Web Technology and Mobile Applications

Experiment- 09

Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, multiplication, and Division.

AIM:

To develop an Android application using Button, TextView, and EditText controls to design a simple calculator that performs basic arithmetic operations such as Addition, Subtraction, Multiplication, and Division.

Algorithm:

Step 1: Start the Android Studio Project

- Create a new project with an Empty Activity.
- Set the project language as Kotlin.

Step 2: Design the User Interface (UI)

- Open the activity_main.xml file.
- Add the following UI elements:
    - A TextView to display the result or operations.
    - Two EditText fields for user input (first number and second number).
    - Four Buttons for operations: Add, Subtract, Multiply, Divide.

Step 3: Initialize UI Components in Kotlin

- In MainActivity.kt, use findViewById() or ViewBinding to link Kotlin variables to XML elements.

Step 4: Add Click Listeners to Buttons

- For each button:
    - Read input values from the EditTexts.
    - Convert them to numeric types (e.g., toDouble() or toFloat()).
    - Perform the selected arithmetic operation.
    - Display the result in the TextView.

Step 5: Handle Invalid Input

- Check if both inputs are not empty.
- Check if the divisor is not zero in case of division.
- Use try-catch to handle input format exceptions.

Step 6: Test the Application

- Run the app on the emulator or a physical device.
- Enter values and press the operation buttons.
- Verify correct results are shown and errors are handled gracefully.

Code:

MainActivity.kt:

```
package com.example.myapplication_1

import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var display: TextView
    private lateinit var expressionDisplay: TextView
    private var currentInput: String = ""
    private var operator: String = ""
    private var firstNumber: String = ""
    private var isOperatorPressed = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        display = findViewById(R.id.display)
```

```
        expressionDisplay = findViewById(R.id.expressionDisplay)

        val numberButtons = listOf(
            R.id.btn0, R.id.btn1, R.id.btn2, R.id.btn3,
            R.id.btn4, R.id.btn5, R.id.btn6, R.id.btn7,
            R.id.btn8, R.id.btn9
        )

        for (id in numberButtons) {
            findViewById<Button>(id).setOnClickListener {
                val number = (it as Button).text.toString()
                currentInput += number
                display.text = currentInput
            }
        }

        findViewById<Button>(R.id.btnAdd).setOnClickListener {
handleOperator("+") }
        findViewById<Button>(R.id.btnSubtract).setOnClickListener {
handleOperator("-") }
        findViewById<Button>(R.id.btnMultiply).setOnClickListener {
handleOperator("*") }
        findViewById<Button>(R.id.btnDivide).setOnClickListener {
handleOperator("/") }

        findViewById<Button>(R.id.btnClear).setOnClickListener {
            clearAll()
        }

        findViewById<Button>(R.id.btnEquals).setOnClickListener {
            if (currentInput.isNotEmpty() && firstNumber.isNotEmpty() &&
operator.isNotEmpty()) {
                val result = calculate(firstNumber.toDouble(),
currentInput.toDouble(), operator)
                expressionDisplay.text = "$firstNumber $operator
$currentInput"
                display.text = result.toString()
                currentInput = result.toString()
                firstNumber = ""
                operator = ""
                isOperatorPressed = false
            }
        }
    }

    private fun handleOperator(op: String) {
        if (currentInput.isNotEmpty()) {
            if (!isOperatorPressed) {
                firstNumber = currentInput
                operator = op
                expressionDisplay.text = "$firstNumber $operator"
                currentInput = ""
                isOperatorPressed = true
            } else {
                // Change operator mid-way
                operator = op
                expressionDisplay.text = "$firstNumber $operator"
            }
        }
    }
```

```
    private fun calculate(num1: Double, num2: Double, op: String): Double {
        return when (op) {
            "+" -> num1 + num2
            "-" -> num1 - num2
            "*" -> num1 * num2
            "/" -> if (num2 != 0.0) num1 / num2 else Double.NaN
            else -> 0.0
        }
    }

    private fun clearAll() {
        currentInput = ""
        firstNumber = ""
        operator = ""
        display.text = "0"
        expressionDisplay.text = ""
        isOperatorPressed = false
    }
}
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="#F9F9F9">

    <!-- Expression Display -->
    <TextView
        android:id="@+id/expressionDisplay"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="20sp"
        android:textColor="#888888"
        android:gravity="end"
        android:padding="8dp" />

    <!-- Result Display -->
    <TextView
        android:id="@+id/display"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:text="0"
        android:textSize="36sp"
        android:textColor="#000000"
        android:gravity="end"
        android:padding="16dp"
        android:background="#EEEEEE" />

    <!-- Add some space -->
    <View
        android:layout_width="match_parent"
        android:layout_height="16dp" />

    <!-- Each row of buttons with margins -->
```

