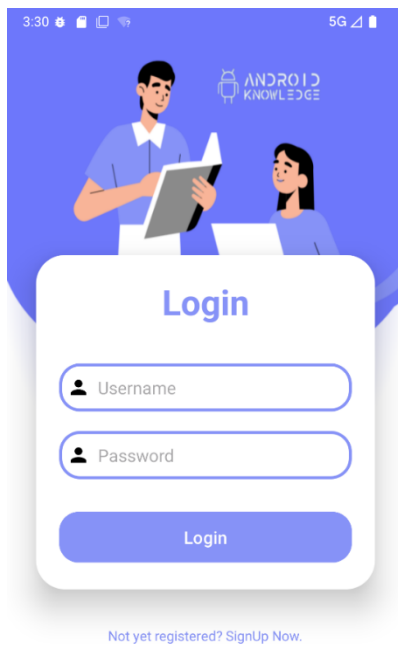
```
</LinearLayout>
```

```
</androidx.cardview.widget.CardView>
```

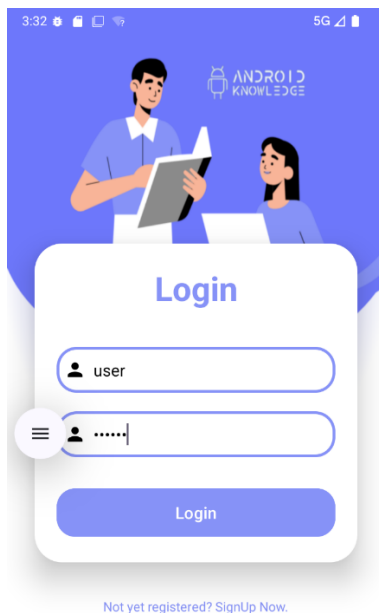
```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:padding="8dp"  
    android:text="Not yet registered? SignUp Now."  
    android:textSize="14sp"  
    android:textAlignment="center"  
    android:id="@+id/signupText"  
    android:textColor="@color/purple"  
    android:layout_marginBottom="20dp"/>
```

```
</LinearLayout>
```

## Output



Login Success



Login Failed

**Experiment 7: " Basic Number Addition "**

**Objective:** To learn how to get two numerical inputs from EditText fields and display their sum in a TextView on the screen.

**Question:** Design an Android screen where a user can type two numbers into two separate input boxes (EditText). Add a Button labeled "Add Numbers" and a TextView below it. When the "Add Numbers" button is clicked, add the two numbers typed by the user and display the total sum in the TextView on the screen.

**MainActivity.java**

```
package com.example.experiment7;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
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 {

    EditText editTextNumber1, editTextNumber2;
    TextView textViewResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTextNumber1 = findViewById(R.id.editTextNumber1);
        editTextNumber2 = findViewById(R.id.editTextNumber2);
        Button buttonAdd = findViewById(R.id.buttonAdd);
        textViewResult = findViewById(R.id.textViewResult);
        buttonAdd.setOnClickListener(view -> {
            String num1Str = editTextNumber1.getText().toString().trim();
            String num2Str = editTextNumber2.getText().toString().trim();

            if (!num1Str.isEmpty() && !num2Str.isEmpty()) {
                try {
                    int num1 = Integer.parseInt(num1Str);
```

```

        int num2 = Integer.parseInt(num2Str);
        int sum = num1 + num2;
        textViewResult.setText("Sum: " + sum);
    } catch (NumberFormatException e) {
        textViewResult.setText("Invalid number format");
    }
} else {
    textViewResult.setText("Please enter both numbers");
}
});
}
}

```

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

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="24dp">

    <EditText
        android:id="@+id/editTextNumber1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter first number"
        android:inputType="number"
        android:padding="12dp" />

    <EditText
        android:id="@+id/editTextNumber2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter second number"
        android:inputType="number"
        android:padding="12dp"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/buttonAdd"

```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Add Numbers" />
```

```
<TextView  
    android:id="@+id/textViewResult"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:textSize="18sp"  
    android:padding="12dp"  
    android:layout_marginTop="16dp" />
```

```
</LinearLayout>
```

## Output



Enter first number

Enter second number

Add Numbers



15

17

Add Numbers

Sum: 32

### **Experiment 8: " Image Toggle "**

**Objective:** To learn how to use FrameLayout to stack UI elements and toggle their visibility.

**Question:** Develop an application that toggles an image using FrameLayout. When a button is clicked, make one image disappear and another appear in its place.

#### **MainActivity.java**

```
package com.example.experiment8;
import android.media.Image;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
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 implements
View.OnClickListener {
    ImageView Img;
    boolean flg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        Button btn=(Button)findViewById(R.id.btn);
        Img=(ImageView)findViewById(R.id.img);
        btn.setOnClickListener((View.OnClickListener) this);
        flg=true;
    }

    @Override
    public void onClick(View view) {
        if(flg){
            Img.setImageResource(R.drawable.image1);
            flg=false;
        }
    }
}
```



```
        else{
            Img.setImageResource(R.drawable.image2);
        }
    }
}
```

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

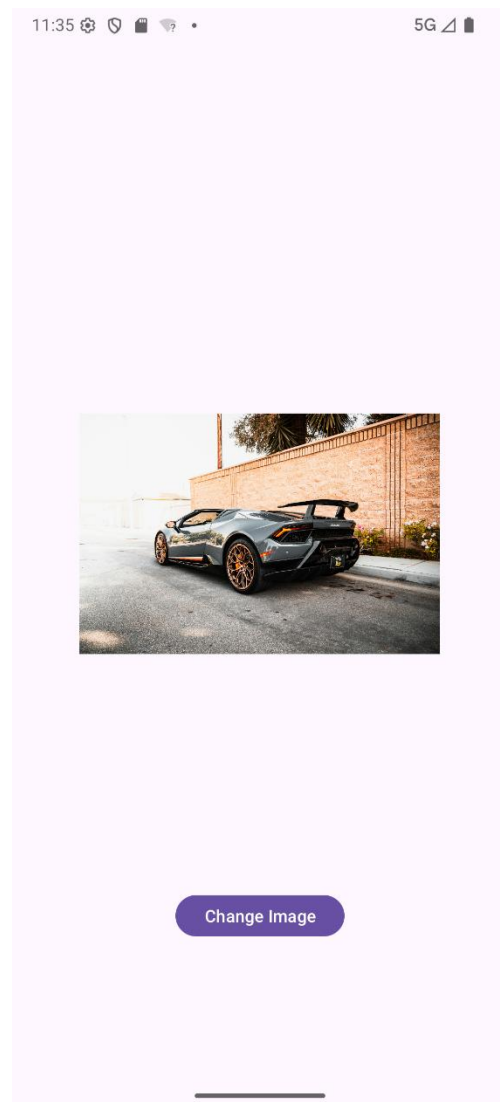
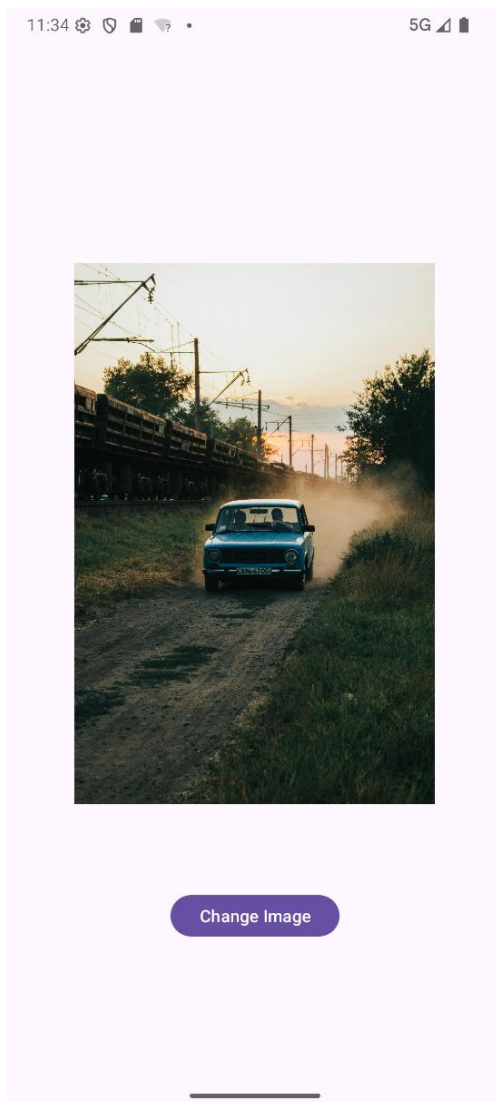
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <ImageView
        android:id="@+id/img"
        android:layout_width="299dp"
        android:layout_height="592dp"
        android:src="@drawable/image1"
        android:text="Hello World!" />

