Name of Student: Pushkar Sane		
Roll Number: 45		Lab Assignment Number: 2
Title of Lab Assignment: To study different types of layouts and Toast class.		
DOP: 04-09-2024		DOS: 05-09-2024
CO Mapped: CO1, CO2	PO Mapped: PO1, PO2, PO3, PSO1	Signature:

Practical No. 2

Aim: To study different types of layouts and Toast class.

1. Design a Registration form to show the working of different layouts.
2. Create an application to design a simple calculator to perform addition, subtraction, multiplication and division. Show message for divide by zero error using Toast.
3. Create an application for Unit Conversion.

Theory:

Layouts in Android

Layouts in Android define how UI elements are arranged on the screen. They are crucial for creating user interfaces. Here are some common types of layouts:

1. LinearLayout
   - Orientation: Can be vertical or horizontal.
   - Usage: Arranges child elements in a single row or column.
   - Example: A vertical LinearLayout might stack buttons one on top of the other.
2. RelativeLayout
   - Positioning: Allows you to position elements relative to each other or to the parent container.
   - Usage: Useful when you need flexible positioning of elements.
   - Example: Positioning a button below a text view or aligning an image to the right of a text view.
3. ConstraintLayout
   - Design: Allows you to create complex layouts with a flat view hierarchy by defining constraints between views.
   - Usage: Provides more flexibility and better performance compared to nested layouts.
   - Example: Positioning elements in a grid or aligning them relative to each other or the parent.
4. FrameLayout
   - Stacking: Displays child elements stacked on top of each other.
   - Usage: Useful for overlapping views or when only one child view is expected.
   - Example: Overlaying a loading spinner over other UI components.

5. GridLayout
   - Structure: Organizes child views into a grid.
   - Usage: Useful for creating forms or tables where elements need to align in rows and columns.
   - Example: A calendar view where each cell represents a day of the month.
6. TableLayout
   - Arrangement: Arranges child views into rows and columns, similar to a table.
   - Usage: Useful for organizing data in a tabular format.
   - Example: Creating a layout for a calculator with buttons arranged in rows and columns.

Toast Class in Android

Toast is a class in Android used for displaying brief messages to the user. These messages pop up on the screen for a short duration and then automatically disappear. They are typically used for feedback or notifications that do not require user interaction.

- Usage: Toast is used to provide simple feedback to the user, such as confirming an action or displaying a small piece of information.
- Methods:
  - Toast.makeText(Context context, CharSequence text, int duration)
    - Parameters:
      - context: The context in which the Toast should appear.
      - text: The message to be displayed.
      - duration: The duration for which the Toast should be visible (Toast.LENGTH_SHORT or Toast.LENGTH_LONG).
  - show()
    - Usage: Displays the Toast message on the screen.

Code:

1. **Registration form**

   MainActivity.java

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

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Uncomment the layout you want to test
        setContentView(R.layout.linearlayout); // LinearLayout
        // setContentView(R.layout.relativelayout); // RelativeLayout
        // setContentView(R.layout.constraintlayout); // ConstraintLayout
        // setContentView(R.layout.gridlayout); // GridLayout
    }
}
```

Constraint Layout.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:padding="16dp">

    <EditText
        android:id="@+id/username"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="text"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/email"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/username"
        app:layout_constraintVertical_chainStyle="packed" />

    <EditText
        android:id="@+id/password"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/email" />

    <Button
        android:id="@+id/registerButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/password"
        app:layout_constraintVertical_bias="0.1" />

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

LinearLayout.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">

    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="text" />

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/registerButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_gravity="center_horizontal" />

</LinearLayout>
```

GridLayout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:columnCount="1"
    android:rowCount="5"
    android:orientation="vertical">
    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="text" />

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/registerButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_row="4"
        android:layout_gravity="center_horizontal" />
</GridLayout>
```

RelativeLayout.xml

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

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:layout_below="@id/username"
        android:layout_marginTop="16dp" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_below="@id/email"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/registerButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_below="@id/password"
        android:layout_marginTop="24dp"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

2. **Calculator**

MainActivity.java

