

Lab Record (Lab 1-4) Submission Report

LAB NO: 1 BASICS OF ANDROID MOBILE APPLICATION DEVELOPMENT TOOL

Objectives

- To familiarize with mobile application development tool.
- To gain knowledge about how to develop simple mobile application using android features.

Lab Exercises:

Q.1. Create an Android application to show the demo of displaying text with justifying elements, changing text colors ,fonts etc.

Ans. *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="16dp">

        <!-- Title Text -->
        <TextView
            android:id="@+id/titleText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Text Styling Demo"
            android:textSize="24sp"
            android:textColor="#1E88E5"
            android:textStyle="bold"
            android:gravity="center"
            android:paddingBottom="12dp" />

        <!-- Justified Paragraph -->
        <TextView
```

```
    android:id="@+id/justifiedText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:textColor="#333333"
    android:justificationMode="inter_word"
    android:paddingBottom="12dp"
    android:text="Android allows developers to customize text by changing colors, fonts,
sizes, and alignment. Justified text improves readability and gives a clean professional
appearance to paragraphs displayed in applications." />
```

```
<!-- Colored Text -->
<TextView
    android:id="@+id/coloredText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:paddingBottom="12dp"
    android:text="This text changes colors dynamically!" />
```

```
<!-- Font Style Text -->
<TextView
    android:id="@+id/fontText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Different font and italic style"
    android:textSize="18sp"
    android:textStyle="italic"
    android:textColor="#4CAF50" />

</LinearLayout>
</ScrollView>
```

MainActivity.java

```
package com.example.textstyle;

import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.style.ForegroundColorSpan;
```

```
import android.text.style.StyleSpan;
import android.graphics.Typeface;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

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

        // Creating multi-colored text
        SpannableString spannable = new SpannableString("This text changes colors dynamically!");

        spannable.setSpan(new ForegroundColorSpan(Color.RED),
            0, 9, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        spannable.setSpan(new ForegroundColorSpan(Color.BLUE),
            10, 17, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        spannable.setSpan(new ForegroundColorSpan(Color.MAGENTA),
            18, spannable.length(), Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        // Bold part of text
        spannable.setSpan(new StyleSpan(Typeface.BOLD),
            10, 17, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);

        coloredText.setText(spannable);
    }
}
```

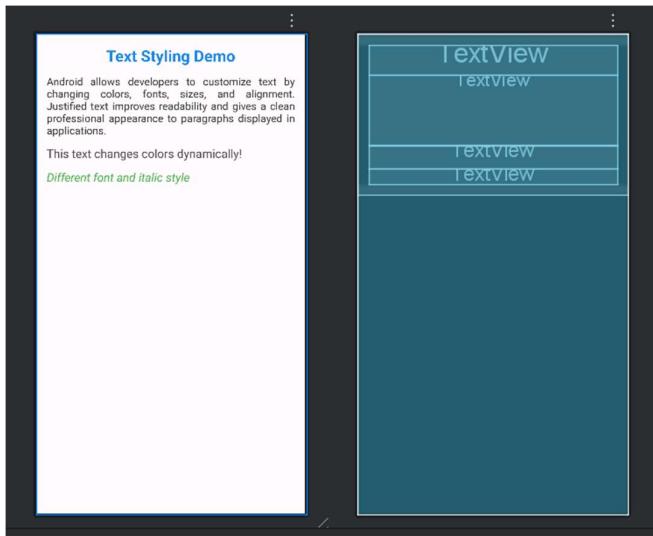


Fig 1.1: Q1 XML output



Fig 1.2: Q1 Interface Output

Q.2. Find the “hello word” text in the XML document and modify the text.

Ans. *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:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/helloText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello Android! Text Modified Successfully"
        android:textSize="22sp"
        android:textColor="#3F51B5"
        android:textStyle="bold" />

</LinearLayout>

```

ActivityMain.java

```

package com.example.myapplication;

import android.os.Bundle;

```

```

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

```

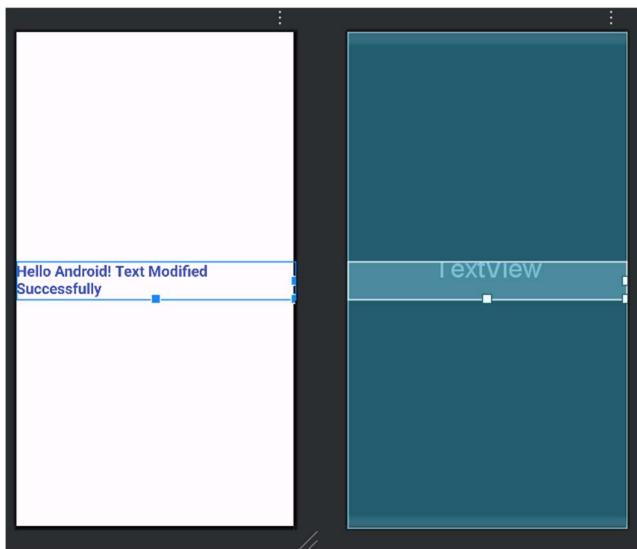


Fig 1.3: Q2 XML output

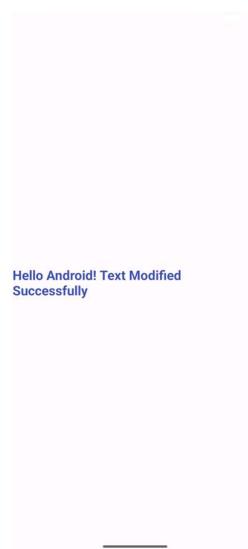


Fig 1.4: Q2 Interface Output

Questions Given in Class:

Q.1. Create an Android login UI using ConstraintLayout with Username and Password fields and Login and Clear buttons, positioning all views using constraints only.

Ans. *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:paddingStart="16dp"
    android:paddingEnd="16dp"
    tools:context=".MainActivity">

    <!-- Username label -->
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="Username"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
    <!-- Username input -->
    <EditText
        android:id="@+id/editTextText4"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter username"
        android:inputType="text"
        android:layout_marginStart="12dp"
        app:layout_constraintStart_toEndOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="@+id/textView2" />
```

```
    <!-- Password label -->
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="24dp"
        android:text="Password"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextText4" />
```

```
    <!-- Password input -->
    <EditText
        android:id="@+id/editTextText5"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter password"
```

```

        android:inputType="textPassword"
        android:layout_marginStart="12dp"
        app:layout_constraintStart_toEndOf="@+id/textView3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="@+id/textView3" />

    <!-- Login button -->

    <!-- Clear button -->
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:text="Login"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextText5" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Clear"
        app:layout_constraintStart_toEndOf="@+id/button3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="@+id/button3"
        android:layout_marginStart="24dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.loginconstraint;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

```

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

public class MainActivity extends AppCompatActivity {

    // Tag for logging
    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Enable EdgeToEdge for modern Android devices
        EdgeToEdge.enable(this);

        // Set your layout
        setContentView(R.layout.activity_main);

        // Adjust padding for system bars (status & navigation)
        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;
        });

        // ---- Link your views ----
        EditText etUsername = findViewById(R.id.editTextText4);
        EditText etPassword = findViewById(R.id.editTextText5);

        Button btnLogin = findViewById(R.id.button3);
        Button btnClear = findViewById(R.id.button4);

        // ---- Login button functionality ----
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String username = etUsername.getText().toString();
                String password = etPassword.getText().toString();

                // Simple toast showing entered username and password
                Toast.makeText(MainActivity.this,
```

```

    "Login clicked\nUsername: " + username + "\nPassword: " + password,
    Toast.LENGTH_SHORT).show();

// ---- Logcat output ----
Log.d(TAG, "Login button clicked");
Log.d(TAG, "Username: " + username);
Log.d(TAG, "Password: " + password);
}

});

// ---- Clear button functionality ----
btnClear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        etUsername.setText("");
        etPassword.setText("");

        // Simple toast
        Toast.makeText(MainActivity.this,
                "Cleared!", Toast.LENGTH_SHORT).show();

        // ---- Logcat output ----
        Log.d(TAG, "Clear button clicked, fields cleared");
    }
});
}
}

