

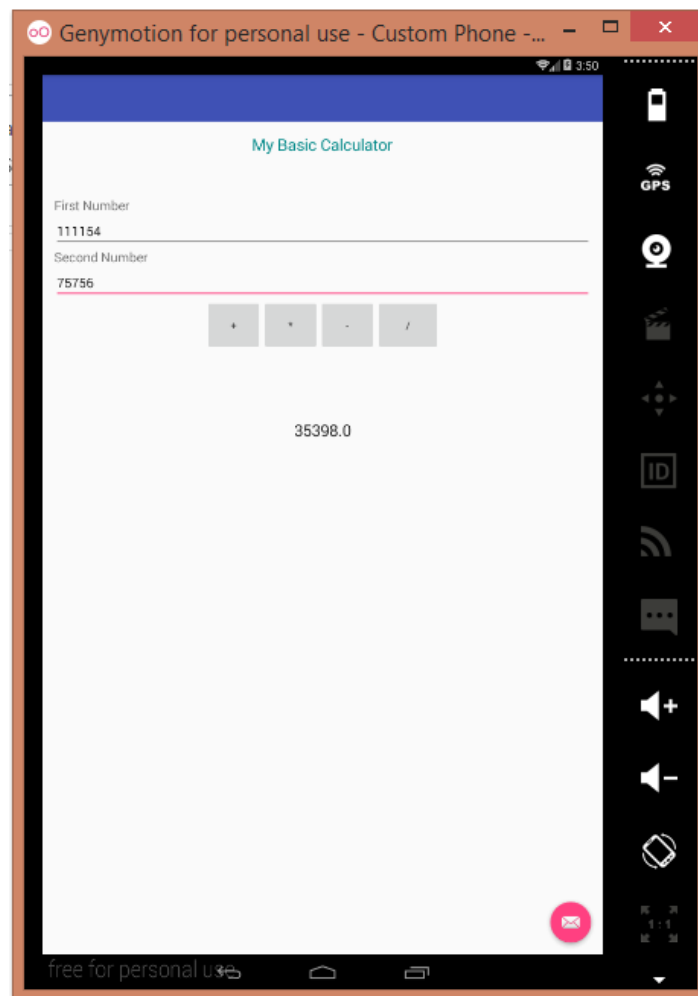
**Android App Development**

# **PRACTICAL 1**

**Part A**

**AIM:** Android Layouts (Linear & Relative Layout) and Signal Handler.

**Scenario:** Create a Calculator using relative layout and Linear Layout for the buttons, which would accept two numbers and onClick of any of the four Buttons (+,-,\*,/) would display the answer in a TextField. The onClick Listener has to be registered to the ActivityMain class.



**Part B (to be completed by students)**

# SVKM's NMIMS University

## Mukesh Patel School of Technology Management & Engineering

Mobile Application Development, B.Tech/MBA Tech IT/Cyber security Sem VI/IV

### Android App Development

(Students must submit the soft copy as per the following segments. The soft copy must be uploaded on the Blackboard. The filename should be Batch\_RollNo\_Exp\_No)

Roll No.: K036	Name: Arjun Mehta
Prog/Yr/Sem: B. Tech CSE Cybersecurity Sem3	Batch: K1
Date of Experiment: 15-01-2025	Date of Submission: 15-01-2025

1. **Program Scenario and Program code:** (Write Scenario and Paste your program code (Java, xml resource and layout)).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="@android:color/white">

    <TextView
        android:id="@+id/header_lab"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="K036 Mobile Application Development LAB 2"
        android:textSize="24sp"
        android:textStyle="bold"
        android:gravity="center"
        android:layout_marginBottom="8dp" />

    <!-- Header for Simple Calculator -->
    <TextView
        android:id="@+id/header_calculator"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simple Calculator"
        android:textSize="20sp"
        android:textStyle="bold"
        android:gravity="center"
        android:layout_marginBottom="16dp" />