    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="42dp"
        android:text="Change Image" />

</LinearLayout>
```

## Output



### **Experiment 9: " Full-Featured Calculator with Layout Design "**

**Objective:** To learn how to build a complete, working calculator combining arithmetic logic with advanced layout techniques using GridLayout and LinearLayout.

**Question:** Build a complete calculator app. Design its entire screen using GridLayout for all the number buttons (0-9) and operation buttons (+, -, \*, /, equals, clear). Use a LinearLayout for the display area at the top where numbers are typed and results are shown. Make sure the calculator can perform all basic math operations correctly, and displays the answers in the top display area.

#### **MainActivity.java**

```
package com.example.experiment9;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
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 {
    EditText editText;
    String oldNumber = "";
    String operator = "";
    boolean isNewOp = true;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        editText = findViewById(R.id.editText);
        editText.setShowSoftInputOnFocus(false); // disables keyboard popup
    }
    // Handles all button clicks
    public void operatorEvent(View view) {
        Button btn = (Button) view;
        String btnText = btn.getText().toString();
        String number = editText.getText().toString();
```

```
switch (btnText) {
    case "C":
        editText.setText("0.0");
        isNewOp = true;
        oldNumber = "";
        operator = "";
        break;
    case "+":
    case "-":
    case "*":
    case "/":
        oldNumber = number;
        operator = btnText;
        editText.setText(number + operator);
        isNewOp = false;
        break;
    case "=":
        String[] parts = number.split("[ + \\+ \\- \\* / + ]");
        if (parts.length < 2) return;
        double first = Double.parseDouble(parts[0]);
        double second = Double.parseDouble(parts[1]);
        double result = 0.0;
        switch (operator) {
            case "+": result = first + second; break;
            case "-": result = first - second; break;
            case "*": result = first * second; break;
            case "/":
                if (second != 0) result = first / second;
                else {
                    editText.setText("Error");
                    return;
                }
            break;
        }
        editText.setText(String.valueOf(result));
        isNewOp = true;
        break;
    default: // digits
        if (isNewOp || number.equals("0.0")) {
            editText.setText(btnText);
```

```

        isNewOp = false;
    } else {
        editText.setText(number + btnText);
    }
    break;
}
}
}

```

### **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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="421dp"
        android:id="@+id/editText"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="20dp"
        android:layout_weight="1"
        android:textColor="@color/orange"
        android:ems="10"
        android:layout_gravity="right|center"
        android:text="0.0"
        android:textSize="100dp"
        android:inputType="number"/>
    <GridLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:rowCount="4"
        android:columnCount="4"

```

```
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_marginBottom="10dp">
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b1"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
    android:text="C"
    android:textSize="40dp"
    tools:ignore="OnClick" />
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b2"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
    android:text="0"
    android:textSize="40dp"
    tools:ignore="OnClick" />
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b3"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
    android:text="*"
    android:textSize="40dp"
    tools:ignore="OnClick" />
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b4"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
```

```
        android:text="-"
        android:textSize="40dp"
        tools:ignore="OnClick" />

</GridLayout>
<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:rowCount="4"
    android:columnCount="4"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp">
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b5"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="7"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b6"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="8"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b7"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
```

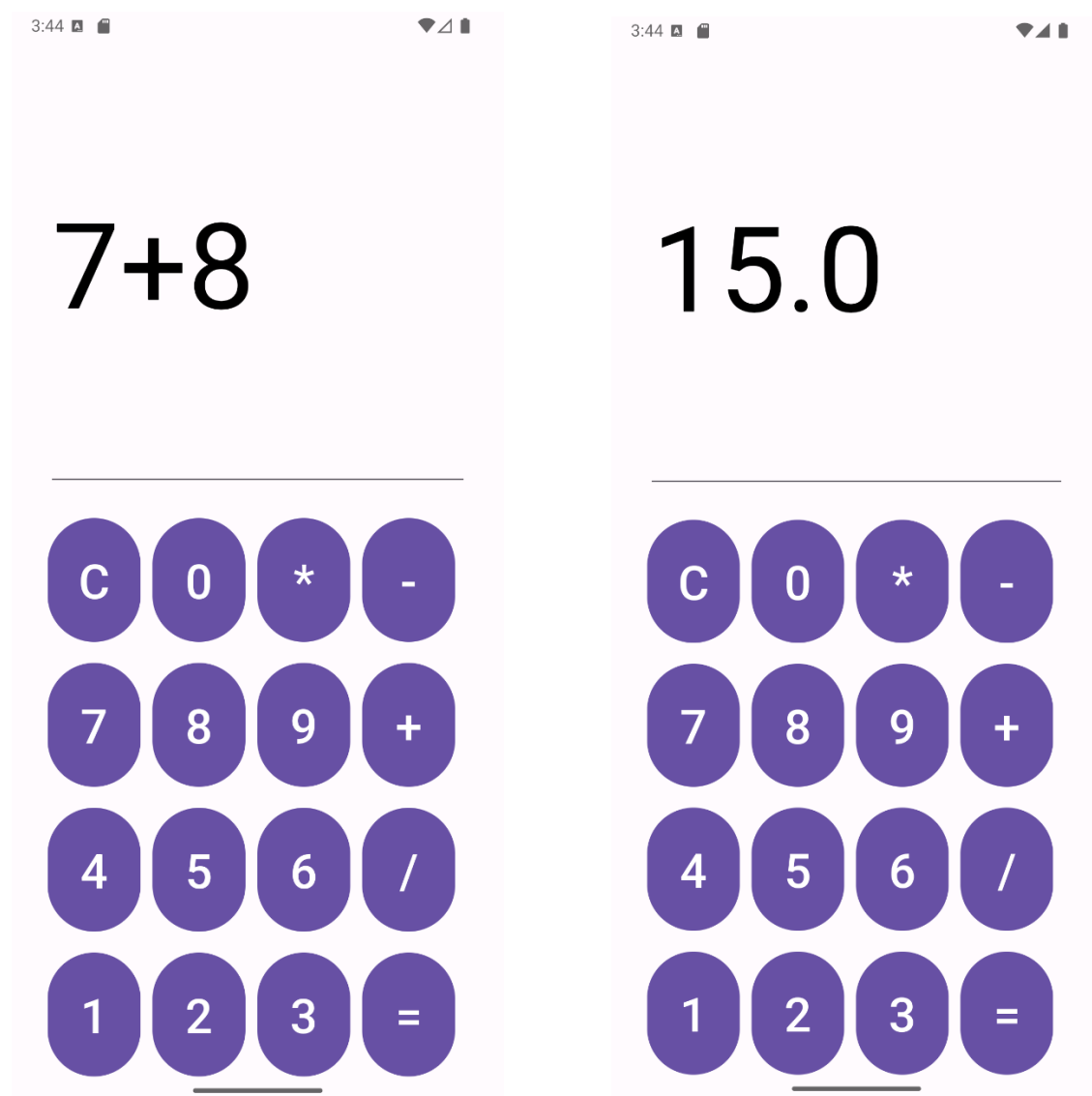
```
        android:text="9"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b8"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="+"
        android:textSize="40dp"
        tools:ignore="OnClick" />
</GridLayout>
<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:rowCount="4"
    android:columnCount="4"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp">
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b9"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="4"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b10"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="5"
```



```
        android:textSize="40dp"
        tools:ignore="OnClick" />
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b11"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
    android:text="6"
    android:textSize="40dp"
    tools:ignore="OnClick" />
<Button
    android:layout_width="35pt"
    android:layout_height="50pt"
    android:id="@+id/b12"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:onClick="operatorEvent"
    android:text="/"
    android:textSize="40dp"
    tools:ignore="OnClick" />
</GridLayout>
<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:rowCount="4"
    android:columnCount="4"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp">
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b13"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="1"
        android:textSize="40dp"
```

```
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b14"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="2"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b15"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text="3"
        android:textSize="40dp"
        tools:ignore="OnClick" />
    <Button
        android:layout_width="35pt"
        android:layout_height="50pt"
        android:id="@+id/b16"
        android:layout_marginRight="10dp"
        android:layout_weight="1"
        android:onClick="operatorEvent"
        android:text=""
        android:textSize="40dp"
        tools:ignore="OnClick" />

</GridLayout>
</LinearLayout>
```

Output

## **Experiment 10: " Student Registration Form with Checks "**

**Objective:** Learn to build a form with different input fields and check if the information typed is correct.

**Question:** Design a student sign-up form. Include boxes for Name, Email, Password, and Phone Number. Add a "Register" button. When clicked, check if each box has corrected info (e.g., email has "@", phone is 10 digits, password isn't empty). If everything is good, show a message "Signed Up!". If something is wrong, show a message saying what needs fixing.