```

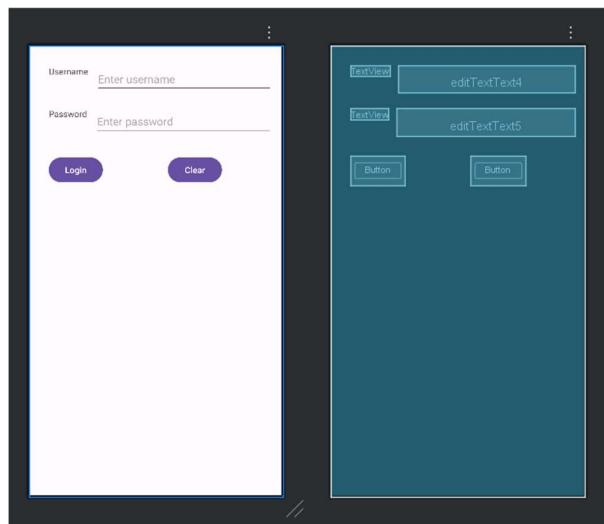


Fig 1.5: Q1 XML output



Fig 1.6: Q1 Interface Output

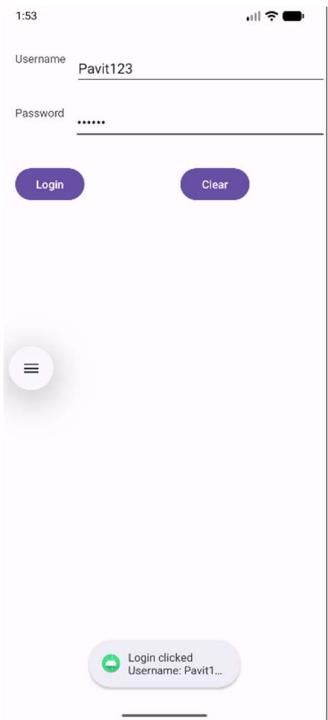


Fig 1.7: Output after clicking Login button

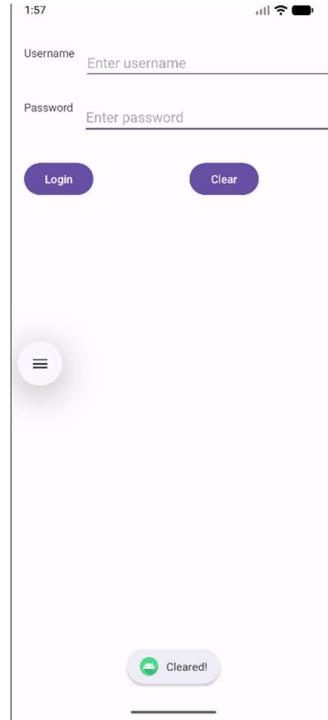


Fig 1.8: Output after clicking Login button

Q.2. Create an Android login screen using RelativeLayout with Username and Password fields and Login and Clear buttons, and implement button click actions using Toast and Logcat in Java.

Ans. *activity_main.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <!-- Username label -->
    <TextView
        android:id="@+id/tvUsername"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Username"
        android:layout_marginTop="32dp"
        android:layout_marginStart="16dp" />

    <!-- Username input -->
```

```
<EditText  
    android:id="@+id/etUsername"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_toEndOf="@+id/tvUsername"  
    android:layout_alignTop="@+id/tvUsername"  
    android:layout_marginStart="12dp"  
    android:hint="Enter username"  
    android:inputType="text" />
```

```
<!-- Password label -->  
<TextView  
    android:id="@+id/tvPassword"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/etUsername"  
    android:layout_marginTop="24dp"  
    android:text="Password"  
    android:layout_marginStart="16dp" />
```

```
<!-- Password input -->  
<EditText  
    android:id="@+id/etPassword"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_toEndOf="@+id/tvPassword"  
    android:layout_alignTop="@+id/tvPassword"  
    android:layout_marginStart="12dp"  
    android:hint="Enter password"  
    android:inputType="textPassword" />
```

```
<!-- Login button -->  
<Button  
    android:id="@+id/btnLogin"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/etPassword"  
    android:layout_marginTop="32dp"  
    android:layout_marginStart="16dp"  
    android:text="Login" />
```

```
<!-- Clear button -->  
<Button  
    android:id="@+id/btnClear"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toEndOf="@+id/btnLogin"
        android:layout_alignTop="@+id/btnLogin"
        android:layout_marginStart="24dp"
        android:text="Clear" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

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

public class MainActivity extends AppCompatActivity {

    // Tag for logging
    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Enable EdgeToEdge for modern Android devices
        EdgeToEdge.enable(this);

        // Set the layout
        setContentView(R.layout.activity_main);

        // Adjust padding for system bars (status & navigation)
        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;
});

// ---- Link your views ----
EditText etUsername = findViewById(R.id.etUsername);
EditText etPassword = findViewById(R.id.etPassword);

Button btnLogin = findViewById(R.id.btnLogin);
Button btnClear = findViewById(R.id.btnClear);

// ---- Login button click ----
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String username = etUsername.getText().toString();
        String password = etPassword.getText().toString();

        // Show Toast
        Toast.makeText(MainActivity.this,
                "Login clicked\nUsername: " + username + "\nPassword: " + password,
                Toast.LENGTH_SHORT).show();

        // ---- Logcat output ----
        Log.d(TAG, "Login button clicked");
        Log.d(TAG, "Username: " + username);
        Log.d(TAG, "Password: " + password);
    }
});

// ---- Clear button click ----
btnClear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        etUsername.setText("");
        etPassword.setText("");

        // Show Toast
        Toast.makeText(MainActivity.this,
                "Cleared!", Toast.LENGTH_SHORT).show();
    }
});
```

```

// ---- Logcat output ----
Log.d(TAG, "Clear button clicked, fields cleared");
}
});
}
}
}

```

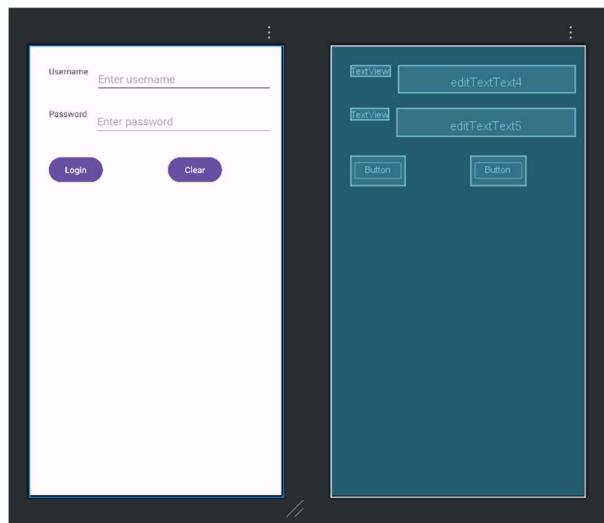


Fig 1.9: Q2 XML output

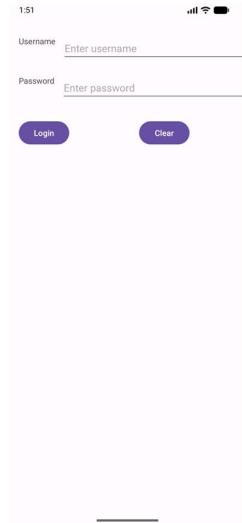


Fig 1.10: Q2 Interface Output

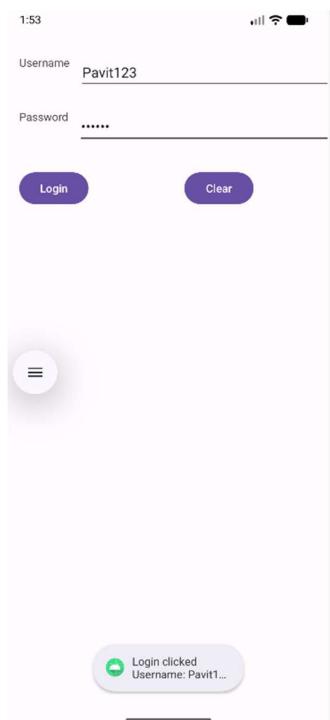


Fig 1.11: Output after clicking Login button

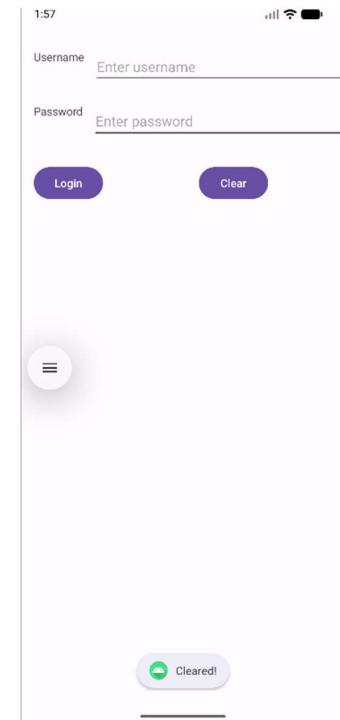


Fig 1.12: Output after clicking Login button

LAB NO: 2 INTRODUCTION TO ACTIVITY AND LAYOUTS IN ANDROID

Objectives

- To apply the concepts of layouts to enrich the user interface:
- Understanding layout types and how to use them effectively improves the UI design, enhancing the user experience.
- To understand different activity and fragments to use in android application.

Lab Exercises:

Q.1. Create an app that illustrates that the activity lifecycle method being triggered by various action. Understand when onCreate(), onStart(), onResume event occur.

Ans. *activity_main.xml*

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

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

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

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.util.Log;
```

```
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "LifecycleDemo";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate called");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(TAG, "onStart called");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(TAG, "onResume called");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "onPause called");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop called");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");
    }
}
```