    <!-- Input for First Number -->
    <EditText
        android:id="@+id/ed_num1"
```

**Android App Development**

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter first number"
        android:inputType="numberDecimal"
        android:layout_marginBottom="16dp" />
<!-- Input for Second Number -->
<EditText
    android:id="@+id/ed_num2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter second number"
    android:inputType="numberDecimal"
    android:layout_marginBottom="16dp" />
<!-- Buttons for Operations -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_marginBottom="16dp">
    <Button
        android:id="@+id/btn_add"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="+" />
    <Button
        android:id="@+id/btn_subtract"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-" />
    <Button
        android:id="@+id/btn_multiply"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="*" />
    <Button
        android:id="@+id/btn_divide"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="/" />
</LinearLayout>
<!-- Output Label -->
<TextView
```

Android App Development

```
        android:id="@+id/lbout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Result"
        android:textSize="18sp"
        android:gravity="center"
        android:layout_marginTop="16dp" />
</LinearLayout>
```

```
package com.example.k036_lab2_mad;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    // Declare the views
    EditText ed_num1, ed_num2;
    TextView lbout;
    Button btnadd, btnsubtract, btnmultiply, btndivide;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Initialize the views
        ed_num1 = findViewById(R.id.ed_num1);
        ed_num2 = findViewById(R.id.ed_num2);
        lbout = findViewById(R.id.lbout);
        btnadd = findViewById(R.id.btn_add);
        btnsubtract = findViewById(R.id.btn_subtract);
        btnmultiply = findViewById(R.id.btn_multiply);
        btndivide = findViewById(R.id.btn_divide);
        // Set onClickListeners for buttons
        btnadd.setOnClickListener(this);
        btnsubtract.setOnClickListener(this);
        btnmultiply.setOnClickListener(this);
        btndivide.setOnClickListener(this);
    }
    @Override
    public void onClick(View view) {
        String num1Str = ed_num1.getText().toString();
        String num2Str = ed_num2.getText().toString();
        // Validate inputs
```

Android App Development

```
        if (num1Str.isEmpty() || num2Str.isEmpty()) {
            Toast.makeText(this, "Input boxes are empty",
Toast.LENGTH_SHORT).show();
            return;
        }
        double num1, num2;
        try {
            num1 = Double.parseDouble(num1Str);
            num2 = Double.parseDouble(num2Str);
        } catch (NumberFormatException e) {
            Toast.makeText(this, "Please enter valid numbers",
Toast.LENGTH_SHORT).show();
            return;
        }
        double result = 0;
        // Use if-else instead of switch-case
        if (view.getId() == R.id.btn_add) {
            result = num1 + num2;
        } else if (view.getId() == R.id.btn_subtract) {
            result = num1 - num2;
        } else if (view.getId() == R.id.btn_multiply) {
            result = num1 * num2;
        } else if (view.getId() == R.id.btn_divide) {
            if (num2 == 0) {
                Toast.makeText(this, "Cannot divide by zero",
Toast.LENGTH_SHORT).show();
                return;
            }
            result = num1 / num2;
        }
        // Display the result
        lbout.setText(String.valueOf(result));
    }
}
```