### **MainActivity.java**

```
package com.example.experiment10;
import android.os.Bundle;
import android.view.View;
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;
import com.google.android.material.button.MaterialButton;
public class MainActivity extends AppCompatActivity {

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

        // 1. Initialize all EditText and Button views
        EditText username = findViewById(R.id.username);
        EditText useremail = findViewById(R.id.useremail);
        EditText phoneNumber = findViewById(R.id.phonenumber);
        EditText password = findViewById(R.id.password);
        // Based on your XML, cpassword is the fourth EditText, so we'll assume it's for
        the phone number.
        MaterialButton regbtn = findViewById(R.id.signupbtn);

        // 2. Set an OnClickListener for the registration button
```

```

regbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // 3. Get the text from each input field and trim whitespace
        String name = username.getText().toString().trim();
        String email = useremail.getText().toString().trim();
        String phone = phoneNumber.getText().toString().trim();
        String pass = password.getText().toString().trim();

        // 4. Perform validation checks
        if (name.isEmpty()) {
            Toast.makeText(MainActivity.this, "Name cannot be empty.",
Toast.LENGTH_SHORT).show();
        } else if (!email.contains("@")) {
            Toast.makeText(MainActivity.this, "Please enter a valid email address.",
Toast.LENGTH_SHORT).show();
        } else if (phone.length() != 10) {
            Toast.makeText(MainActivity.this, "Phone number must be 10 digits.",
Toast.LENGTH_SHORT).show();
        } else if (pass.isEmpty()) {
            Toast.makeText(MainActivity.this, "Password cannot be empty.",
Toast.LENGTH_SHORT).show();
        } else {
            // 5. If all validations pass, show a success message
            Toast.makeText(MainActivity.this, "Signed Up! Welcome, " + name +
".", Toast.LENGTH_LONG).show();
        }
    }
});
}
}

```

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

```

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

```

```
android:layout_height="match_parent"
android:background="@drawable/rbgg"
tools:context=".MainActivity">
```

```
<TextView
```

```
    android:id="@+id/signuptitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginBottom="25dp"
    android:gravity="center"
    android:text="Sign-Up"
    android:textColor="@color/black"
    android:textSize="35dp"
    android:textStyle="bold" />
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/username"
    android:layout_below="@+id/signuptitle"
    android:background="#30ffffff"
    android:hint="Username"
    android:textColorHint="@color/white"
    android:textColor="@color/white"
    android:layout_margin="10dp"
    android:padding="20dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/useremail"
    android:layout_below="@+id/username"
    android:background="#30ffffff"
    android:hint="Email"
    android:textColorHint="@color/white"
    android:textColor="@color/white"
    android:layout_margin="10dp"
    android:padding="20dp"/>
```

```
<EditText
    android:id="@+id/phonenummer"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/useremail"
    android:layout_margin="10dp"
    android:layout_marginTop="129dp"
    android:layout_marginBottom="10dp"
    android:background="#30ffffff"
    android:hint="Phone Number"
    android:padding="20dp"
    android:textColor="@color/white"
    android:textColorHint="@color/white" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/password"
    android:layout_below="@+id/phonenummer"
    android:background="#30ffffff"
    android:hint="Password"
    android:textColorHint="@color/white"
    android:textColor="@color/white"
    android:layout_margin="10dp"
    android:padding="20dp"/>
```

```
<com.google.android.material.button.MaterialButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/signupbtn"
    android:text="Sign-Up"
    android:textSize="25dp"
    android:layout_below="@+id/password"
    android:layout_centerHorizontal="true"
    android:backgroundTint="@color/design_default_color_secondary"
    android:layout_margin="20dp"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/info"
    android:text="or signup with"
    android:textSize="25dp"
    android:textColor="@color/black"
    android:layout_below="@+id/signupbtn"
    android:layout_centerHorizontal="true"/>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/socialicons"
    android:gravity="center"
    android:layout_below="@+id/info"
    android:layout_alignParentBottom="true">
```

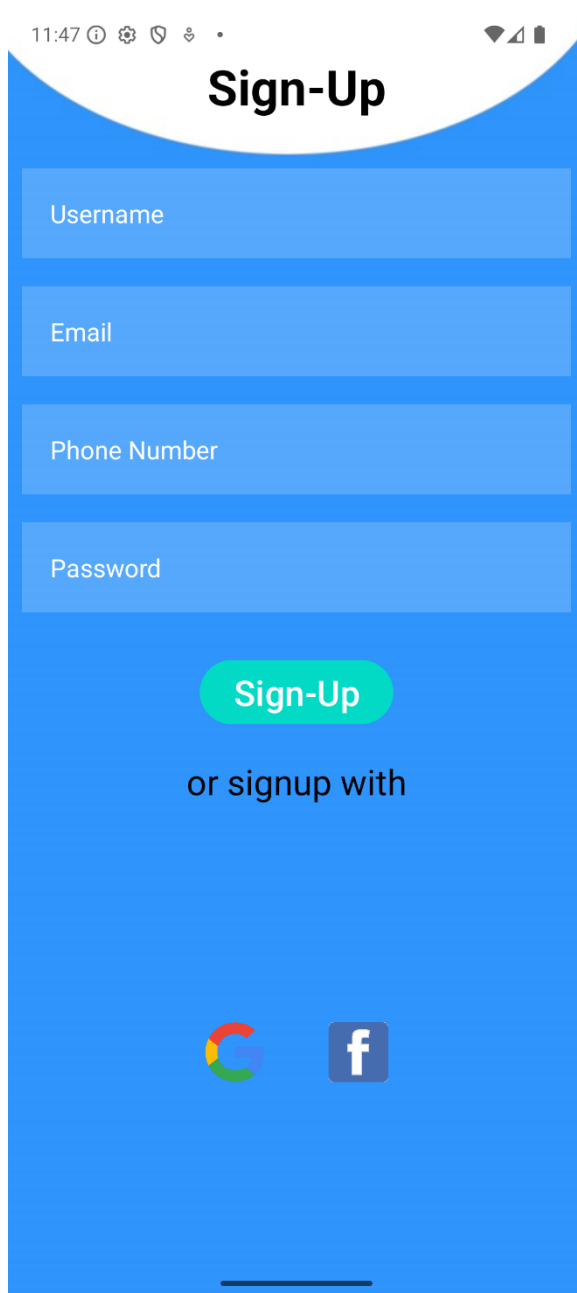
```
<ImageView
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_margin="20dp"
    android:src="@drawable/googleicon"/>
```

```
<ImageView
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_margin="20dp"
    android:src="@drawable/facebookicon"/>
```

```
</LinearLayout>
```

```
</RelativeLayout>
```