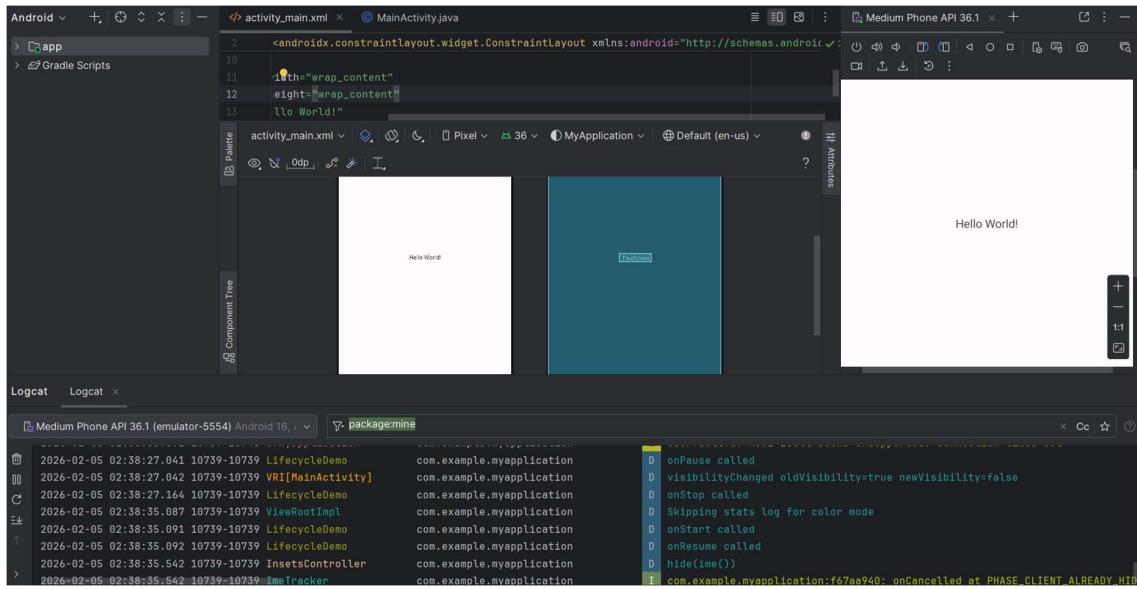


Fig 2.1: Q1 output at logcat

Q.2. Create a Calculator app that does the function of multiplication, addition, division, subtraction but displays the result in the format of:-Num1 operator num2 = result. Back button on the next activity should get back to the calculator activity again.

Ans. *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="20dp">

    <EditText
        android:id="@+id/num1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Number 1"
        android:inputType="number" />

    <EditText
        android:id="@+id/num2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Number 2"
        android:inputType="number" />
    
```

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Add"  
    android:onClick="add" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Subtract"  
    android:onClick="sub" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Multiply"  
    android:onClick="mul" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Divide"  
    android:onClick="div" />  
  
<TextView  
    android:id="@+id/resultText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Result will appear here"  
    android:textSize="18sp"  
    android:paddingTop="20dp" />  
  
</LinearLayout>
```

MainActivity.java

```
package com.example.myapplication;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;
```

```
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    EditText n1, n2;
    TextView resultText;

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

        n1 = findViewById(R.id.num1);
        n2 = findViewById(R.id.num2);
        resultText = findViewById(R.id.resultText);
    }

    public void add(View v) {
        calculate("+");
    }

    public void sub(View v) {
        calculate("-");
    }

    public void mul(View v) {
        calculate("*");
    }

    public void div(View v) {
        calculate("/");
    }

    private void calculate(String op) {

        int a = Integer.parseInt(n1.getText().toString());
        int b = Integer.parseInt(n2.getText().toString());
        int res = 0;

        switch (op) {
            case "+":
                res = a + b;
                break;
```

```

        case "-":
            res = a - b;
            break;
        case "*":
            res = a * b;
            break;
        case "/":
            res = a / b;
            break;
    }

    String output = a + " " + op + " " + b + " = " + res;
    resultText.setText(output);
}

```

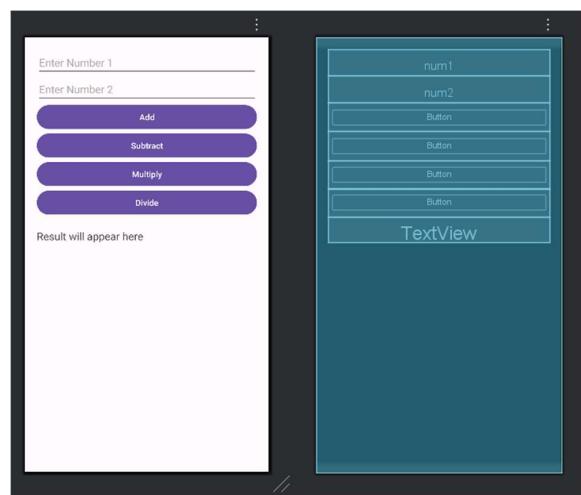


Fig 2.2: Q2 XML output

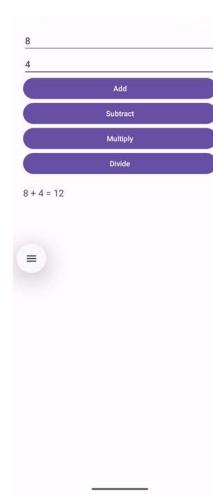
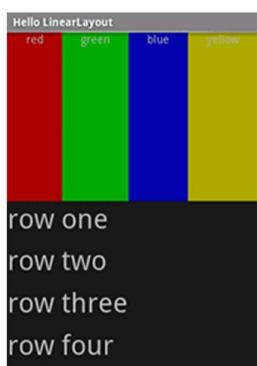


Fig 2.3: Q2 Interface Output

Q.3. Create the following given scenario using linear and relative layout concept.



Ans. Part1) *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="20dp">

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

        <!-- Top Text -->
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:text="Hello LinearLayout"
            android:textSize="20sp"
            android:textStyle="bold" />

        <!-- Color Blocks (Horizontal LinearLayout) -->
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="120dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:background="#FF0000"
                android:gravity="center"
                android:text="red"
                android:textColor="#FFFFFF" />

            <TextView
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:background="#00FF00"
                android:gravity="center"
```

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

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#0000FF"
        android:gravity="center"
        android:text="blue"
        android:textColor="#FFFFFF" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="#CCCC00"
        android:gravity="center"
        android:text="yellow" />
    </LinearLayout>

    <!-- Bottom Section (RelativeLayout) -->
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#000000"
        android:padding="10dp">

        <TextView
            android:id="@+id/row1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="row one"
            android:textColor="#FFFFFF"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/row2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/row1"
            android:layout_marginTop="10dp"
            android:text="row two"
            android:textColor="#FFFFFF"
            android:textSize="18sp" />
    
```

```
<TextView  
    android:id="@+id/row3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/row2"  
    android:layout_marginTop="10dp"  
    android:text="row three"  
    android:textColor="#FFFFFF"  
    android:textSize="18sp" />  
  
<TextView  
    android:id="@+id/row4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/row3"  
    android:layout_marginTop="10dp"  
    android:text="row four"  
    android:textColor="#FFFFFF"  
    android:textSize="18sp" />  
  
</RelativeLayout>  
  
</LinearLayout>  
  
<EditText  
    android:id="@+id/num1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Enter Number 1"  
    android:inputType="number" />  
  
<EditText  
    android:id="@+id/num2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Enter Number 2"  
    android:inputType="number" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Add"  
    android:onClick="add" />
```

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Subtract"  
    android:onClick="sub" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Multiply"  
    android:onClick="mul" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Divide"  
    android:onClick="div" />  
  
<TextView  
    android:id="@+id/resultText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Result will appear here"  
    android:textSize="18sp"  
    android:paddingTop="20dp" />  
  
</LinearLayout>
```

MainActivity.java

```
package com.example.myapplication;  
  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        // Connect Java with activity_main.xml
```