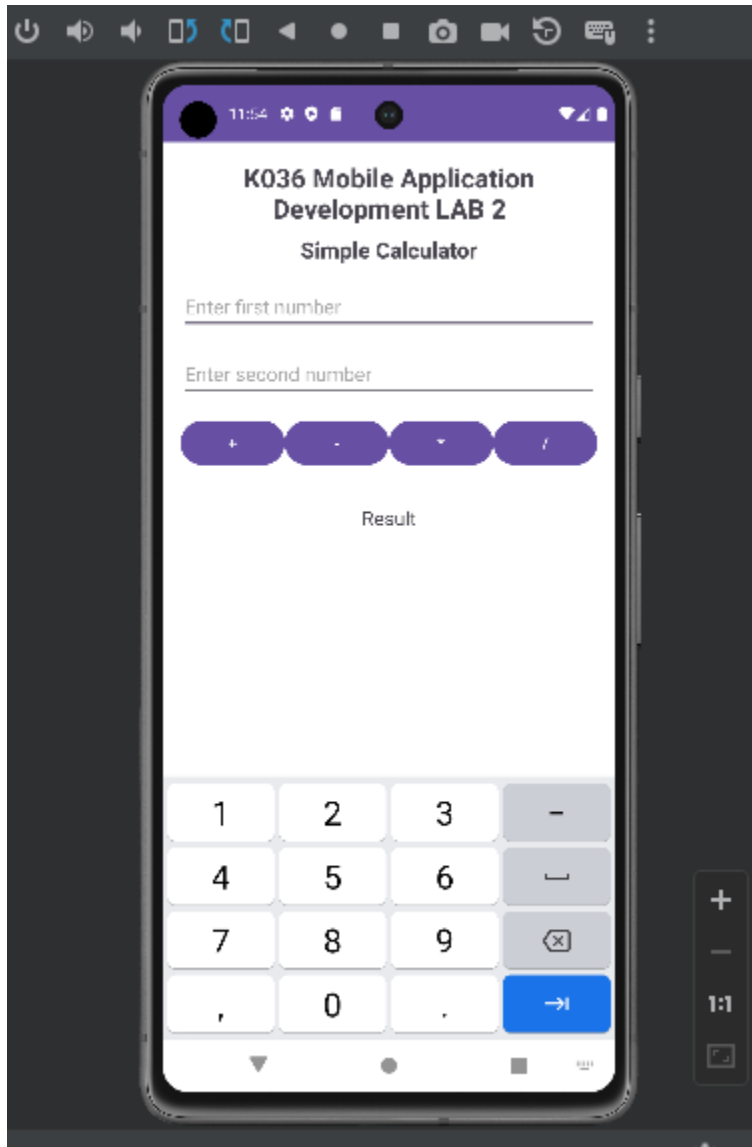
2. **Output:** (Paste your program input and output screen shots).