```
package com.example.calculator;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      EditText input1 = findViewById(R.id.input1);
      EditText input2 = findViewById(R.id.input2);
      Button addButton = findViewById(R.id.addButton);
      Button subtractButton = findViewById(R.id.subtractButton);
      Button multiplyButton = findViewById(R.id.multiplyButton);
      Button divideButton = findViewById(R.id.divideButton);
      TextView resultView = findViewById(R.id.resultView);

      addButton.setOnClickListener(v -> performOperation("add", input1, input2, resultView));
      subtractButton.setOnClickListener(v -> performOperation("subtract", input1, input2, resultView));
      multiplyButton.setOnClickListener(v -> performOperation("multiply", input1, input2, resultView));
      divideButton.setOnClickListener(v -> performOperation("divide", input1, input2, resultView));
   }
```

```
    private void performOperation(String operation, EditText input1, EditText input2,
TextView resultView) {
        try {
            double num1 = Double.parseDouble(input1.getText().toString());
            double num2 = Double.parseDouble(input2.getText().toString());
            double result = 0;

            switch (operation) {
                case "add":
                    result = num1 + num2;
                    break;
                case "subtract":
                    result = num1 - num2;
                    break;
                case "multiply":
                    result = num1 * num2;
                    break;
                case "divide":
                    if (num2 == 0) {
                        Toast.makeText(this, "Cannot divide by zero",
Toast.LENGTH_SHORT).show();
                        return;
                    }
                    result = num1 / num2;
                    break;
            }

            resultView.setText(String.valueOf(result));
        } catch (NumberFormatException e) {
            Toast.makeText(this, "Invalid input", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Main_Activity.xml

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

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <EditText
        android:id="@+id/input1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:hint="Enter number"
        android:inputType="numberDecimal"
        android:layout_marginBottom="16dp"
        android:padding="8dp"/>

    <EditText
        android:id="@+id/input2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/input1"
        android:layout_centerHorizontal="true"
        android:hint="Enter number"
        android:inputType="numberDecimal"
        android:layout_marginBottom="16dp"
        android:padding="8dp"/>
    <Button
        android:id="@+id/addButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/input2"
        android:layout_centerHorizontal="true"
        android:text="Add" />

    <Button
        android:id="@+id/subtractButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/addButton"
        android:layout_centerHorizontal="true"
        android:text="Subtract" />

    <Button
        android:id="@+id/multiplyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/subtractButton"
        android:layout_centerHorizontal="true"
        android:text="Multiply" />

    <Button
        android:id="@+id/divideButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/multiplyButton"
        android:layout_centerHorizontal="true"
        android:text="Divide" />

    <TextView
        android:id="@+id/resultView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/divideButton"
        android:layout_centerHorizontal="true"
        android:text="Result"
        android:textSize="18sp"
        android:layout_marginTop="16dp" />
</RelativeLayout>
```

3. **Unit Conversion**

MainActivity.java

```
package com.example.unitconverter;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    private Spinner conversionTypeSpinner;
    private Spinner fromUnitSpinner;
    private Spinner toUnitSpinner;
    private EditText inputValue;
    private Button convertButton;
    private TextView resultTextView;

    private Map<String, String[]> unitMap;

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

        conversionTypeSpinner = findViewById(R.id.conversionTypeSpinner);
        fromUnitSpinner = findViewById(R.id.fromUnitSpinner);
        toUnitSpinner = findViewById(R.id.toUnitSpinner);
        inputValue = findViewById(R.id.inputValue);
        convertButton = findViewById(R.id.convertButton);
        resultTextView = findViewById(R.id.resultTextView);

        setupUnitMap();
        setupConversionTypeSpinner();
        setupConvertButton();
    }

    private void setupUnitMap() {
        unitMap = new HashMap<>();
        unitMap.put("Length", getResources().getStringArray(R.array.length_units));
        unitMap.put("Weight", getResources().getStringArray(R.array.weight_units));
                                                        unitMap.put("Temperature",
getResources().getStringArray(R.array.temperature_units));
    }

    private void setupConversionTypeSpinner() {
                                      ArrayAdapter<CharSequence>      adapter      =
ArrayAdapter.createFromResource(this,
            R.array.conversion_types, android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
        conversionTypeSpinner.setAdapter(adapter);