```

        setContentView(R.layout.activity_main);

    }

}

```

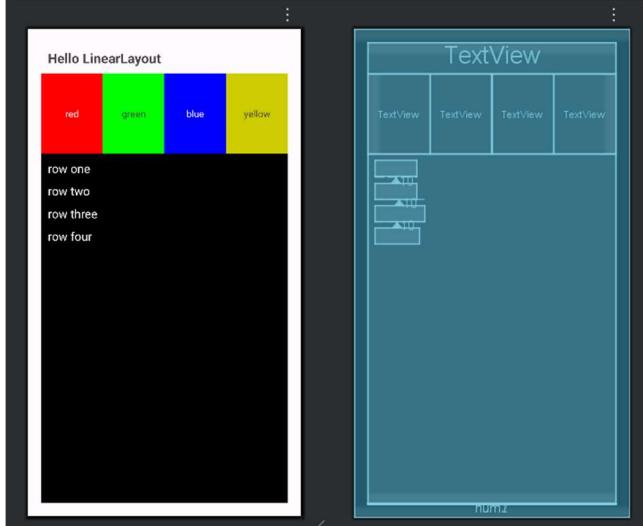


Fig 2.4: Q3 Part1 XML output

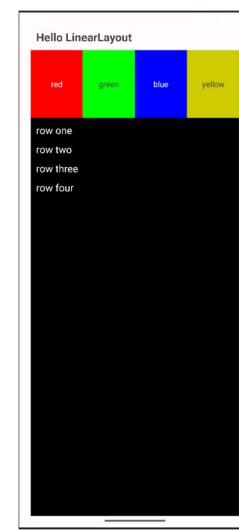


Fig 2.5: Q2 Part1 Interface Output

Ans. Part2) *activity_main.xml*

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    android:padding="16dp">

    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="27dp"
        android:layout_marginBottom="16dp"
        android:text="Hello RelativeLayout"
        android:textColor="#FFFFFF"
        android:textSize="18sp" />

    <TextView

```

```
    android:id="@+id/label"
    android:text="Type here:"
    android:textColor="#FFFFFF"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/title"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="8dp" />
```

```
<EditText
    android:id="@+id/input"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_below="@id/label"
    android:layout_alignParentStart="true"
    android:background="@android:drawable/edit_text"
    android:padding="8dp"
    android:textColor="#000000"
    android:layout_marginBottom="16dp" />
```

```
<Button
    android:id="@+id/button_cancel"
    android:text="Cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/input"
    android:layout_alignParentStart="true"
    android:layout_marginEnd="8dp" />
```

```
<Button
    android:id="@+id/button_ok"
    android:text="OK"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/input"
    android:layout_toEndOf="@id/button_cancel" />
```

```
</RelativeLayout>
```

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;
import android.text.TextUtils;
import android.widget.Button;
import android.widget.EditText;
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); // Set layout

        // Get views
        EditText inputEditText = findViewById(R.id.input);
        Button cancelButton = findViewById(R.id.button_cancel);
        Button okButton = findViewById(R.id.button_ok);

        // Cancel button clears input
        cancelButton.setOnClickListener(v -> {
            inputEditText.setText("");
            Toast.makeText(MainActivity.this, "Input cleared", Toast.LENGTH_SHORT).show();
        });

        // OK button shows input text
        okButton.setOnClickListener(v -> {
            String text = inputEditText.getText().toString().trim();
            if (TextUtils.isEmpty(text)) {
                Toast.makeText(MainActivity.this, "Please enter something",
                        Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(MainActivity.this, "You typed: " + text,
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

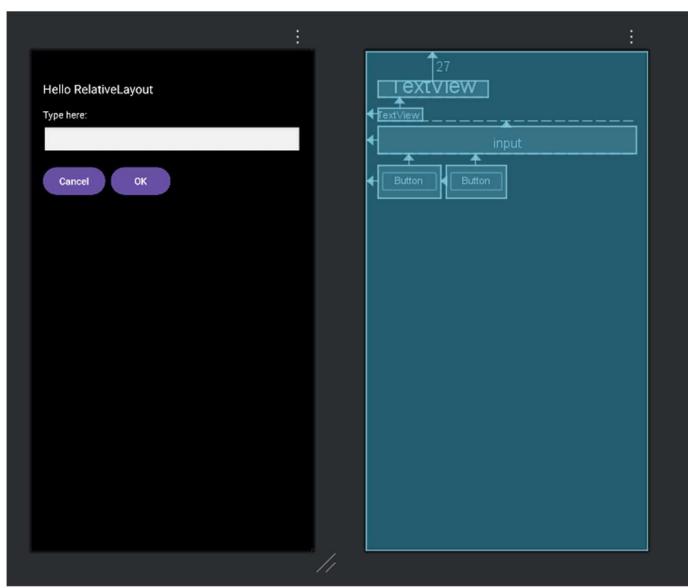


Fig 2.6: Q3 Part1 XML output

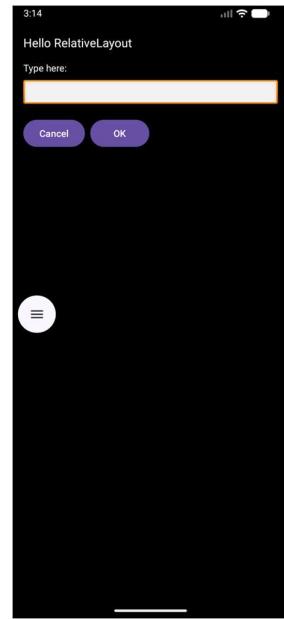


Fig 2.7: Q3 Part2 Interface Output

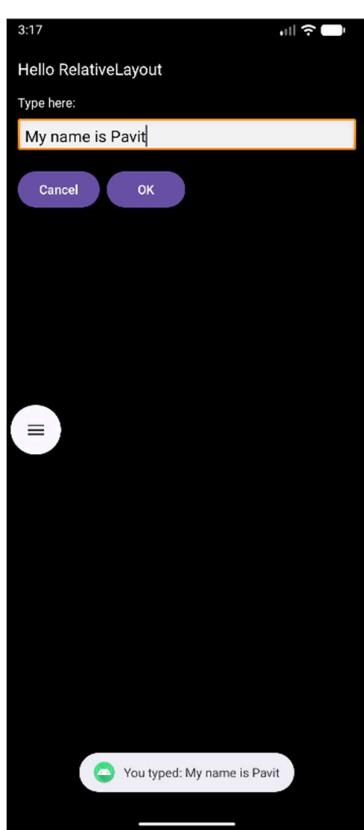


Fig 2.8: Clicked OK

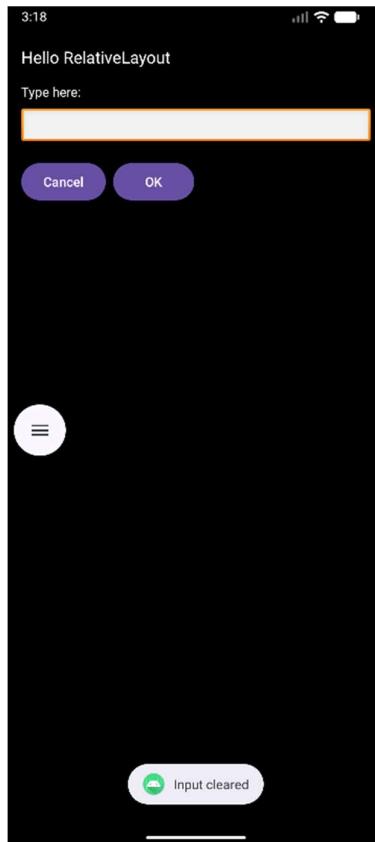


Fig 2.9: Clicked Cancel

Q.4. Create an app such that when the user click on the given URL typed by the user, it visits the corresponding page.

