EX 4	Implementation of a Four-Function Calculator

Aim

To implement a four-function calculator that performs addition, subtraction, multiplication, and division by using platform-specific UI controls, and validate user input.

Definitions

Calculator

A simple calculator is a program that can perform addition, subtraction, multiplication, and division of two numbers provided as input.

User Input Validation

User input validation in Android is a crucial process for ensuring data integrity, improving user experience, and preventing security vulnerabilities like injection attacks. Validation should be performed on both the client-side for immediate user feedback and the server-side as a final security measure.

Platform-Specific UI Controls

In Android development, **platform-specific UI controls**, known as **views** or **widgets**, are the fundamental building blocks provided by the Android framework to create the user interface of an application. These controls adhere to Google's **Material Design** guidelines for a consistent and intuitive user experience.

Procedure

1. Open Android Studio IDE → go to File → New → New Project → specify the application name "Calculator" and company domain "com.mad.cal" → click "next" → choose Minimum SDK "API 17:Android 4.2(Jelly Bean)" → click "Next" → choose "Blank Activity" → click "next" → specify the Activity Name "MainActivity" → click "Finish".
2. Open MainActivity.java under app/java/ cal.mad.com.calculator and type the following codes:

MainActivity.java

```
package cal.mad.com.calculator;

import android.app.Activity;
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 Activity {

   EditText etNum1, etNum2;
   TextView tvResult;

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

      etNum1 = (EditText) findViewById(R.id.etNum1);
      etNum2 = (EditText) findViewById(R.id.etNum2);
      tvResult = (TextView) findViewById(R.id.tvResult);

      Button btnAdd = (Button) findViewById(R.id.btnAdd);
      Button btnSub = (Button) findViewById(R.id.btnSub);
      Button btnMul = (Button) findViewById(R.id.btnMul);
      Button btnDiv = (Button) findViewById(R.id.btnDiv);

      btnAdd.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            calculate('+');
         }
      });

      btnSub.setOnClickListener(new View.OnClickListener() {
         @Override
```

```
        public void onClick(View v) {
            calculate('-');
        }
    });

    btnMul.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('*');
        }
    });

    btnDiv.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('/');
        }
    });
}

private void calculate(char operator) {
    String s1 = etNum1.getText().toString().trim();
    String s2 = etNum2.getText().toString().trim();

    // Empty input validation
    if (s1.length() == 0 || s2.length() == 0) {
        showToast("Please enter both numbers");
        return;
    }

    double num1, num2;

    // Numeric validation
    try {
        num1 = Double.parseDouble(s1);
        num2 = Double.parseDouble(s2);
    } catch (NumberFormatException e) {
        showToast("Invalid number format");
        return;
    }

    double result;

    switch (operator) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
```

```
            case '*':
                result = num1 * num2;
                break;
            case '/':
                if (num2 == 0) {
                    showToast("Division by zero not allowed");
                    return;
                }
                result = num1 / num2;
                break;
            default:
                return;
        }

        tvResult.setText("Result: " + result);
    }

    private void showToast(String msg) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
    }
}
```

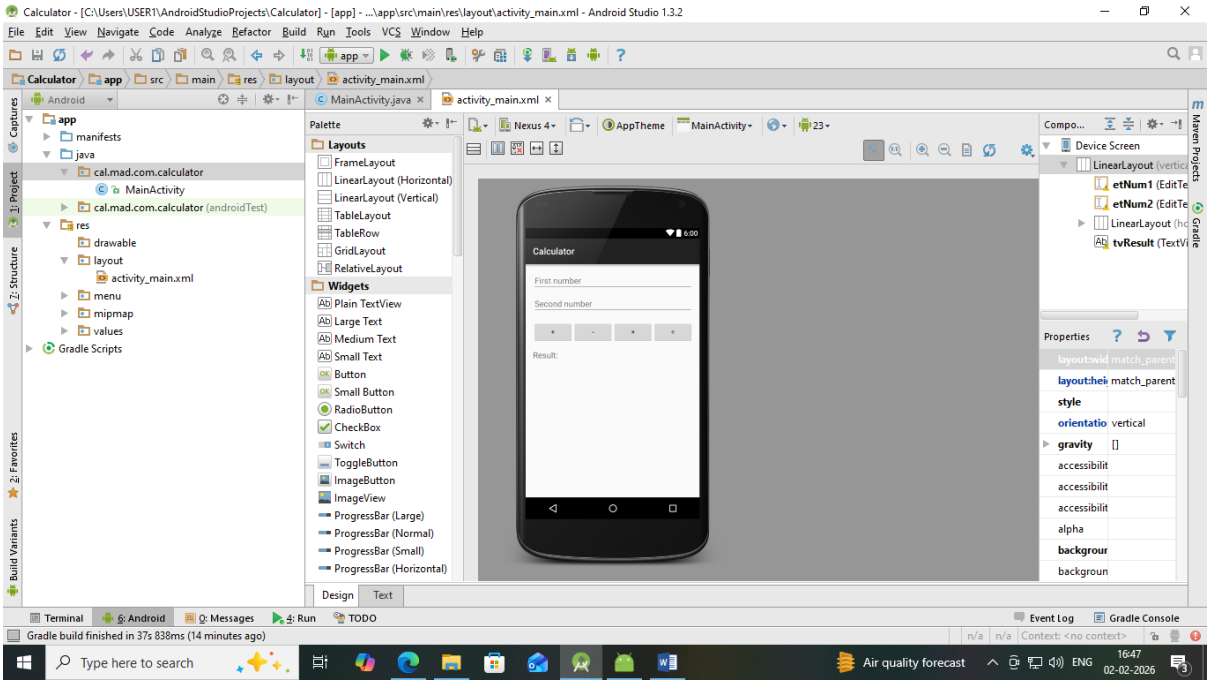
3. Open activity_main.xml under app/res/layout and type the following codes:

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">

    <EditText
        android:id="@+id/etNum1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="First number"
        android:inputType="numberDecimal" />

    <EditText
        android:id="@+id/etNum2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Second number"
        android:inputType="numberDecimal"
        android:layout_marginTop="8dp" />

    <LinearLayout
```

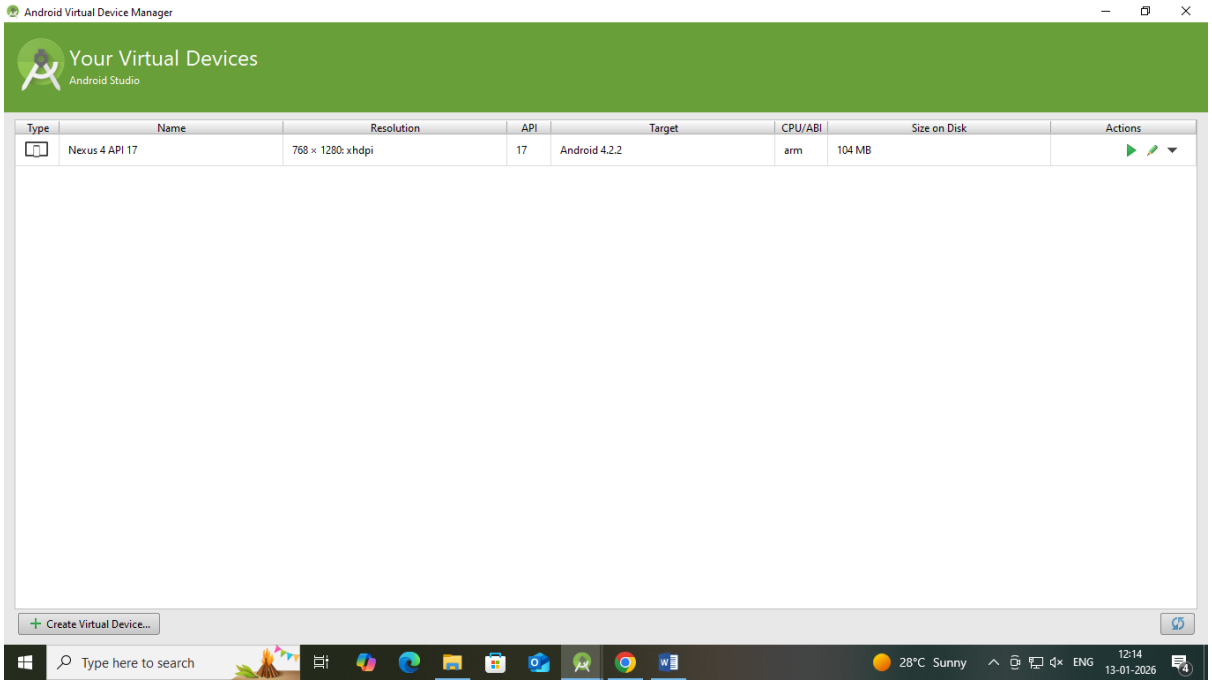
```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="16dp">

        <Button
            android:id="@+id/btnAdd"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="+" />

        <Button
            android:id="@+id/btnSub"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="-" />

        <Button
            android:id="@+id/btnMul"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="×" />

        <Button
            android:id="@+id/btnDiv"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="÷" />
    </LinearLayout>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Result:"
        android:textSize="18sp"
        android:layout_marginTop="16dp" />