                                  conversionTypeSpinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
              public void onItemSelected(AdapterView<?> parent, View view, int position,
long id) {
                String selectedType = parent.getItemAtPosition(position).toString();
                updateUnitSpinners(selectedType);
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
            }
        });
    }

    private void updateUnitSpinners(String type) {
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
                android.R.layout.simple_spinner_item, unitMap.get(type));

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
        fromUnitSpinner.setAdapter(adapter);
        toUnitSpinner.setAdapter(adapter);
    }

    private void setupConvertButton() {
        convertButton.setOnClickListener(v -> {
            String fromUnit = fromUnitSpinner.getSelectedItem().toString();
            String toUnit = toUnitSpinner.getSelectedItem().toString();
            double value = Double.parseDouble(inputValue.getText().toString());

            double result = performConversion(fromUnit, toUnit, value);
            resultTextView.setText(String.valueOf(result));
        });
    }

    private double performConversion(String fromUnit, String toUnit, double value) {
        if (fromUnit.equals(toUnit)) {
            return value;
        }

        // Length Conversion
        if (fromUnit.equals("Meter")) {
            if (toUnit.equals("Kilometer")) return value / 1000;
            if (toUnit.equals("Centimeter")) return value * 100;
            if (toUnit.equals("Millimeter")) return value * 1000;
            if (toUnit.equals("Inch")) return value * 39.3701;
            if (toUnit.equals("Foot")) return value * 3.28084;
        }
        if (fromUnit.equals("Kilometer")) {
            if (toUnit.equals("Meter")) return value * 1000;
            if (toUnit.equals("Centimeter")) return value * 100000;
            if (toUnit.equals("Millimeter")) return value * 1000000;
            if (toUnit.equals("Inch")) return value * 39370.1;
            if (toUnit.equals("Foot")) return value * 3280.84;
        }
        if (fromUnit.equals("Centimeter")) {
            if (toUnit.equals("Meter")) return value / 100;
            if (toUnit.equals("Kilometer")) return value / 100000;
            if (toUnit.equals("Millimeter")) return value * 10;
            if (toUnit.equals("Inch")) return value * 0.393701;
            if (toUnit.equals("Foot")) return value * 0.0328084;
        }
        if (fromUnit.equals("Millimeter")) {
            if (toUnit.equals("Meter")) return value / 1000;
            if (toUnit.equals("Kilometer")) return value / 1000000;
            if (toUnit.equals("Centimeter")) return value / 10;
            if (toUnit.equals("Inch")) return value * 0.0393701;
            if (toUnit.equals("Foot")) return value * 0.00328084;
        }
        if (fromUnit.equals("Inch")) {
            if (toUnit.equals("Meter")) return value / 39.3701;
            if (toUnit.equals("Kilometer")) return value / 39370.1;
            if (toUnit.equals("Centimeter")) return value * 2.54;
            if (toUnit.equals("Millimeter")) return value * 25.4;
            if (toUnit.equals("Foot")) return value / 12;
        }
        if (fromUnit.equals("Foot")) {
            if (toUnit.equals("Meter")) return value / 3.28084;
            if (toUnit.equals("Kilometer")) return value / 3280.84;
            if (toUnit.equals("Centimeter")) return value * 30.48;
            if (toUnit.equals("Millimeter")) return value * 304.8;
            if (toUnit.equals("Inch")) return value * 12;
```

```
}

// Weight Conversion
if (fromUnit.equals("Gram")) {
    if (toUnit.equals("Kilogram")) return value / 1000;
    if (toUnit.equals("Pound")) return value * 0.00220462;
    if (toUnit.equals("Ounce")) return value * 0.035274;
}
if (fromUnit.equals("Kilogram")) {
    if (toUnit.equals("Gram")) return value * 1000;
    if (toUnit.equals("Pound")) return value * 2.20462;
    if (toUnit.equals("Ounce")) return value * 35.274;
}
if (fromUnit.equals("Pound")) {
    if (toUnit.equals("Gram")) return value * 453.592;
    if (toUnit.equals("Kilogram")) return value * 0.453592;
    if (toUnit.equals("Ounce")) return value * 16;
}
if (fromUnit.equals("Ounce")) {
    if (toUnit.equals("Gram")) return value * 28.3495;
    if (toUnit.equals("Kilogram")) return value * 0.0283495;
    if (toUnit.equals("Pound")) return value / 16;
}