Ans. *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="20dp">

    <EditText
        android:id="@+id/editTextUrl"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter URL"
        android:inputType="textUri" />

    <Button
        android:id="@+id/buttonOpen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open URL"
        android:layout_marginTop="20dp" />

</LinearLayout>
```

MainActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.example.myapplication.R;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
```

```
EditText editTextUrl;
Button buttonOpen;

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

    editTextUrl = findViewById(R.id.editTextUrl);
    buttonOpen = findViewById(R.id.buttonOpen);

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

            String url = editTextUrl.getText().toString().trim();

            if (url.isEmpty()) {
                Toast.makeText(MainActivity.this, "Please enter a URL",
Toast.LENGTH_SHORT).show();
                return;
            }

            // Add https:// if user didn't type it
            if (!url.startsWith("http://") && !url.startsWith("https://")) {
                url = "https://" + url;
            }

            Intent intent = new Intent(Intent.ACTION_VIEW);
            intent.setData(Uri.parse(url));
            startActivity(intent);
        }
    });
}
```

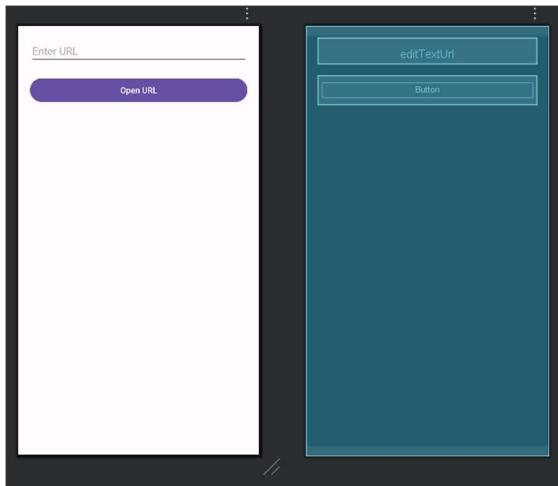


Fig 2.10: Q4 XML output



Fig 2.11: Interface

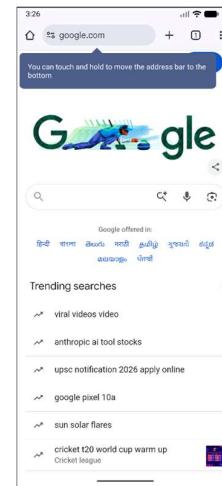


Fig 2.12: Google

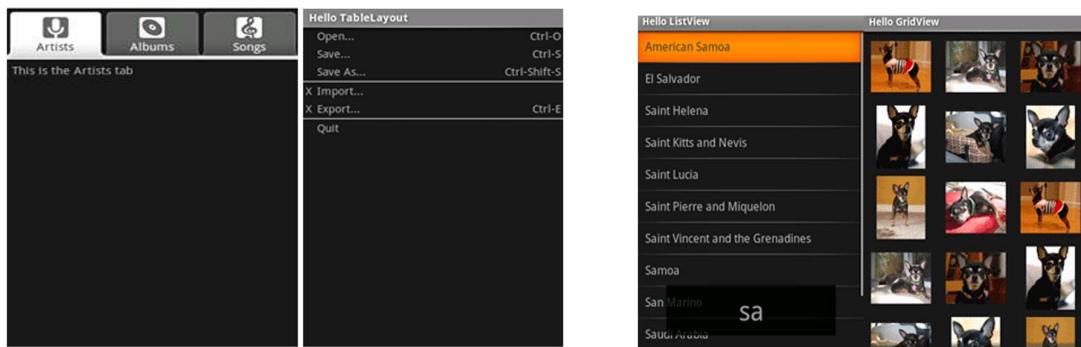
LAB NO: 3 ACTIVITY AND LAYOUT CONTINUED

Objectives

- To acquire knowledge on list view, grid view, table view in android application.
- To develop interactive mobile application with multiple pages.

Q.1. Using given scenario develop an application to perform following layout operations

- List view
- Grid view
- Tab layout
- Table layout



Ans. *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"
    android:fitsSystemWindows="true"
    android:paddingTop="24dp">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabMode="fixed"
        app:tabGravity="fill"/>

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"/>

    </LinearLayout>
```

tab_grid.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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="3"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:stretchMode="columnWidth">

</GridView>
```

tab_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</ListView>
```

tab_table.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <TableRow>
        <TextView android:text="Name" android:padding="8dp"/>
        <TextView android:text="Age" android:padding="8dp"/>
    </TableRow>

    <TableRow>
```

```
<TextView android:text="Ashwin" android:padding="8dp"/>
<TextView android:text="25" android:padding="8dp"/>
</TableRow>

<TableRow>
    <TextView android:text="Sumit" android:padding="8dp"/>
    <TextView android:text="22" android:padding="8dp"/>
</TableRow>

</TableLayout>
```

MainActivity.java

```
package com.example.myapplication;

import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.ListView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.PagerAdapter;
import androidx.viewpager.widget.ViewPager;

import com.google.android.material.tabs.TabLayout;

public class MainActivity extends AppCompatActivity {

    ViewPager viewPager;
    TabLayout tabLayout;

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

        tabLayout = findViewById(R.id.tabLayout);
        viewPager = findViewById(R.id.viewPager);
```

```
// Set adapter
viewPager.setAdapter(new MyPagerAdapter(this));
tabLayout.setupWithViewPager(viewPager);
}

// ◆ INNER PAGER ADAPTER CLASS
private class MyPagerAdapter extends PagerAdapter {

    Context context;
    int[] layouts = {
        R.layout.tab_list,
        R.layout.tab_grid,
        R.layout.tab_table
    };

    MyPagerAdapter(Context context) {
        this.context = context;
    }

    @Override
    public int getCount() {
        return layouts.length;
    }

    @Override
    public boolean isViewFromObject(View view, Object object) {
        return view == object;
    }

    @Override
    public CharSequence getPageTitle(int position) {
        switch (position) {
            case 0: return "ListView";
            case 1: return "GridView";
            case 2: return "TableLayout";
            default: return "";
        }
    }

    @Override
    public Object instantiateItem(ViewGroup container, int position) {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(layouts[position], container, false);

        // Setup ListView
    }
}
```

```
if (position == 0) {
    ListView listView = view.findViewById(R.id.listView);
    String[] countries = {
        "American Samoa",
        "El Salvador",
        "Saint Helena",
        "Saint Lucia",
        "Samoa",
        "San Marino"
    };
    ArrayAdapter<String> adapter =
        new ArrayAdapter<>(context,
            android.R.layout.simple_list_item_1, countries);
    listView.setAdapter(adapter);
}

// Setup GridView
if (position == 1) {
    GridView gridView = view.findViewById(R.id.gridView);
    String[] animals = {
        "Dog", "Cat", "Tiger",
        "Lion", "Horse", "Fox"
    };
    ArrayAdapter<String> adapter =
        new ArrayAdapter<>(context,
            android.R.layout.simple_list_item_1, animals);
    gridView.setAdapter(adapter);
}

container.addView(view);
return view;
}

@Override
public void destroyItem(ViewGroup container, int position, Object object) {
    container.removeView((View) object);
}
}
```

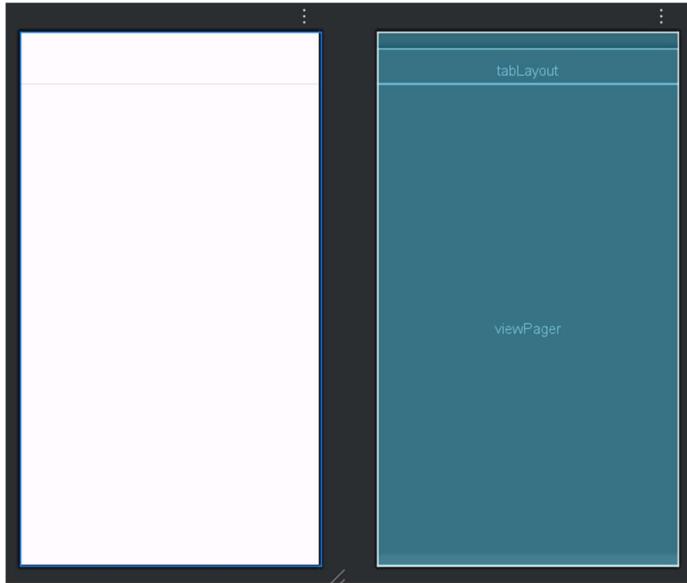


Fig 3.1: XML Output

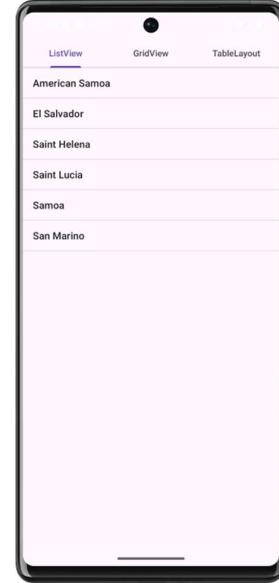


Fig 3.2: ListView



Fig 3.3: GridView



Fig 3.4: TableLayout

Q.2. Write a program to list different sports, and when a sport is selected, display a message showing the selected sport.