# SVKM's NMIMS University

## Mukesh Patel School of Technology Management & Engineering

Mobile Application Development, B.Tech/MBA Tech IT/Cyber security Sem VI/IV

### Android App Development

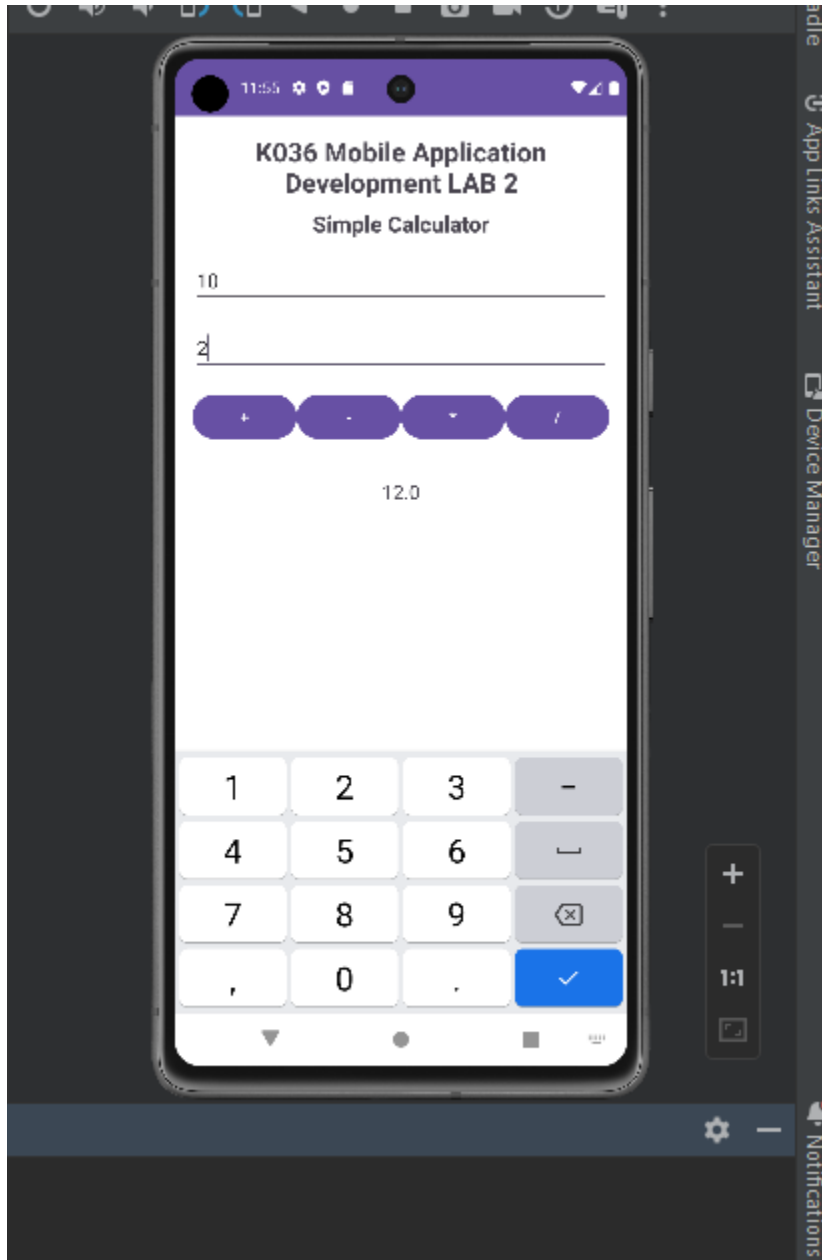


**SVKM's NMIMS University**

**Mukesh Patel School of Technology Management & Engineering**

Mobile Application