**Output**

11:47 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Username



Email

Phone Number

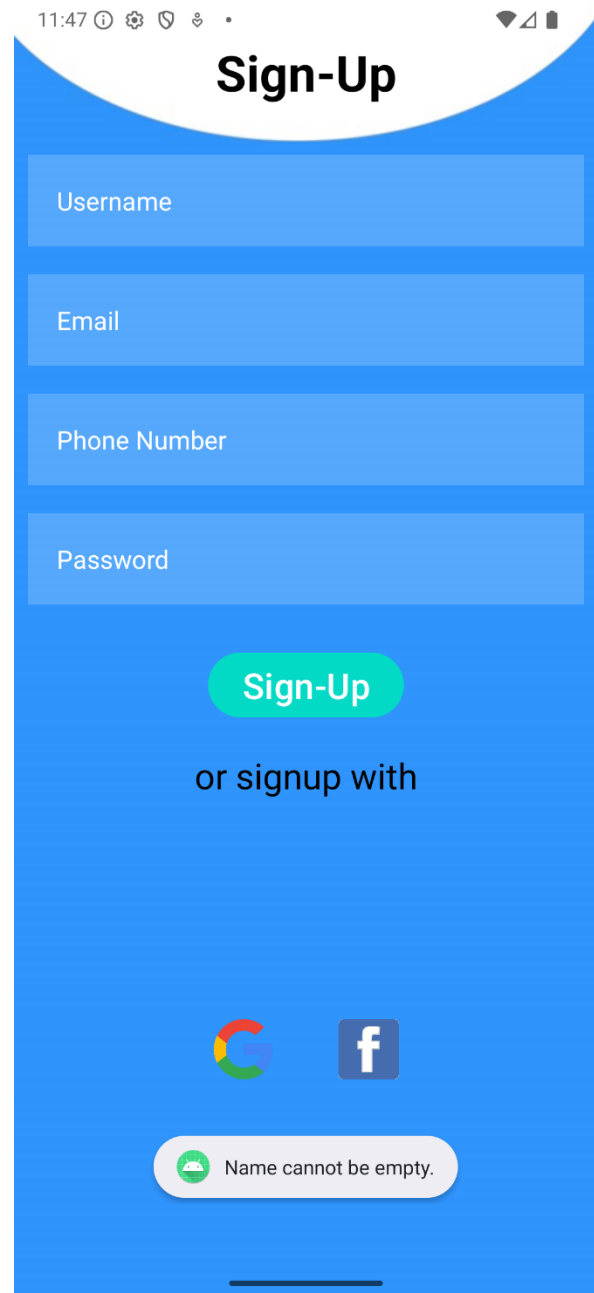
Password

Sign-Up

or signup with

Horizontal line at the bottom



11:47 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Username



Email


Phone Number

Password

Sign-Up

or signup with

 Name cannot be empty.

Horizontal line at the bottom

11:47 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Gokul

gokul@gmail.com



Phone Number


Password

☰

Sign-Up

or signup with

 Please enter a valid email address.

11:52 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Gokul

gokul@gmail.com



1234


Password

☰

Sign-Up

or signup with

 Phone number must be 10 digits.

11:53 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Gokul

gokul@gmail.com



9061393951


Password

☰

Sign-Up

or signup with

 Password cannot be empty.

11:53 ⓘ ⚙️ 🔒 🔔 •

## Sign-Up

Gokul

gokul@gmail.com



9061393951


gokul123

☰

Sign-Up

or signup with

 Signed Up! Welcome, Gokul.

## **Experiment 11: "Interest Selection Profile Page"**

**Objective:** Learn to use CheckBox for multiple selections and ListView to display selected items dynamically in a profile settings layout.

**Question:** Design a profile settings page for a user. Add several CheckBoxes with options like "Music", "Sports", "Travel", "Movies", etc., representing the user's interests. Below the CheckBoxes, add a "Submit" button. When the button is clicked, display the selected interests in a TextView below.

### **MainActivity.java**

```
package com.example.experiment11;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;
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 {
    CheckBox cbMusic, cbSports, cbTravel, cbMovies, cbReading;
    Button btnSubmit;
    TextView tvResult;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        cbMusic = findViewById(R.id.cbMusic);
        cbSports = findViewById(R.id.cbSports);
        cbTravel = findViewById(R.id.cbTravel);
        cbMovies = findViewById(R.id.cbMovies);
        cbReading = findViewById(R.id.cbReading);
        btnSubmit = findViewById(R.id.btnSubmit);
        tvResult = findViewById(R.id.tvResult);
        btnSubmit.setOnClickListener(v -> {
            StringBuilder selected = new StringBuilder("Your Interests: "); if
(cbMusic.isChecked()) selected.append("Music, ");
            if (cbSports.isChecked()) selected.append("Sports, ");
            if (cbTravel.isChecked()) selected.append("Travel, ");
```

```

        if (cbMovies.isChecked()) selected.append("Movies, ");
        if (cbReading.isChecked()) selected.append("Reading, ");
        if (selected.toString().equals("Your Interests: ")) {
            tvResult.setText("No interests selected.");
        } else {
            selected.setLength(selected.length() - 2);
            tvResult.setText(selected.toString());
        }
    });
}
}

```

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

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="24dp">
        <TextView
            android:id="@+id/tvTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Select Your Interests"
            android:textSize="20sp"
            android:textStyle="bold"
            android:gravity="center"
            android:layout_marginBottom="16dp" />
        <CheckBox
            android:id="@+id/cbMusic"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Music" />
        <CheckBox
            android:id="@+id/cbSports"
            android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:text="Sports" />
<CheckBox
    android:id="@+id/cbTravel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Travel" />
<CheckBox
    android:id="@+id/cbMovies"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Movies" />
<CheckBox
    android:id="@+id/cbReading"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reading" />
<Button
    android:id="@+id/btnSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"/>

<TextView
    android:id="@+id/tvResult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:layout_marginTop="24dp"
    android:textStyle="italic"
    android:text="Your selected interests will appear here." /> </LinearLayout>
</ScrollView>
```

## Output

### Select Your Interests

☐ Music

☐ Sports

☐ Travel

☐ Movies

☐ Reading

Submit

Your selected interests will appear here.

### Select Your Interests

☒ Music

☒ Sports

☒ Travel

☒ Movies

☐ Reading

Submit

Your Interests: Music, Sports, Travel, Movies

## **Experiment 12: "Image Slider with Button Clicks"**

**Objective :** Learn to switch between multiple images (more than two) using `FrameLayout` and control their visibility using `Next/Previous` buttons.

**Question:** Design an Android app that displays one image at a time using a `FrameLayout`. Place at least inside the layout, one on top of the other. Add two buttons: `next` and `prev`. When the user clicks `next`, show the next image; clicking `prev` shows the previous image. The images should loop correctly from last to first and vice versa.

### **MainActivity.java**

```
package com.example.experiment12;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
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 {
    private ImageView[] images;
    private int current = 0; // Index of currently visible image
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        images = new ImageView[] {
            findViewById(R.id.image1),
            findViewById(R.id.image2),
            findViewById(R.id.image3)
        };
        // Add more if you have more images

        Button nextBtn = findViewById(R.id.btnNext);
        Button prevBtn = findViewById(R.id.btnPrev);
        nextBtn.setOnClickListener(new View.OnClickListener() {
            @Override
```



```

        public void onClick(View v) {
            images[current].setVisibility(View.GONE);
            current = (current + 1) % images.length;
            images[current].setVisibility(View.VISIBLE);
        }
    });
    prevBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            images[current].setVisibility(View.GONE);
            current = (current - 1 + images.length) % images.length;
            images[current].setVisibility(View.VISIBLE);
        }
    });
}
}

```

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

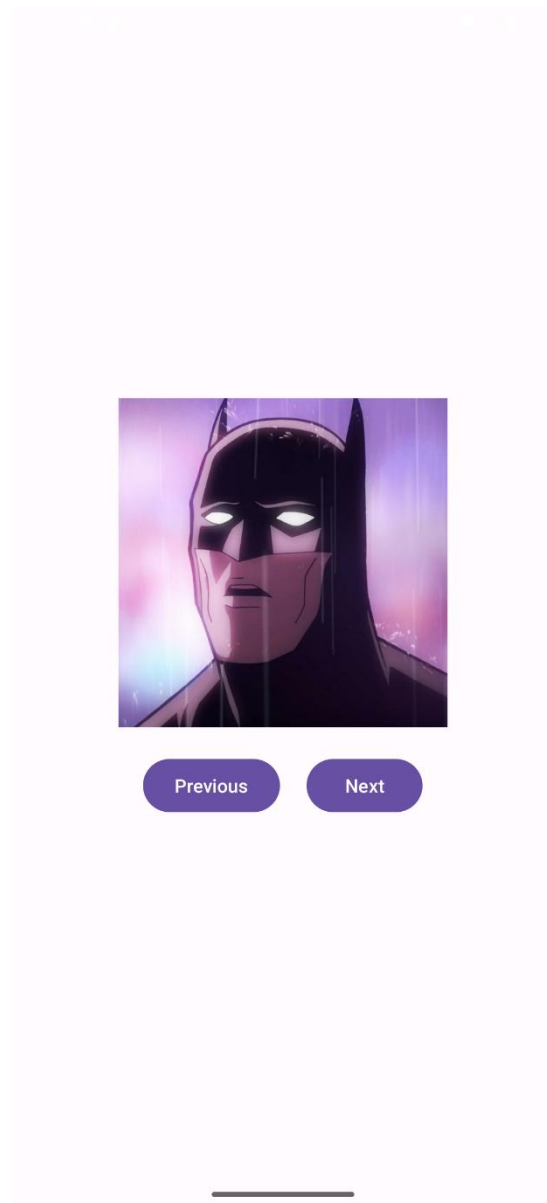
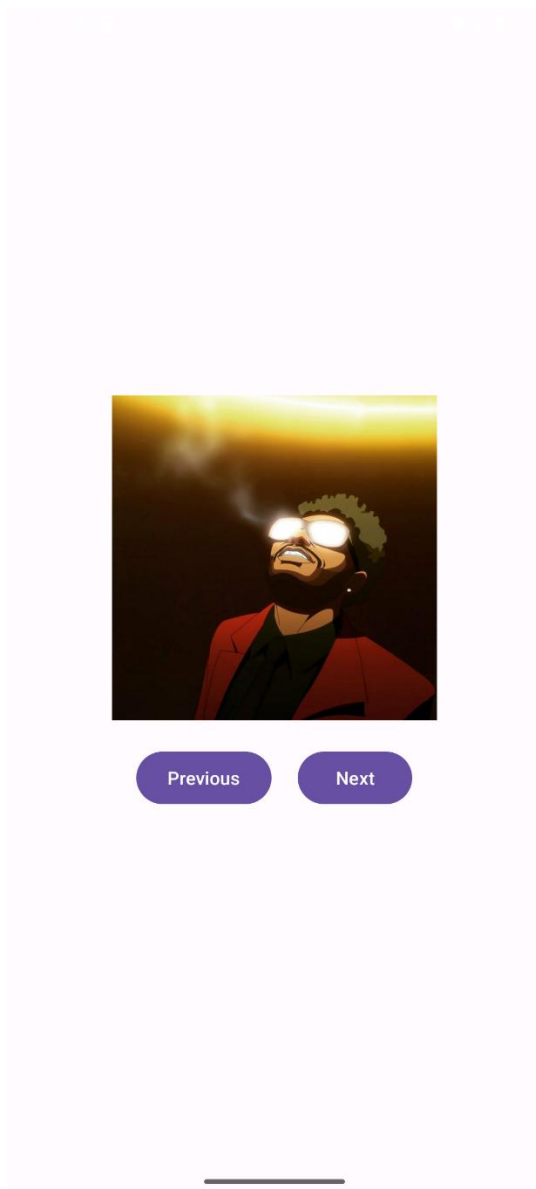
```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">
    <!-- Image container -->
    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="250dp"
        android:layout_height="250dp">
        <ImageView
            android:id="@+id/image1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:src="@drawable/img1"
            android:scaleType="centerCrop"
            android:visibility="visible" />
        <ImageView
            android:id="@+id/image2"
            android:layout_width="match_parent"

```