Ans. *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">
```

```

        android:background="#F5F5F5"
        android:padding="16dp"
        android:fitsSystemWindows="true">

    <!-- App Title -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Select Your Favorite Sport"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="#000"
        android:gravity="center"
        android:padding="12dp" />

    <!-- Card-style container -->
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:cardElevation="6dp"
        app:cardCornerRadius="12dp"
        android:layout_marginTop="12dp">

        <ListView
            android:id="@+id/sportsListView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:divider="#DDDDDD"
            android:dividerHeight="1dp"
            android:padding="8dp" />
    </androidx.cardview.widget.CardView>

</LinearLayout>

```

MainActivity.java

```

package com.example.sportsactivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

```

```
public class MainActivity extends AppCompatActivity {

    ListView sportsListView;

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

        sportsListView = findViewById(R.id.sportsListView);

        // Sports data
        String[] sports = {
            "Cricket",
            "Football",
            "Hockey",
            "Basketball",
            "Tennis",
            "Badminton",
            "Volleyball",
            "Swimming"
        };

        // Adapter
        ArrayAdapter<String> adapter = new ArrayAdapter<>(
            this,
            android.R.layout.simple_list_item_1,
            sports
        );

        sportsListView.setAdapter(adapter);

        // Click listener
        sportsListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                String selectedSport = sports[position];

                Toast.makeText(
                    MainActivity.this,
                    "Selected Sport: " + selectedSport,
                    Toast.LENGTH_SHORT
                ).show();
            }
        });
    }
}
```

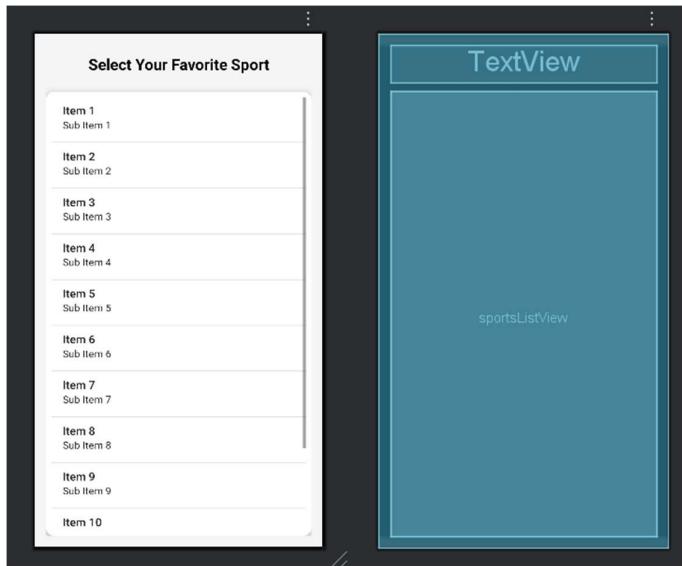


Fig 3.5: XML Output

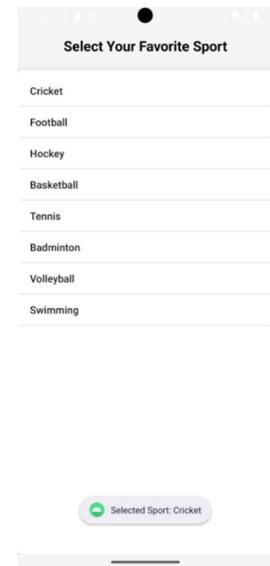


Fig 3.6: Interface

Q.3. Design an news application and Implement the navigation between sections like Top Stories, Sports, and Entertainment using tab layout.

Ans. *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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <!-- App Title -->
    <TextView
        android:id="@+id/titleText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/daily_news"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>
```

```

<!-- Tab Layout -->
<com.google.android.material.tabs.TabLayout
    android:id="@+id/tabLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    app:tabIndicatorColor="@color/black"
    app:tabSelectedTextColor="@color/black"
    app:tabTextColor="@color/darker_gray"
    app:layout_constraintTop_toBottomOf="@+id/titleText"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

<!-- ViewPager -->
<androidx.viewpager2.widget.ViewPager2
    android:id="@+id/viewPager"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintTop_toBottomOf="@+id/tabLayout"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Fragment_news.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:padding="16dp"
    android:orientation="vertical">

<TextView
    android:id="@+id/sectionTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textStyle="bold"
    android:textColor="@color/black"/>

```

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

MainActivity.java

```
package com.example.newsapp;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.viewpager2.adapter.FragmentStateAdapter;  
import androidx.viewpager2.widget.ViewPager2;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.android.material.tabs.TabLayoutMediator;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        TabLayout tabLayout = findViewById(R.id.tabLayout);  
        ViewPager2 viewPager = findViewById(R.id.viewPager);  
  
        ViewPagerAdapter adapter = new ViewPagerAdapter(this);  
        viewPager.setAdapter(adapter);  
  
        new TabLayoutMediator(tabLayout, viewPager,
```

```
(tab, position) -> {
    switch (position) {
        case 0:
            tab.setText("Top Stories");
            break;
        case 1:
            tab.setText("Sports");
            break;
        case 2:
            tab.setText("Entertainment");
            break;
    }
}
).attach();
}

// -----
// ViewPager Adapter (Merged)
// -----
class ViewPagerAdapter extends FragmentStateAdapter {

    public ViewPagerAdapter(@NonNull AppCompatActivity activity) {
        super(activity);
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        switch (position) {
            case 0:
                return NewsFragment.newInstance(
                    "Top Stories",
                    "Latest breaking news from around the world.");
            case 1:
                return NewsFragment.newInstance(
                    "Sports",
                    "Live scores, match highlights, and sports updates.");
            case 2:
                return NewsFragment.newInstance(
                    "Entertainment",
                    "Movies, celebrities, music, and TV news.");
        }
    }
}
```

```
    default:
        return new Fragment();
    }
}

@Override
public int getItemCount() {
    return 3;
}
}

// -----
// News Fragment (Merged)
// -----
public static class NewsFragment extends Fragment {

    private static final String ARG_TITLE = "title";
    private static final String ARG_CONTENT = "content";

    public static NewsFragment newInstance(String title, String content) {
        NewsFragment fragment = new NewsFragment();
        Bundle args = new Bundle();
        args.putString(ARG_TITLE, title);
        args.putString(ARG_CONTENT, content);
        fragment.setArguments(args);
        return fragment;
    }

    @Nullable
    @Override
    public View onCreateView(
        @NonNull LayoutInflater inflater,
        @Nullable ViewGroup container,
        @Nullable Bundle savedInstanceState) {

        View view = inflater.inflate(R.layout.fragment_news, container, false);

        TextView titleText = view.findViewById(R.id.sectionTitle);
        TextView contentText = view.findViewById(R.id.sectionContent);

        if (getArguments() != null) {
            titleText.setText(getArguments().getString(ARG_TITLE));
            contentText.setText(getArguments().getString(ARG_CONTENT));
        }
    }
}
```

```
        return view;
    }
}
```



Fig 3.7: XML Output



Fig 3.8: Top Stories

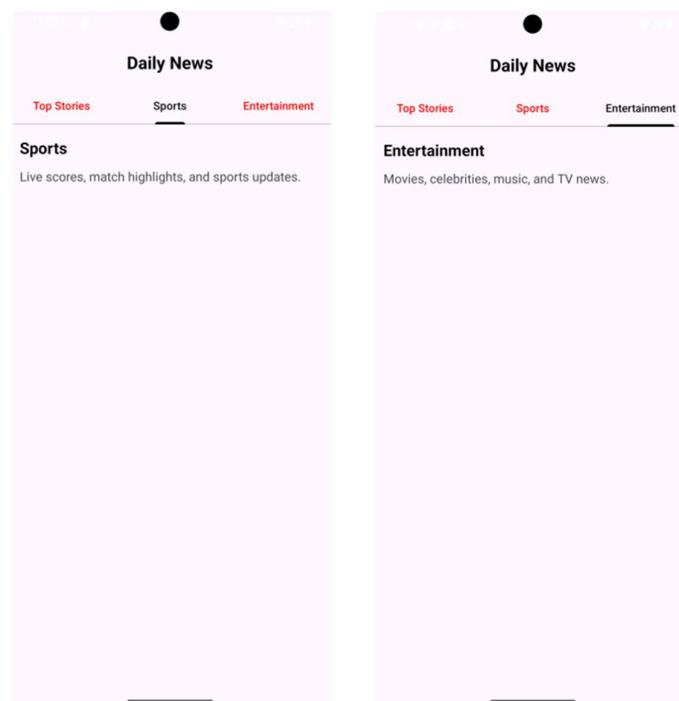


Fig 3.9: Sports

Fig 3.10: Entertainment

LAB NO: 4 INPUT CONTROLS IN ANDROID

Objectives

- To learn the usage of interactive components in application.
- To learn the interactive mobile application development using variety of control inputs.