Development, B.Tech/MBA Tech IT/Cyber security Sem VI/IV

**Android App Development**

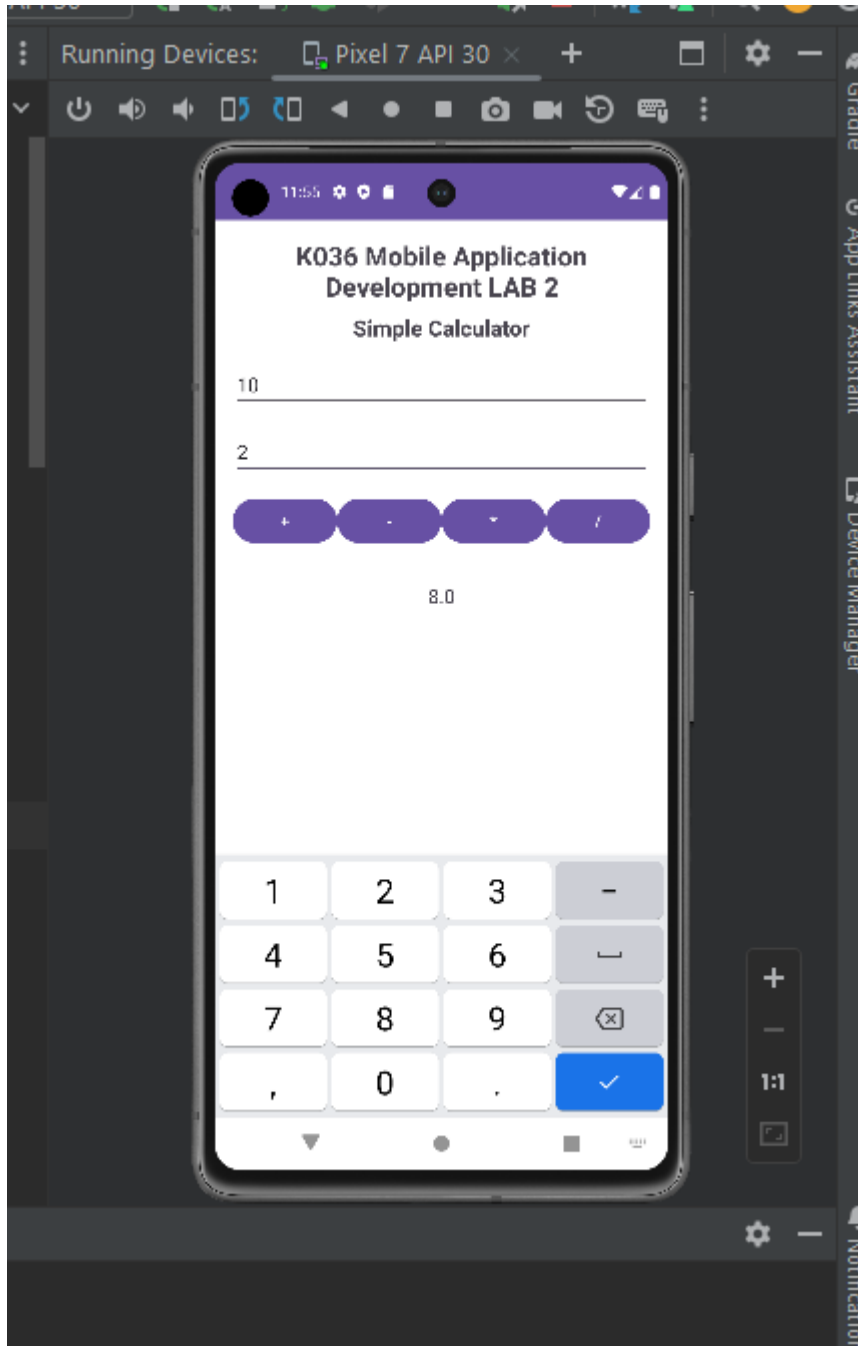


**SVKM's NMIMS University**

**Mukesh Patel School of Technology Management & Engineering**

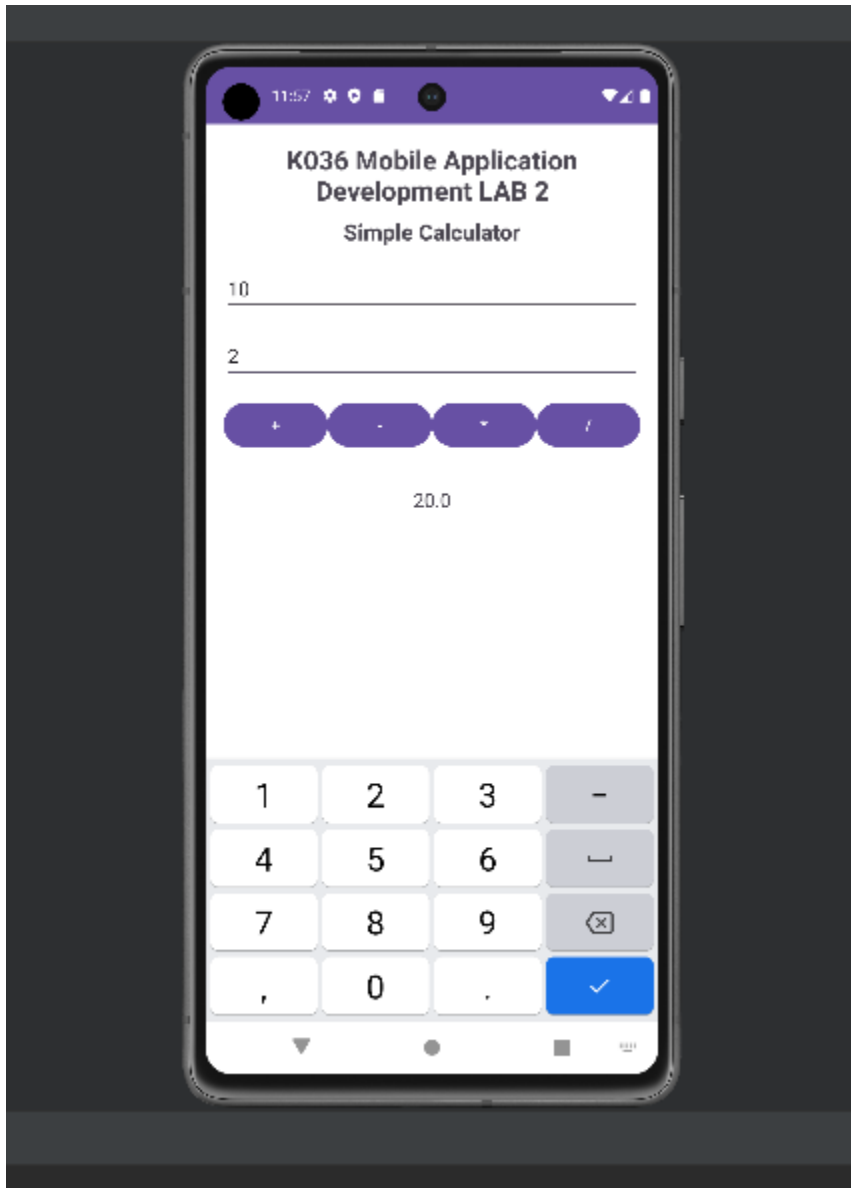
Mobile Application Development, B.Tech/MBA Tech IT/Cyber security Sem VI/IV