```
        android:layout_height="match_parent"
        android:src="@drawable/img2"
        android:scaleType="centerCrop"
        android:visibility="gone" />
    <ImageView
        android:id="@+id/image3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/img3"
        android:scaleType="centerCrop"
        android:visibility="gone" />
</FrameLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="20dp">
    <Button
        android:id="@+id/btnPrev"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Previous" />
    <Button
        android:id="@+id/btnNext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Next"
        android:layout_marginStart="20dp" />
</LinearLayout>
</LinearLayout>
```

## Output





Previous

Next



Previous

Next

### **Experiment 13: "Colour Picker with ListView"**

**Objective:** Learn to change the background color of an app's screen based on user selection from a ListView (ListBox).

**Question:** Create an app with a main screen that serves as the background. Add a ListView (ListBox) displaying at least three color options — for example: Red, Green, and Blue. When the user taps on a color from the list, change the background color of the screen to the selected color.

#### **MainActivity.java**

```
package com.example.experiment13;
import android.annotation.SuppressLint;
import android.graphics.Color;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.LinearLayout;
import android.widget.ListView;
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 {
    private LinearLayout mainLayout;
    private ListView colorList;
    private String[] colors = {"Red", "Green", "Blue"};
    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mainLayout = findViewById(R.id.mainLayout);
        colorList = findViewById(R.id.colorList);
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
            android.R.layout.simple_list_item_1, colors);
        colorList.setAdapter(adapter);
        colorList.setOnItemClickListener((parent, view, position, id) -> {
            switch (colors[position]) {
```

```

        case "Red":
            mainLayout.setBackgroundColor(Color.RED);
            break;
        case "Green":
            mainLayout.setBackgroundColor(Color.GREEN);
            break;
        case "Blue":
            mainLayout.setBackgroundColor(Color.BLUE);
            break;
    }
    });
}
}

```

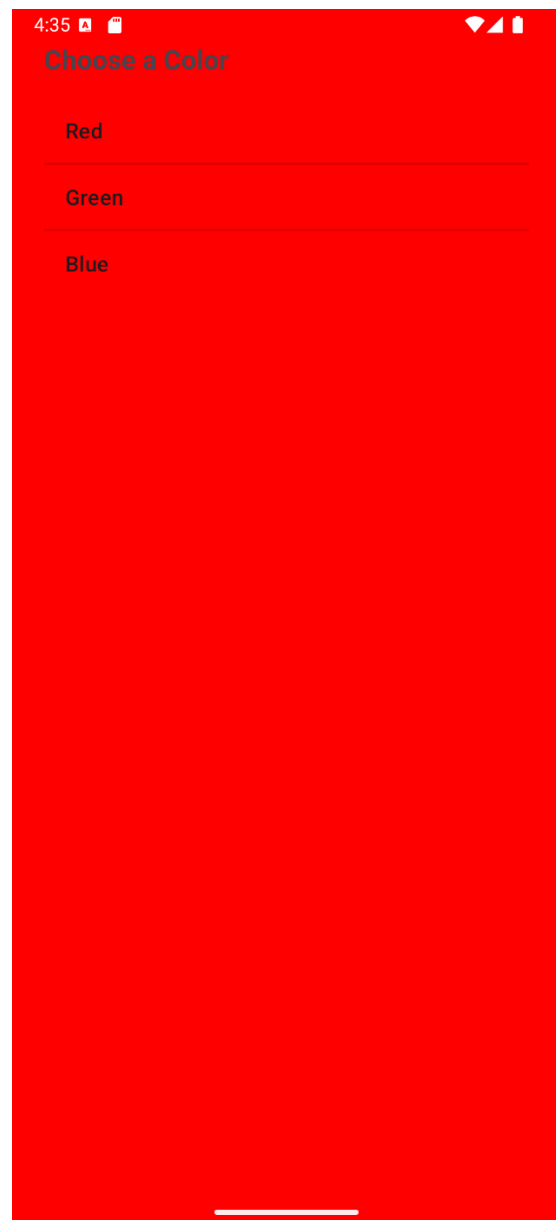
### **activity\_main.xml**

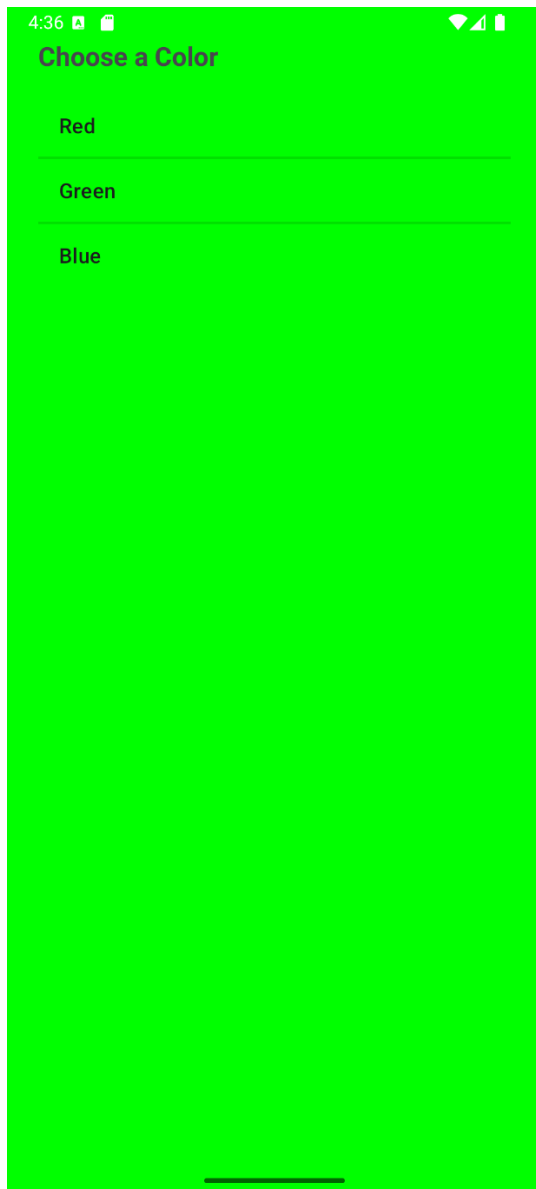
```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mainLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="24dp">
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Choose a Color"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp" />
    <ListView
        android:id="@+id/colorList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:dividerHeight="2dp" />
</LinearLayout>

```

## Output







## CO4: Intents, menus, inter-component navigation

### Experiment 14: "Multi-Activity Navigation"

**Objective:** To learn how to navigate between different activities (screens) in an application using Intents.

**Question:** Implement Intents to navigate between multiple activities in login application.

#### MainActivity.java