Q.1. Develop a "Test App" that includes a layout with a Button and a ToggleButton. When each button is clicked, a custom Toast message should be displayed with different images as their content.

Ans. *main_activity.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">

    <Button
        android:id="@+id	btnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Button" />

    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="ON"
        android:textOff="OFF"
        android:layout_marginTop="20dp" />

</LinearLayout>
```

toast_layout.xml

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

```
    android:orientation="horizontal"
    android:padding="10dp"
    android:background="#AA000000">>

    <ImageView
        android:id="@+id/toastImage"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_marginRight="10dp" />

    <TextView
        android:id="@+id/toastText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFFFFF"
        android:textSize="16sp" />

</LinearLayout>
```

MainActivity.java

```
package com.example.testapp;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    Button btnClick;
    ToggleButton toggleButton;

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

```
btnClick = findViewById(R.id.btnClick);
toggleButton = findViewById(R.id.toggleButton);

// Normal Button click
btnClick.setOnClickListener(v ->
    showCustomToast("Button Clicked", R.drawable.button_img)
);

// ToggleButton click
toggleButton.setOnClickListener(v -> {
    if (toggleButton.isChecked()) {
        showCustomToast("Toggle ON", R.drawable.toggle_on);
    } else {
        showCustomToast("Toggle OFF", R.drawable.toggle_off);
    }
});

private void showCustomToast(String message, int imageRes) {

    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.toast_layout, null);

    ImageView imageView = view.findViewById(R.id.toastImage);
    TextView textView = view.findViewById(R.id.toastText);

    imageView.setImageResource(imageRes);
    textView.setText(message);

    Toast toast = new Toast(getApplicationContext());
    toast.setDuration(Toast.LENGTH_SHORT);
    toast.setView(view);
    toast.show();
}
```

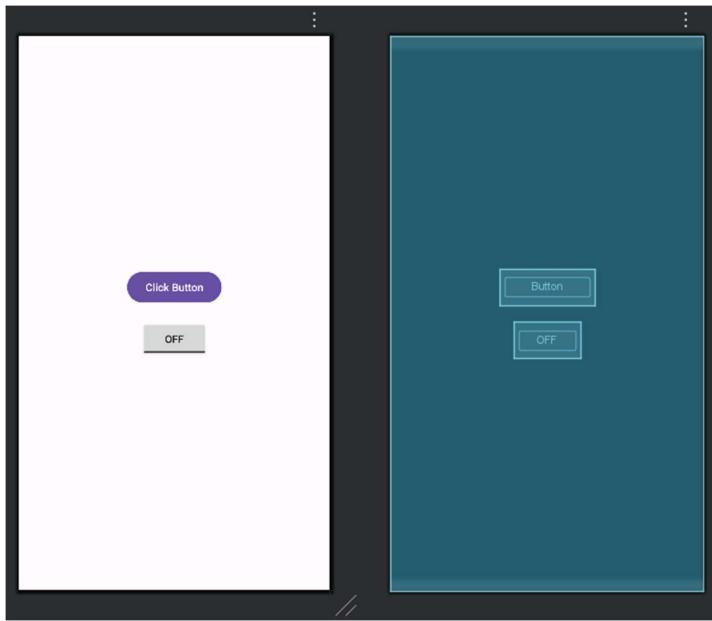


Fig 4.1: XML Output

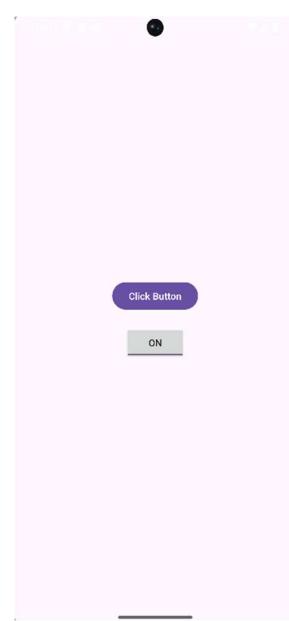


Fig 4.2: Interface

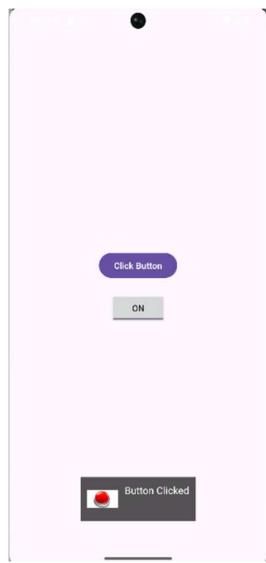


Fig 4.3: Button Clicked

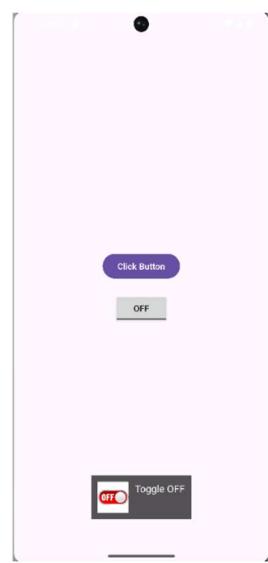


Fig 4.4: Toggle OFF

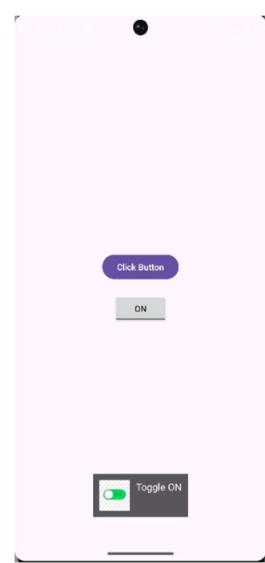


Fig 4.5: Toggle ON

Q.2. Create an app that contains a view with multiple buttons, each labeled with different Android versions. When a button is clicked, a Toast message should appear, displaying the corresponding Android version's name along with its associated icon.

Ans. *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">

    <Button
        android:id="@+id(btnLollipop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Android Lollipop" />

    <Button
        android:id="@+id	btnMarshmallow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Android Marshmallow"
        android:layout_marginTop="10dp" />

    <Button
        android:id="@+id	btnNougat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Android Nougat"
        android:layout_marginTop="10dp" />

</LinearLayout>
```

toast_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp"
    android:background="#AA000000">

    <ImageView
        android:id="@+id/toastImage"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_marginRight="10dp" />

    <TextView
```

```
        android:id="@+id/toastText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFFFFF"
        android:textSize="16sp" />

    </LinearLayout>
```

MainActivity.java

```
package com.example.androidversions;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    Button btnLollipop, btnMarshmallow, btnNougat;

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

        btnLollipop = findViewById(R.id.btnLollipop);
        btnMarshmallow = findViewById(R.id.btnMarshmallow);
        btnNougat = findViewById(R.id.btnNougat);

        btnLollipop.setOnClickListener(v ->
            showCustomToast("Android Lollipop", R.drawable.android_lollipop)
        );

        btnMarshmallow.setOnClickListener(v ->
            showCustomToast("Android Marshmallow", R.drawable.android_marshmallow)
        );
    }
}
```

```

        btnNougat.setOnClickListener(v ->
            showCustomToast("Android Nougat", R.drawable.android_nougat)
        );
    }

    private void showCustomToast(String message, int imageRes) {
        LayoutInflator inflater = getLayoutInflater();
        View view = inflater.inflate(R.layout.toast_layout, null);

        ImageView imageView = view.findViewById(R.id.toastImage);
        TextView textView = view.findViewById(R.id.toastText);

        imageView.setImageResource(imageRes);
        textView.setText(message);

        Toast toast = new Toast(getApplicationContext());
        toast.setDuration(Toast.LENGTH_SHORT);
        toast.setView(view);
        toast.show();
    }
}

```

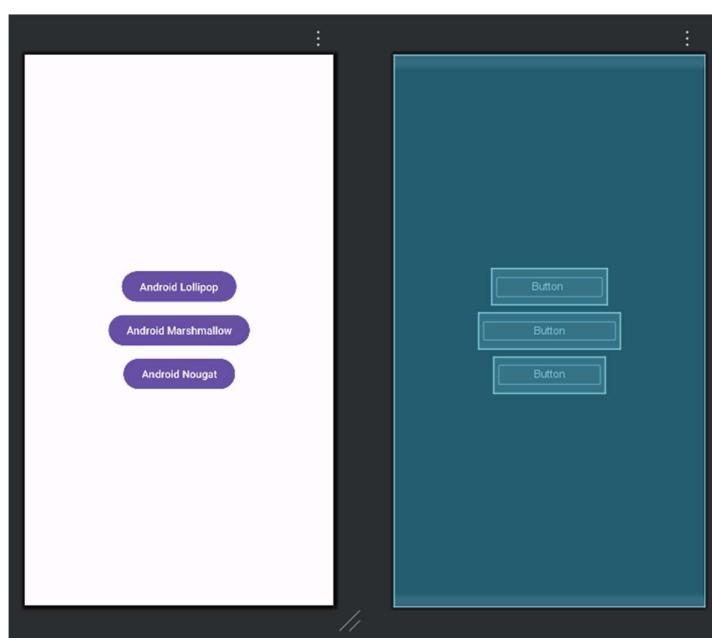


Fig 4.6: XML Output



Fig 4.7: Interface



Fig 4.8: Lollipop



Fig 4.9: Marshmallow



Fig 4.10: Nougat

Q.3. Develop a view with a ToggleButton labeled "Current Mode" that has two states: "Wi-Fi" and "Mobile Data." Based on the state of the toggle button, an image corresponding to the selected mode should appear, and a Toast message should display the current mode. Additionally, when the user clicks the "Change Mode" button, the app should switch to the corresponding mode and update the image accordingly.