// Temperature Conversion
if (fromUnit.equals("Celsius")) {
    if (toUnit.equals("Fahrenheit")) return value * 9/5 + 32;
    if (toUnit.equals("Kelvin")) return value + 273.15;
}
if (fromUnit.equals("Fahrenheit")) {
    if (toUnit.equals("Celsius")) return (value - 32) * 5/9;
    if (toUnit.equals("Kelvin")) return (value - 32) * 5/9 + 273.15;
}
if (fromUnit.equals("Kelvin")) {
    if (toUnit.equals("Celsius")) return value - 273.15;
    if (toUnit.equals("Fahrenheit")) return (value - 273.15) * 9/5 + 32;
```

```
        }

        return value; // Default case, should not happen
    }
}
```

activity_main.xml

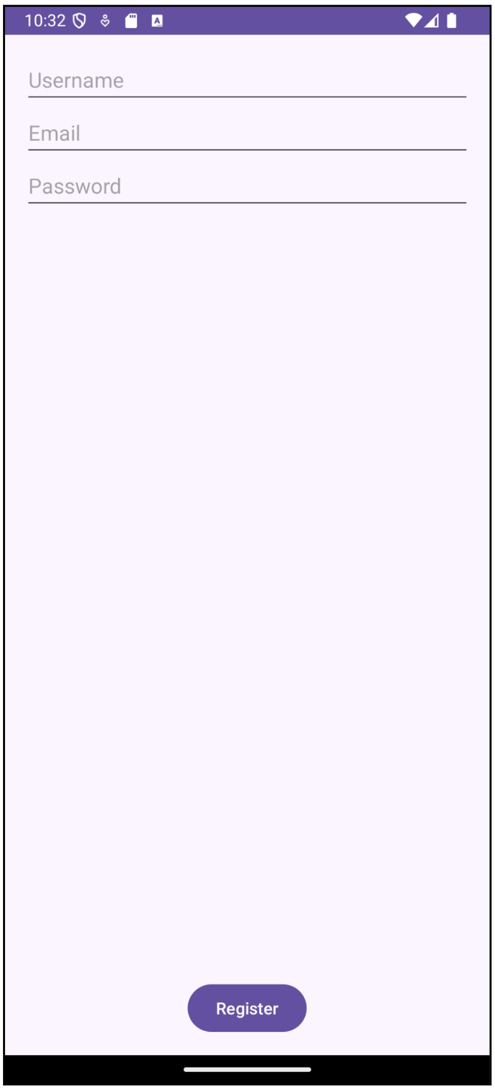
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <Spinner
        android:id="@+id/conversionTypeSpinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:entries="@array/conversion_types" />

    <Spinner
        android:id="@+id/fromUnitSpinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/conversionTypeSpinner"
        android:layout_marginTop="16dp" />

    <Spinner
        android:id="@+id/toUnitSpinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/fromUnitSpinner"
        android:layout_marginTop="16dp" />

    <EditText
        android:id="@+id/inputValue"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/toUnitSpinner"
        android:layout_marginTop="16dp"
        android:hint="Enter value"
        android:inputType="numberDecimal" />

    <Button
        android:id="@+id/convertButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Convert"
        android:layout_below="@id/inputValue"
        android:layout_marginTop="16dp"
        android:layout_alignParentEnd="true" />

    <TextView
        android:id="@+id/resultTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/convertButton"
        android:layout_marginTop="16dp"
        android:textSize="18sp"
        android:text="Result will be shown here" />
</RelativeLayout>
```