```
package com.example.experiment14;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText username;
    EditText password;
    Button loginButton;
    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username=findViewById(R.id.username);
        password=findViewById(R.id.password);
        loginButton=findViewById(R.id.loginButton);
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(username.getText().toString().equals("user")&&
                    password.getText().toString().equals("1234"))
                {
                    Intent intent = new Intent(MainActivity.this, MainActivity2.class);
```

```

        startActivity(intent);
    }
    else {
        Toast.makeText(MainActivity.this,"Login
Failed",Toast.LENGTH_SHORT).show();
    }
}

});
}
}

```

### **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"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/lbg"
    tools:context=".MainActivity">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp"
        android:background="@drawable/custom_edittext">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"
            android:padding="24dp">

```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:id="@+id/logintext"
    android:textSize="36sp"

    android:textAlignment="center"
    android:textStyle="bold"
    android:textColor="@color/black"/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:id="@+id/username"
    android:background="@drawable/custom_edittext"

    android:drawablePadding="8dp"
    android:hint="Username"
    android:padding="8dp"
    android:textColor="@color/black"
    android:textColorHighlight="@color/cardview_dark_background"
    android:layout_marginTop="40dp"/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:id="@+id/password"
    android:background="@drawable/custom_edittext"

    android:drawablePadding="8dp"
    android:hint="Password"
    android:padding="8dp"
    android:inputType="textPassword"
    android:textColor="@color/black"
    android:textColorHighlight="@color/cardview_dark_background"
    android:layout_marginTop="20dp"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:id="@+id/loginButton"
    android:text="Login"
```

```

        android:textSize="18sp"
        android:layout_marginTop="30dp"
        android:backgroundTint="@color/black"
        app:cornerRadius="20dp"/>

    </LinearLayout>
</androidx.cardview.widget.CardView>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="8dp"
    android:text="Not yet registered? SignUp Now."
    android:textSize="14sp"
    android:textAlignment="center"
    android:id="@+id/signupText"
    android:textColor="@color/purple"
    android:layout_marginBottom="20dp"/>
</LinearLayout>

```

### **MainActivity2.java**

```

package com.example.experiment14;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2); // Home screen layout
    }
}

```

### **activity\_main2.xml**

```

<!-- res/layout/activity_main2.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

```

```
<TextView
    android:id="@+id/tvWelcome"
    android:text="Welcome to Home Screen!"
    android:textSize="24sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

### **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.Experiment14">

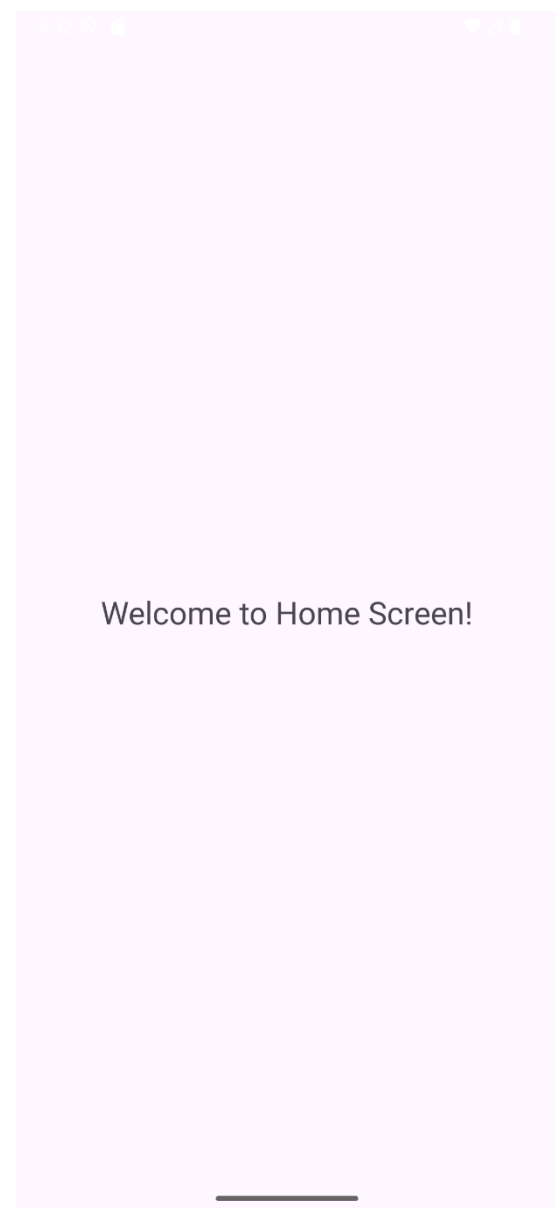
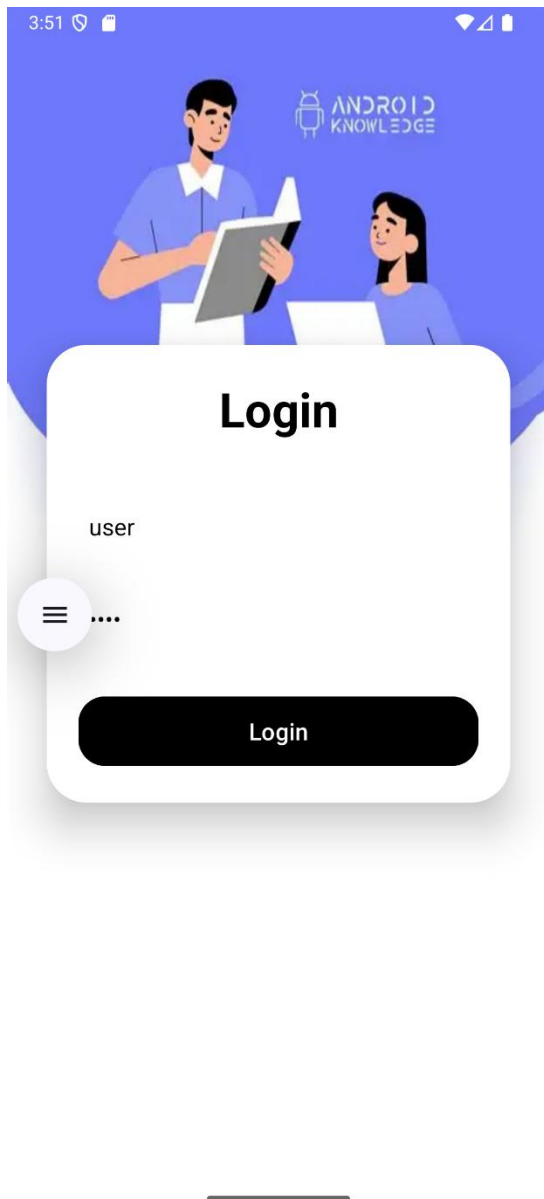
        <!-- Second Activity (MainActivity2) -->
        <activity android:name=".MainActivity2" />

        <!-- Main / Launcher Activity -->
        <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



### **Experiment 15: "Options Menu Navigation"**

**Objective:** To learn how to implement an Options Menu and use it to navigate to different activities.

**Question:** Implement an Options Menu in your application. Use the menu items to navigate to different activities.

#### **MainActivity.java**

```
package com.example.experiment15;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.options_menu, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.menu_first) {
            startActivity(new Intent(this, FirstActivity.class));
            return true;
        } else if (id == R.id.menu_second) {
            startActivity(new Intent(this, SecondActivity.class));
            return true;
        }
    }
}
```

```

    }
    return super.onOptionsItemSelected(item);
}
}

```

### **FirstActivity.java**

```

package com.example.experiment15;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class FirstActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
    }
}

```

### **SecondActivity.java**

```

package com.example.experiment15;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
    }
}