Ans. *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">

    <ToggleButton
        android:id="@+id/toggleMode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="Wi-Fi"
        android:textOff="Mobile Data" />

    <ImageView
        android:id="@+id/modelImage"
        android:layout_width="150dp"
        android:layout_height="150dp" />

```

```

        android:layout_height="150dp"
        android:layout_marginTop="20dp"
        android:src="@drawable/wifi" />

    <Button
        android:id="@+id	btnChangeMode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Change Mode"
        android:layout_marginTop="20dp" />

</LinearLayout>

```

MainActivity.java

```

package com.example.currentmodeapp;

import android.os.Bundle;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import android.widget.ToggleButton;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    ToggleButton toggleMode;
    Button btnChangeMode;
    ImageView modelImage;

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

        toggleMode = findViewById(R.id.toggleMode);
        btnChangeMode = findViewById(R.id.btnChangeMode);
        modelImage = findViewById(R.id.modelImage);

        // ToggleButton state change
        toggleMode.setOnClickListener(v -> updateMode());

        // Change Mode button click
        btnChangeMode.setOnClickListener(v -> {
            toggleMode.setChecked(!toggleMode.isChecked());
            updateMode();
        });
    }

    private void updateMode() {
        if (toggleMode.isChecked()) {
            modelImage.setImageResource(R.drawable.wifi);
        } else {
            modelImage.setImageResource(R.drawable.signal);
        }
    }
}

```

```

    });
}

private void updateMode() {
    if (toggleMode.isChecked()) {
        modelImage.setImageResource(R.drawable.wifi);
        Toast.makeText(this, "Current Mode: Wi-Fi", Toast.LENGTH_SHORT).show();
    } else {
        modelImage.setImageResource(R.drawable.mobile_data);
        Toast.makeText(this, "Current Mode: Mobile Data", Toast.LENGTH_SHORT).show();
    }
}

```

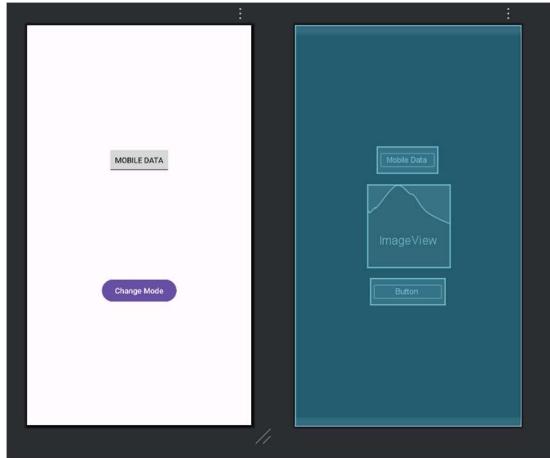


Fig 4.11: XML Output



Fig 4.12: Wifi

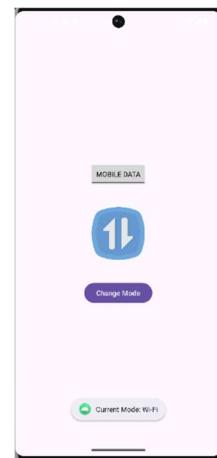


Fig 4.13: Mobile Data

Q.4. Create a “Food Ordering App” which lists food items with check boxes. Once the user checks /unchecks the item and click on the submit button display the items ordered along with cost of each item and total cost in a new activity. Once the user clicks on the submit button he/she should not be allowed to change the order i.e. he should not be allowed to change the state of the items checked.

Ans. *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"
    android:fitsSystemWindows="true">

    <LinearLayout
        android:layout_width="match_parent"

```

```
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="24dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Food Ordering App"
        android:textSize="22sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp" />

    <CheckBox
        android:id="@+id/cbPizza"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pizza - ₹200" />

    <CheckBox
        android:id="@+id/cbBurger"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Burger - ₹120" />

    <CheckBox
        android:id="@+id/cbPasta"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pasta - ₹150" />

    <Button
        android:id="@+id	btnSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit Order"
        android:layout_marginTop="24dp" />

</LinearLayout>

</ScrollView>
```

MainActivity.java

```
package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    CheckBox cbPizza, cbBurger, cbPasta;
    Button btnSubmit;

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

        cbPizza = findViewById(R.id.cbPizza);
        cbBurger = findViewById(R.id.cbBurger);
        cbPasta = findViewById(R.id.cbPasta);
        btnSubmit = findViewById(R.id.btnSubmit);

        btnSubmit.setOnClickListener(v -> {

            StringBuilder orderDetails = new StringBuilder();
            int totalCost = 0;

            if (cbPizza.isChecked()) {
                orderDetails.append("Pizza - ₹200\n");
                totalCost += 200;
            }

            if (cbBurger.isChecked()) {
                orderDetails.append("Burger - ₹120\n");
                totalCost += 120;
            }

            if (cbPasta.isChecked()) {
                orderDetails.append("Pasta - ₹150\n");
                totalCost += 150;
            }
        });
    }
}
```

```

        }

        orderDetails.append("\nTotal Cost: ₹").append(totalCost);

        // Disable checkboxes after submit
        cbPizza.setEnabled(false);
        cbBurger.setEnabled(false);
        cbPasta.setEnabled(false);
        btnSubmit.setEnabled(false);

        // Send data to next activity
        Intent intent = new Intent(MainActivity.this, OrderSummaryActivity.class);
        intent.putExtra("order", orderDetails.toString());
        startActivity(intent);
    });

}

}

```

activity_order_summary.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">

    <TextView
        android:id="@+id/tvOrderSummary"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp" />

</LinearLayout>

```

OrderSummaryActivity.java

```

package com.example.myapplication;

import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

```

```

public class OrderSummaryActivity extends AppCompatActivity {

    TextView tvOrderSummary;

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

        tvOrderSummary = findViewById(R.id.tvOrderSummary);

        String order = getIntent().getStringExtra("order");
        tvOrderSummary.setText(order);
    }
}

```

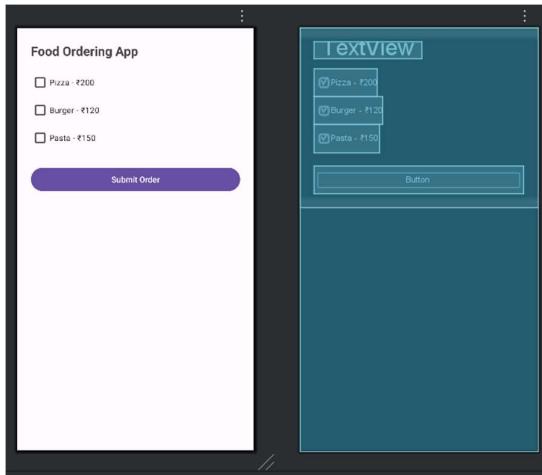


Fig 4.14: XML Output

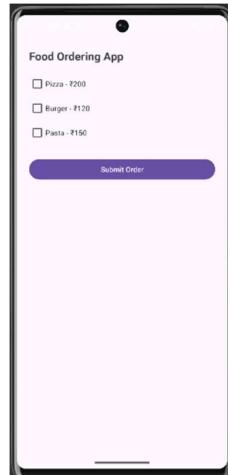


Fig 4.15: App

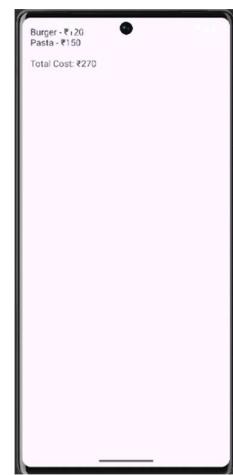


Fig 4.15: Order Bill