**Android App Development**





Android App Development



- 3. Observations:** A brief description of the design aspects and working of the code in your own words.

The provided code implements a simple calculator application in Android using Java. The layout and functionality are as follows:

**Android App Development**

- **Layout:** The user interface is designed using a `LinearLayout` with a vertical orientation. It consists of:
  - Two `EditText` fields for inputting numbers.
  - Four buttons for the arithmetic operations: addition, subtraction, multiplication, and division.
  - A `TextView` that displays the result of the calculation.
  - The design also includes two header `TextView` elements displaying the title "K036 Mobile Application Development LAB 2" and "Simple Calculator."
- **Functionality:**
  - The app captures user input from two `EditText` fields and performs arithmetic operations based on the button clicked.
  - The `onClick()` method handles the logic for performing calculations. If any of the input fields are empty or contain invalid data, the app displays a toast message to prompt the user to correct the input.
  - The result of the calculation is displayed in the `TextView (lbout)`.
  - Basic error handling is implemented for division by zero.

**4. Questions:** Draw & Explain with respect to layouts the Scene Graph of the experiment.

A **Scene Graph** represents the hierarchical structure of the UI components in an application. For this experiment, the layout is based on a `LinearLayout` with a vertical orientation, which means the UI components are arranged in a linear, top-to-bottom fashion. Here's a breakdown of the scene graph:

1. **Root: LinearLayout (Vertical)**
  - This is the root container that holds all other components.
2. **Child 1: TextView (Header: "K036 Mobile Application Development LAB 2")**
  - Displays the title of the experiment at the top of the screen.
3. **Child 2: TextView (Header: "Simple Calculator")**
  - Displays the title of the calculator below the first header.
4. **Child 3: EditText (Input Field for First Number)**
  - A text input field where the user can enter the first number.
5. **Child 4: EditText (Input Field for Second Number)**
  - A text input field where the user can enter the second number.
6. **Child 5: LinearLayout (Horizontal)**
  - This container holds the four buttons for the operations (add, subtract, multiply, divide).

**Android App Development**

- It arranges the buttons horizontally with equal width using `layout_weight`.
  - **Child of LinearLayout:**
    - Button (Add)
    - Button (Subtract)
    - Button (Multiply)
    - Button (Divide)
7. **Child 6: TextView (Output Label: "Result")**
- Displays the result of the calculation.
5. **Conclusion (Learning Outcomes):** How were the outcomes defined for the experiment in Part A fulfilled through the scenarios?

The experiment fulfilled the following learning outcomes:

- **Understanding Layouts:** The experiment demonstrated the use of different types of layouts (`LinearLayout`, `TextView`, `EditText`, and `Button`) to design a user interface for a simple calculator app.
- **Event Handling:** It showed how to use `OnClickListener` to handle button clicks and perform arithmetic operations based on the user's input.
- **Input Validation:** The app implements input validation to ensure the user enters valid numbers and handles edge cases such as division by zero.
- **UI Design Principles:** The experiment emphasized organizing the UI components effectively within the layout to create a user-friendly interface.