```

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

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"

```



```
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:title="Main Page"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/PopupMenuOverlayBlack" />
</LinearLayout>
```

### **activity\_first.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textFirst"
        android:text="This is First Activity"
        android:textSize="24sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

### **activity\_second.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textSecond"
        android:text="This is Second Activity"
        android:textSize="24sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

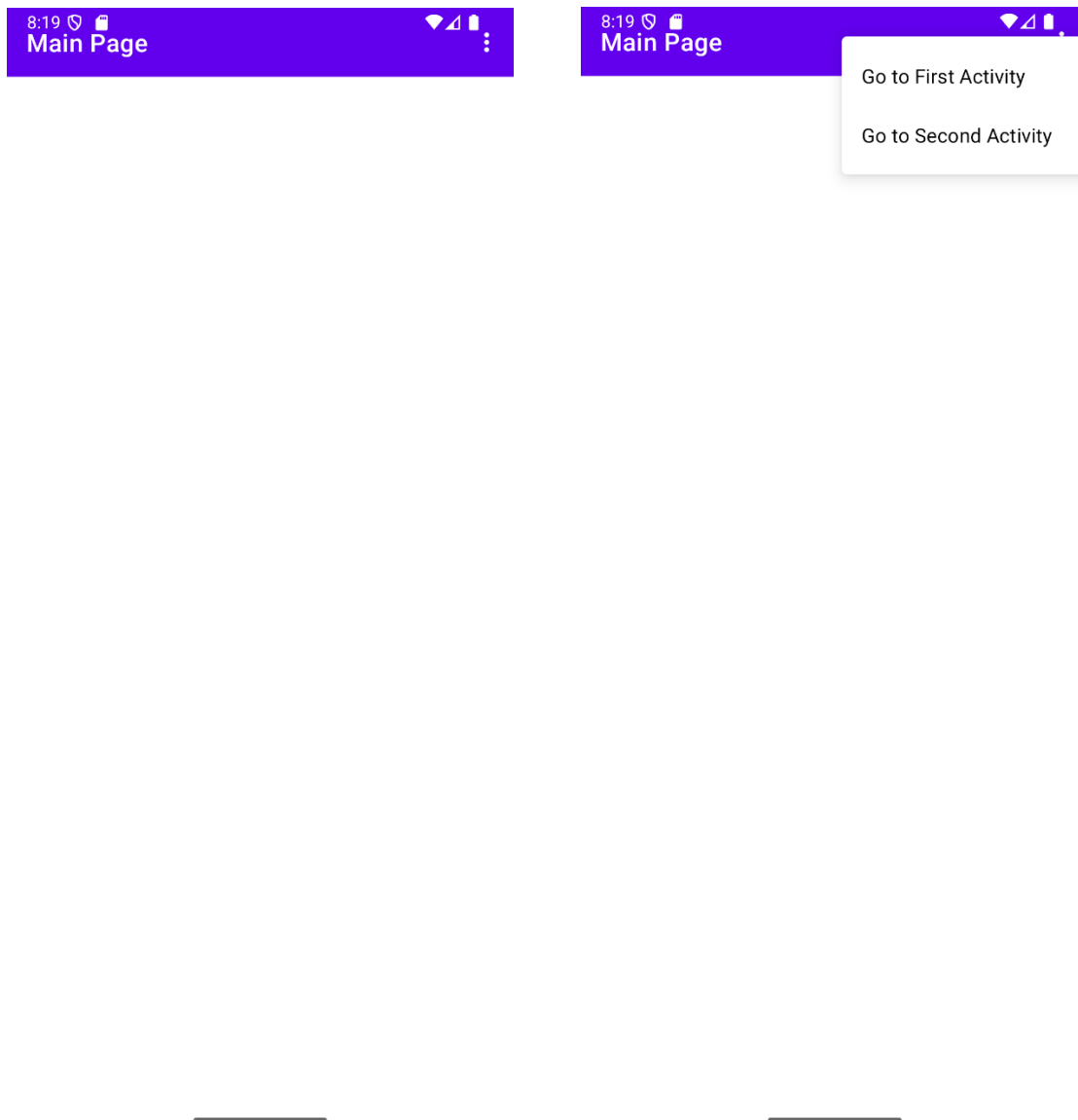
**Options menu.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_first"
        android:title="Go to First Activity" />
    <item
        android:id="@+id/menu_second"
        android:title="Go to Second Activity" />
</menu>
```

**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.Experiment15">
        <activity android:name=".FirstActivity" />
        <activity android:name=".SecondActivity" />
        <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



This is First Activity

This is Second Activity



## CO5: Styles, themes, drawable resources, responsive design patterns

### Experiment 16: "Working with Styles and Themes in Android"

**Objective:** To learn how to apply to Android applications in order to maintain a consistent and attractive UI design.

**Question:** Design an Android application with multiple UI components (e.g., TextView, Button, EditText). Create custom in styles.xml to define properties such as . Apply a global (light or dark) to the entire app. Demonstrate how changing the theme updates the look and feel of all UI components consistently.

#### MainActivity.java

```
package com.example.exp16;
import android.content.SharedPreferences;
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;
public class MainActivity extends AppCompatActivity {
    private boolean isDarkTheme;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
// Load theme preference before setting content view
        SharedPreferences prefs = getSharedPreferences("ThemePrefs",
MODE_PRIVATE);
        isDarkTheme = prefs.getBoolean("isDarkTheme", false);
// Apply the selected theme
        setTheme(isDarkTheme ? R.style.AppTheme_Dark : R.style.AppTheme_Light);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
// Theme toggle button logic
        Button toggleButton = findViewById(R.id.themeToggleButton);
        toggleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        isDarkTheme = !isDarkTheme;
        prefs.edit().putBoolean("isDarkTheme", isDarkTheme).apply();
        recreate(); // Recreate activity to apply new theme
    }
});
}
}

```

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

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
    <TextView
        style="@style/CustomTextView"
        android:text=" Theme Switcher"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        style="@style/CustomEditText"
        android:hint="Enter your name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Button
        style="@style/CustomButton"
        android:text="Submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/themeToggleButton"
        android:text="Toggle Theme Changer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

```

**styles.xml**