</LinearLayout>
```

4. The design of the application will be as follows:



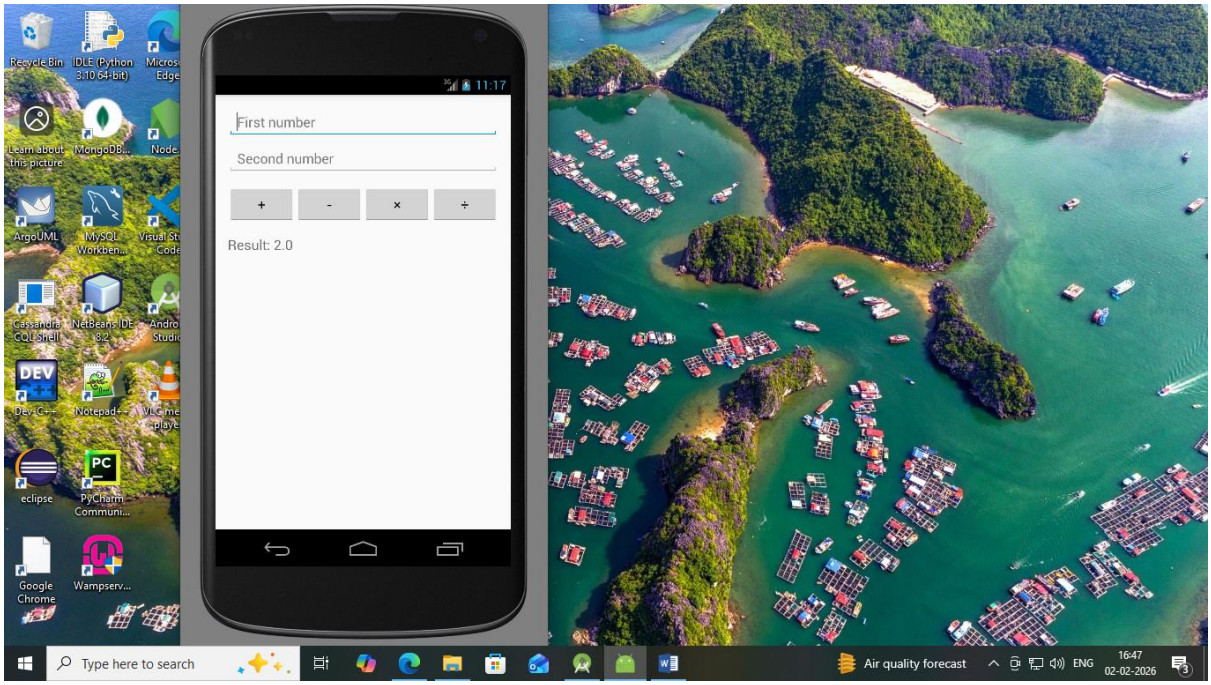
5. Go to Tools → android → AVD Manager → click "+ create a virtual device" → select "phone" from category → select "Nexus 4" from the list → click "next" → select Release name: Jelly Bean, API Level: 17, ABI: armeabi-v7a, Target: Android 4.2.2 from the list → click "next" → Choose orientation "portrait" → click "finish".

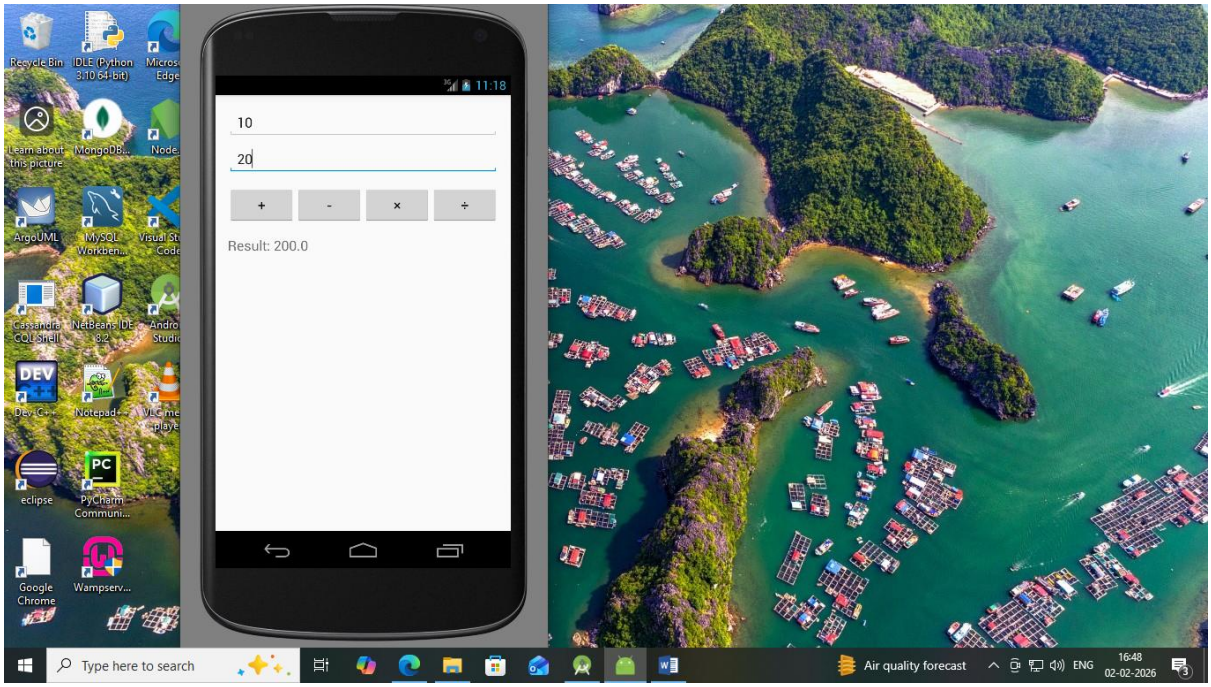
The following window will appear after configuring AVD:

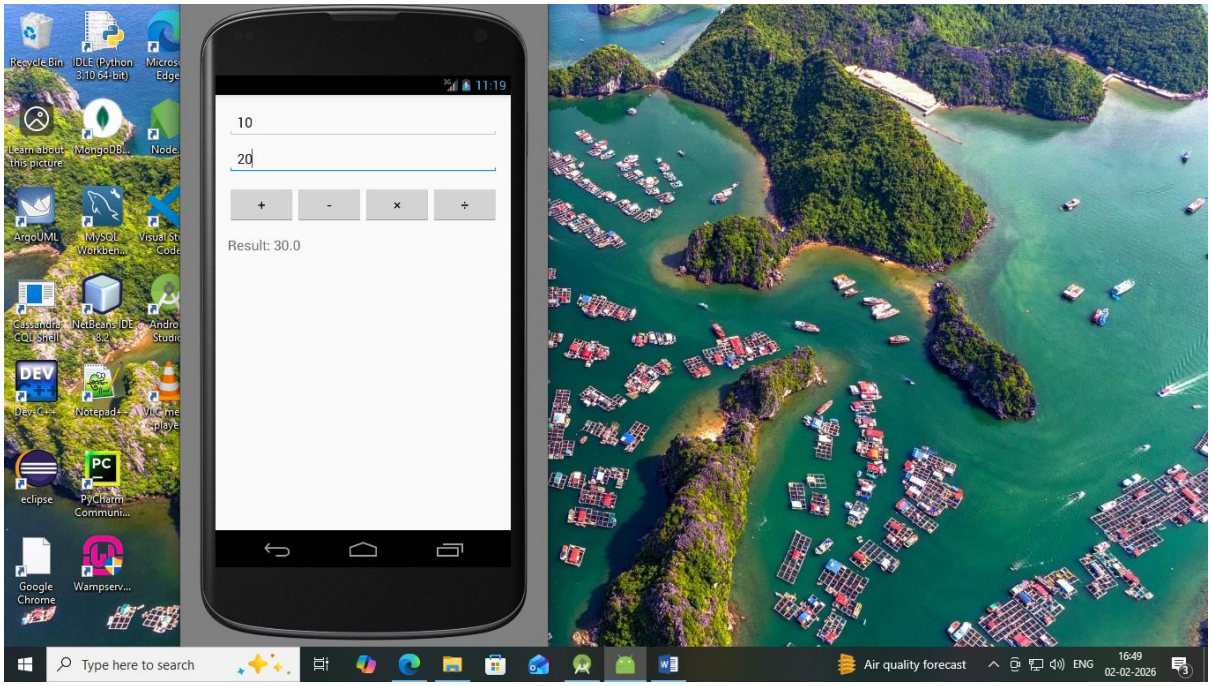


6. Click “Run app” button in the Android Studio → choose android virtual device → click “ok”.

Output






Recycle Bin
IDLE (Python 3.10 64-bit)
Learn about this picture
MongoDB...
ArgoUML
MySQL Workben...
Cassandra CQL Shell
NetBeans IDE 8.2
DEV
Dev-C++
Notepad++
eclipse
PyCharm Communi...
Google Chrome
Wampserv...
11:19
10
20
+
-
×
÷
Result: 30.0
Type here to search
Air quality forecast
ENG
16:49
02-02-2026

Result

Thus, a four-function calculator that performs addition, subtraction, multiplication, and division has been implemented in android.