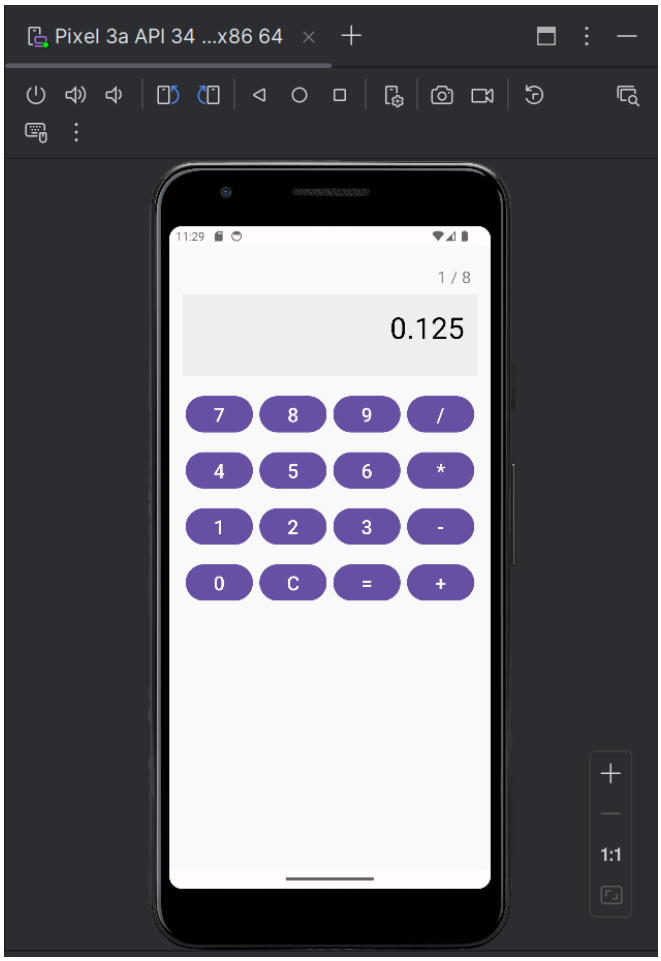
```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="8dp">

        <Button android:id="@+id/btn7" style="@style/CalcButton"
android:text="7" />
        <Button android:id="@+id/btn8" style="@style/CalcButton"
android:text="8" />
        <Button android:id="@+id/btn9" style="@style/CalcButton"
android:text="9" />
        <Button android:id="@+id/btnDivide" style="@style/CalcButton"
android:text="/" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="8dp">

        <Button android:id="@+id/btn4" style="@style/CalcButton"
android:text="4" />
        <Button android:id="@+id/btn5" style="@style/CalcButton"
android:text="5" />
        <Button android:id="@+id/btn6" style="@style/CalcButton"
android:text="6" />
        <Button android:id="@+id/btnMultiply" style="@style/CalcButton"
android:text="*" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="8dp">

        <Button android:id="@+id/btn1" style="@style/CalcButton"
android:text="1" />
        <Button android:id="@+id/btn2" style="@style/CalcButton"
android:text="2" />
        <Button android:id="@+id/btn3" style="@style/CalcButton"
android:text="3" />
        <Button android:id="@+id/btnSubtract" style="@style/CalcButton"
android:text="-" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="8dp">

        <Button android:id="@+id/btn0" style="@style/CalcButton"
android:text="0" />
        <Button android:id="@+id/btnClear" style="@style/CalcButton"
android:text="C" />
        <Button android:id="@+id/btnEquals" style="@style/CalcButton"
android:text="=" />
        <Button android:id="@+id/btnAdd" style="@style/CalcButton"
```

```
android:text="+" />
    </LinearLayout>
</LinearLayout>
```

Styles.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CalcButton">
        <item name="android:layout_width">0dp</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_weight">1</item>
        <item name="android:layout_margin">4dp</item>
        <item name="android:textSize">24sp</item>
    </style>
</resources>
```

Output:



Result:

Thus, a mobile application is created to do the basic functionalities of a calculator using Kotlin.