```

<resources>
    <!-- Base application theme -->
    <style name="AppTheme.Light"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <item name="colorPrimary">#EE0000</item>
        <item name="colorPrimaryVariant">#B30000</item>
        <item name="colorOnPrimary">#FFFFFF</item>
        <item name="android:textColor">#000000</item>
        <item name="android:background">#FFFFFF</item>
    </style>
    <style name="AppTheme.Dark"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <item name="colorPrimary">#00FB37</item>
        <item name="colorPrimaryVariant">#00FF6F</item>
        <item name="colorOnPrimary">#000000</item>
        <item name="android:textColor">#FFFFFF</item>
        <item name="android:background">#121212</item>
    </style>
    <!-- Custom UI component styles -->
    <style name="CustomTextView">
        <item name="android:textSize">20sp</item>
        <item name="android:textStyle">bold</item>
        <item name="android:layout_marginBottom">16dp</item>
    </style>
    <style name="CustomEditText">
        <item name="android:padding">12dp</item>
        <item name="android:backgroundTint">?attr/colorPrimary</item>
    </style>
    <style name="CustomButton">
        <item name="android:backgroundTint">?attr/colorPrimary</item>
        <item name="android:textColor">?attr/colorOnPrimary</item>
    </style>
</resources>

```

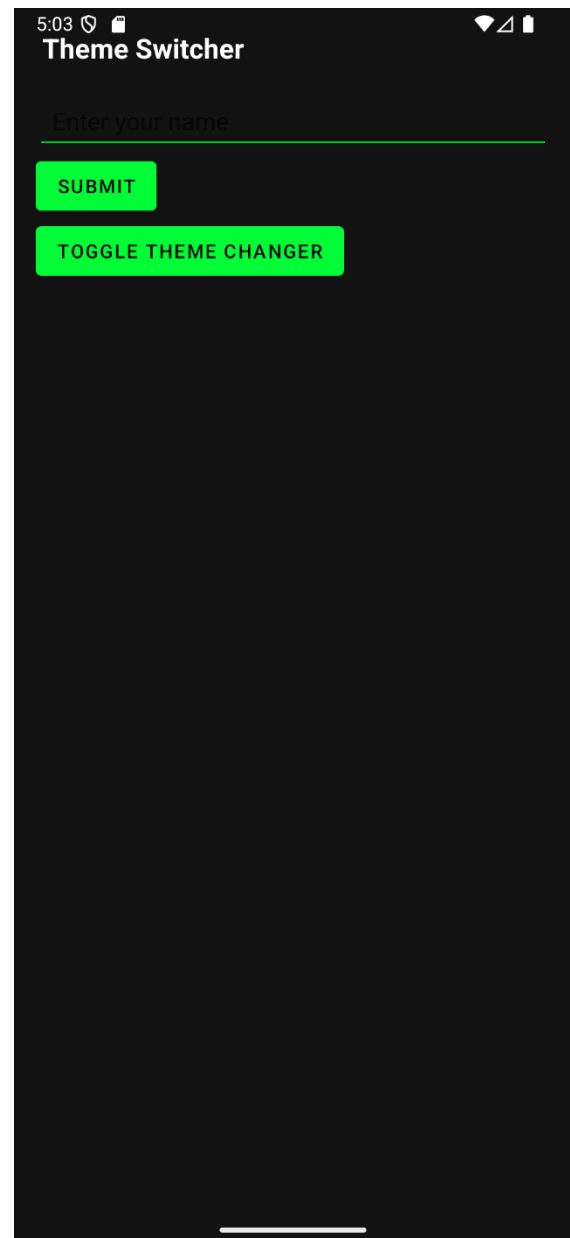
## Output

### Theme Switcher

Enter your name

SUBMIT

TOGGLE THEME CHANGER





### **Experiment 17: "Grid View & Alert Box"**

**Objective:** To learn how to display images in a GridView and show an Alert Box when an item is selected.

**Question:** Develop an application that uses a GridView with images. When an image is selected, display an Alert Box.

#### **MainActivity.java**

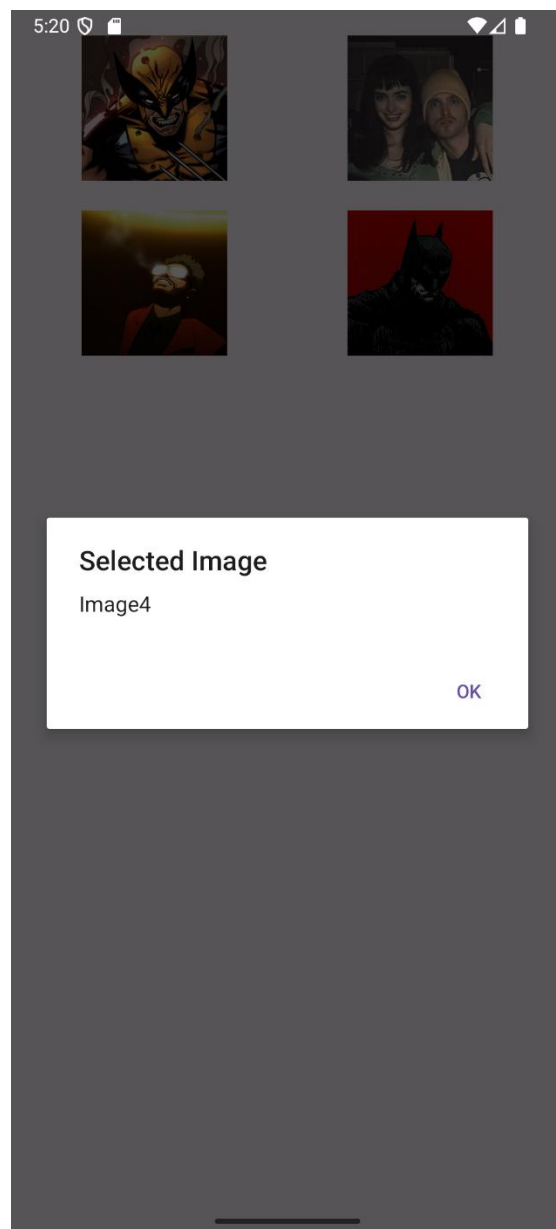
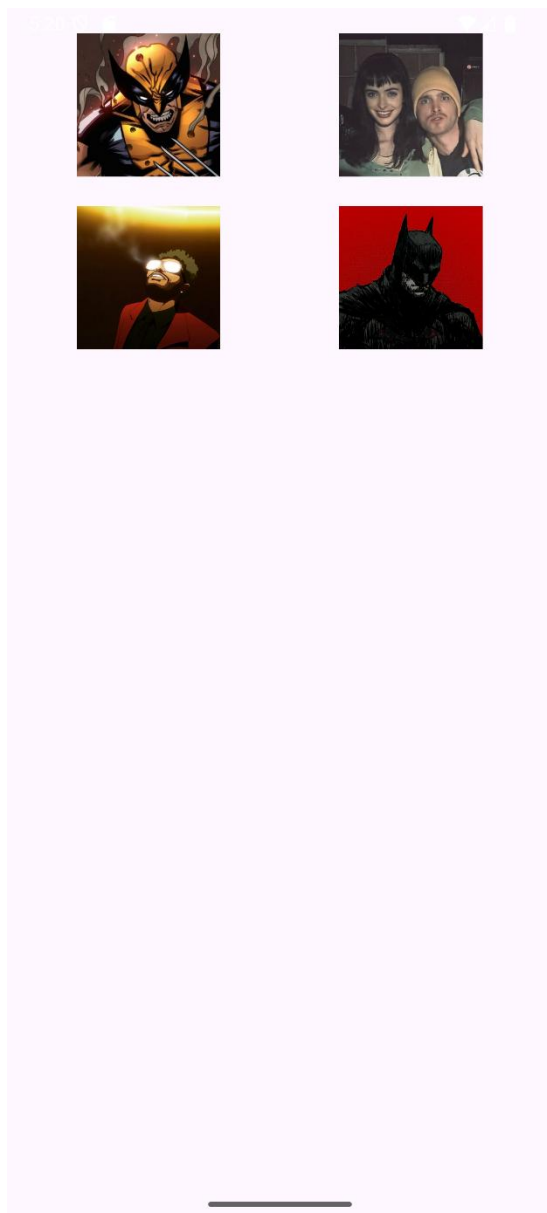
```
package com.example.exp17;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
public class MainActivity extends AppCompatActivity {
    int[] imageIds = {
        R.drawable.image1,
        R.drawable.image2,
        R.drawable.image3,
        R.drawable.image4,
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GridView gridView = findViewById(R.id.gridView);
        gridView.setAdapter(new ImageAdapter(this));
        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position,
            long id) {
```

```
        showAlert(position);
    }
});
}
private void showAlert(int position) {
    new AlertDialog.Builder(this)
        .setTitle("Selected Image")
        .setMessage("Image" + (position + 1))
        .setPositiveButton("OK", null)
        .show();
}
class ImageAdapter extends BaseAdapter {
    private Context context;
    ImageAdapter(Context c) {
        context = c;
    }
    public int getCount() {
        return imageIds.length;
    }
    public Object getItem(int position) {
        return imageIds[position];
    }
    public long getItemId(int position) {
        return position;
    }
    public View getView(int position, View convertView, android.view.ViewGroup
parent) {
        ImageView imageView;
        if (convertView == null) {
            imageView = new ImageView(context);
            imageView.setLayoutParams(new GridView.LayoutParams(300, 300));
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setPadding(8, 8, 8, 8);
        } else {
            imageView = (ImageView) convertView;
        }
        imageView.setImageResource(imageIds[position]);
        return imageView;
    }
}}
```

**activity\_main.xml**

```
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:numColumns="auto_fit"
    android:verticalSpacing="16dp"
    android:horizontalSpacing="16dp"
    android:padding="16dp"
    android:gravity="center"
    android:stretchMode="columnWidth" />
```

## Output



### **Experiment 18: "Spinner Component with Event Handling"**

**Objective:** To learn how to implement a Spinner component (dropdown menu) and handle the user's selection.

**Question:** Develop an application that implements a Spinner component. Populate it with items and perform an action (event handling) when an item is selected.

#### **MainActivity.java**

```
package com.example.exp18;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
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 {
    String[] items = { "Choose a Player", "Messi", "Ronaldo", "Neymar", "Xavi",
    "Iniesta" };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spinner = findViewById(R.id.mySpinner);
        TextView selectedText = findViewById(R.id.selectedItemText);
        ArrayAdapter<String> adapter = new ArrayAdapter<>(
            this, android.R.layout.simple_spinner_item, items
        );
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
        {
            @Override
```

```

        public void onItemSelected(AdapterView<?> parent, View view, int position,
long id) {
            String selected = items[position];
            selectedText.setText("Selected Player: " + selected);
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
            selectedText.setText("No player selected");
        }
    });
}
}

```

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

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="24dp"
    android:gravity="center">
    <TextView
        android:id="@+id/selectedItemText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Player Selected:"
        android:textSize="18sp"
        android:layout_marginBottom="20dp" />
    <Spinner
        android:id="@+id/mySpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:spinnerMode="dropdown" />
</LinearLayout>

```

## Output