strings.xml

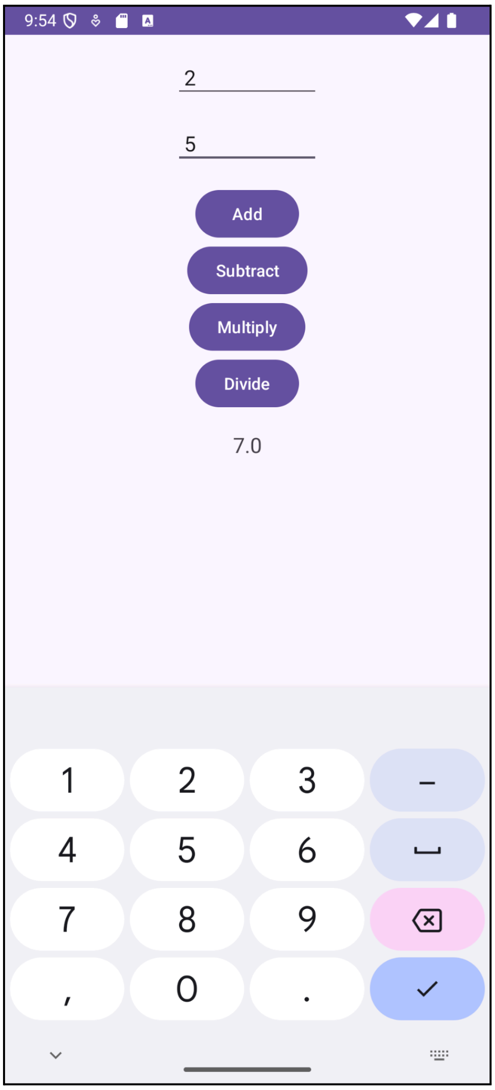
```
<resources>
    <string name="app_name">Unit Conversion App</string>
    <!-- Conversion Types -->
    <string-array name="conversion_types">
        <item>Length</item>
        <item>Weight</item>
        <item>Temperature</item>
    </string-array>

    <!-- Length Units -->
    <string-array name="length_units">
        <item>Meter</item>
        <item>Kilometer</item>
        <item>Centimeter</item>
        <item>Millimeter</item>
        <item>Inch</item>
        <item>Foot</item>
    </string-array>

    <!-- Weight Units -->
    <string-array name="weight_units">
        <item>Gram</item>
        <item>Kilogram</item>
        <item>Pound</item>
        <item>Ounce</item>
    </string-array>
    <!-- Temperature Units -->
    <string-array name="temperature_units">
        <item>Celsius</item>
        <item>Fahrenheit</item>
        <item>Kelvin</item>
    </string-array>
</resources>
```

Output:

1. **Registration Form**



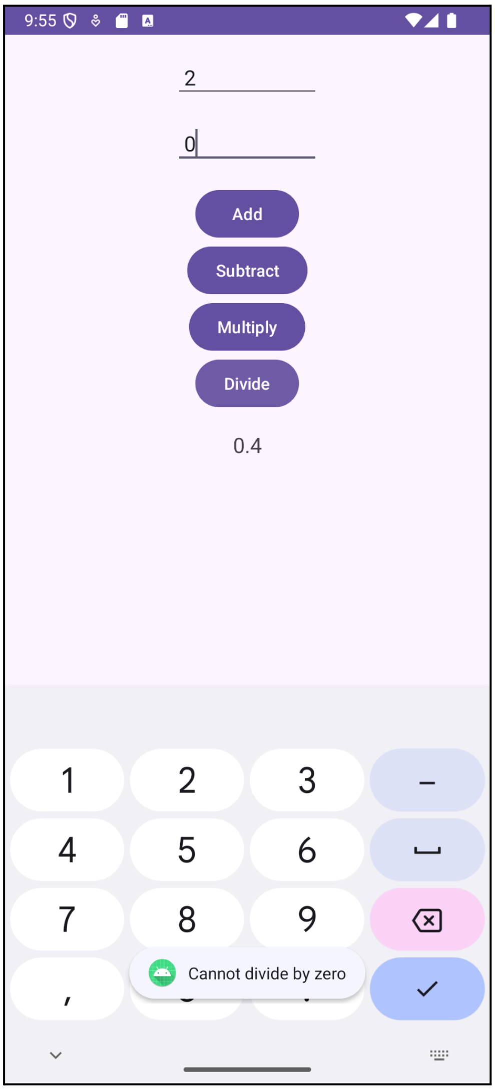
2. Calculator



2

5

7.0



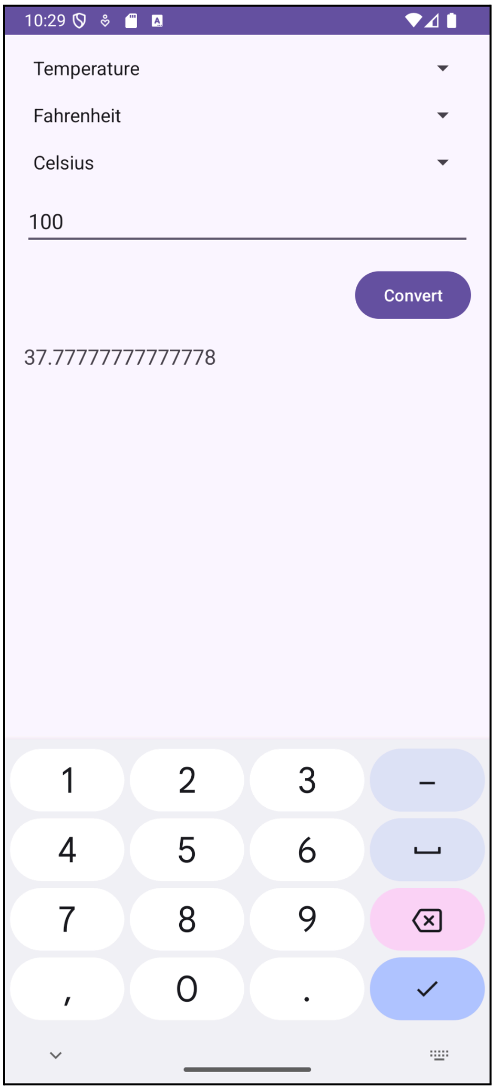
2

0

0.4

Cannot divide by zero

3. Unit Conversion



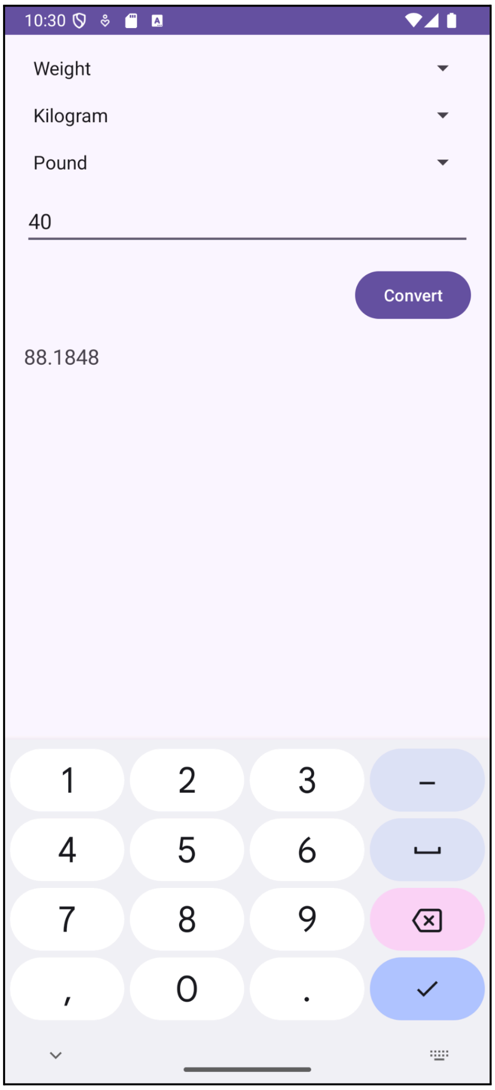
Temperature

Fahrenheit

Celsius

100

37.77777777777778



Weight

Kilogram

Pound

40

88.1848

Conclusion:

Through these practical exercises, we developed essential skills in Android application development, including creating a simple calculator with arithmetic operations and error handling, designing a registration form with various layout managers, and implementing navigation between different layouts. We also built a unit conversion app with comprehensive logic for length, weight, and temperature conversions.