

Practical No: 01

Aim: Create an Android application to demonstrate the Android Activity Lifecycle.

Code:

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.activitylifecycle;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }
}
```

```
}

@Override
protected void onStart() // Corrected: removed the duplicate 'void'
{
    super.onStart();
    Log.d("lifecycle","onStart invoked");
}

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

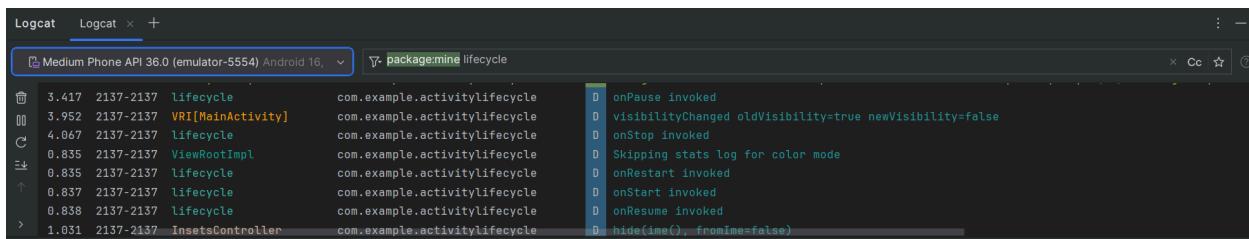
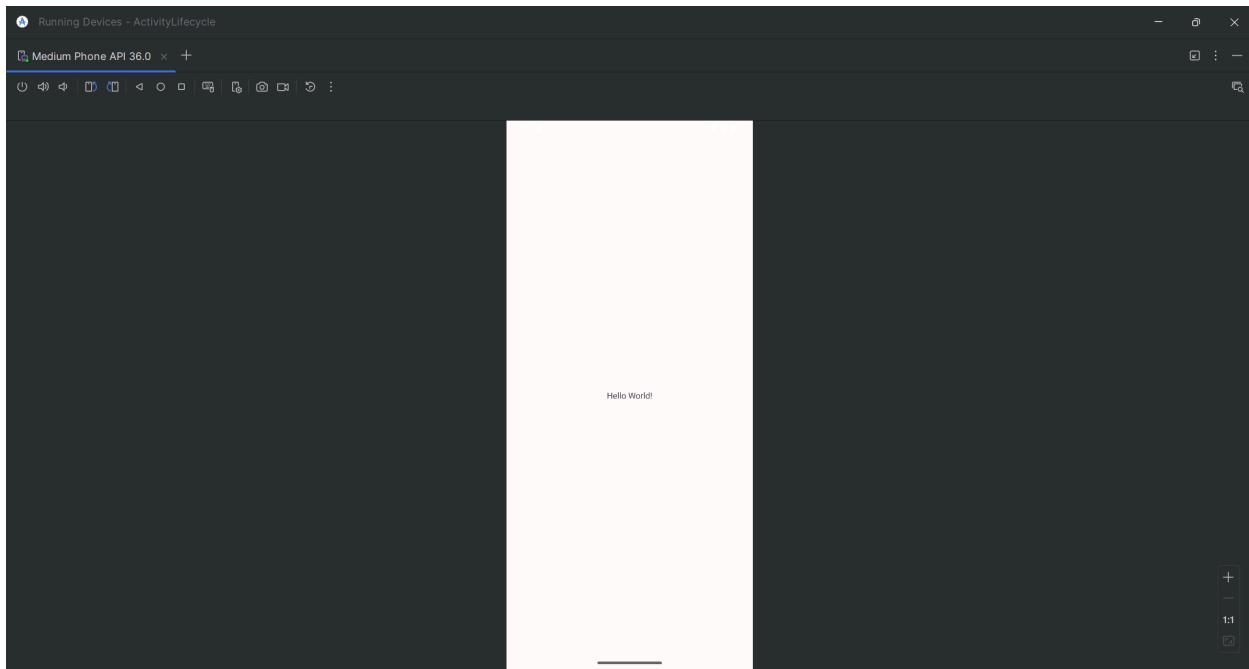
@Override
protected void onPause()
{
    super.onPause();
    Log.d("lifecycle","onPause invoked");
}

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

@Override
protected void onRestart()
{
    super.onRestart();
    Log.d("lifecycle","onRestart invoked");
}

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

Output:



Practical No: 02

Aim: Basic Layout (Text Display)

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter Text"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintVertical_bias="0.3"
        android:autofillHints="name" /> <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintVertical_bias="0.5" />

    <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Result will appear here"
        app:layout_constraintTop_toBottomOf="@+id/button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintVertical_bias="0.5" />

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

MainActivity.java

```
package com.example.textdisplay;

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; // Added for an optional enhancement

public class MainActivity extends AppCompatActivity {

    private Button btn1;
    private EditText ed1;
    private TextView txt1;

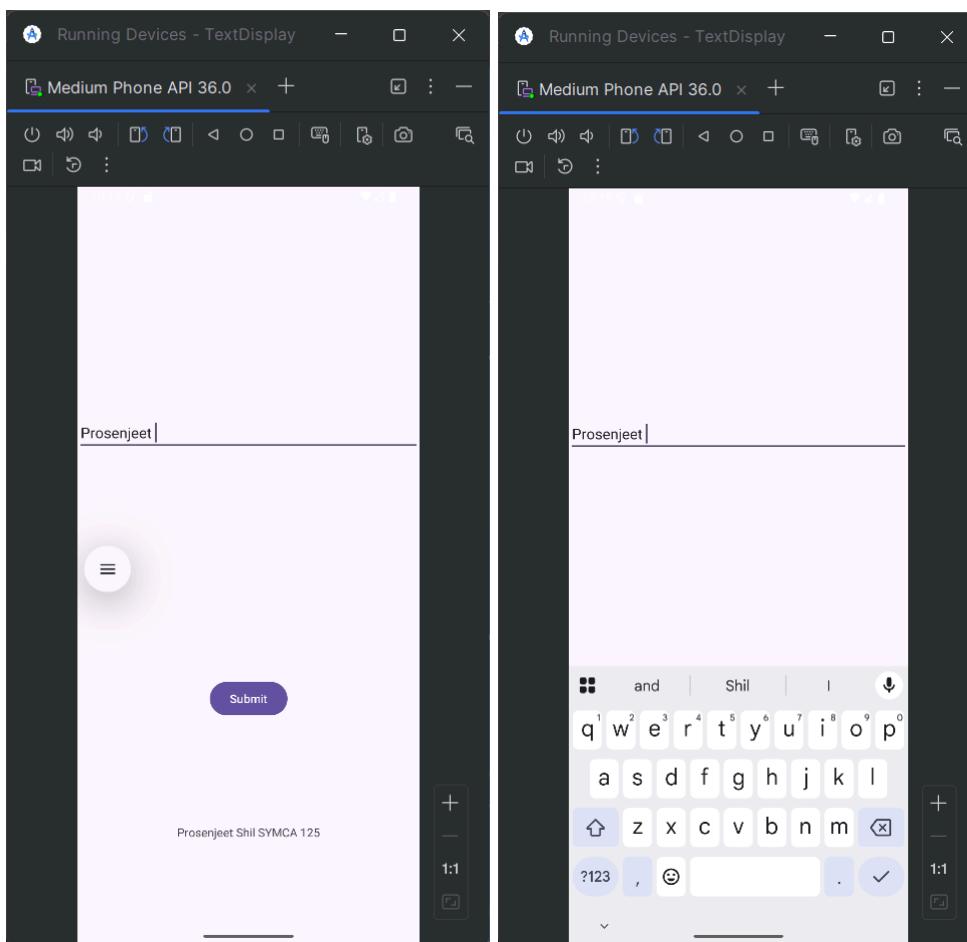
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize the views by finding them by their IDs from activity_main.xml
        btn1 = findViewById(R.id.button); // The ID of your button is "button"
        ed1 = findViewById(R.id.editTextTextPersonName); // The ID of your EditText is
        "editTextTextPersonName"
        txt1 = findViewById(R.id.textview1); // The ID of your TextView is "textview1"

        // Set up the button click listener
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Get the text from EditText and convert it to a String
                String msg = ed1.getText().toString();
```

```
// Check if the EditText is not empty
if (!msg.isEmpty()) {
    txt1.setText(msg); // Set the TextView's text to the entered message
    // Optional: Clear the EditText after setting the text
    ed1.setText("");
} else {
    txt1.setText("Please enter text"); // Display a message if EditText is empty
    // Optional: Show a short Toast message as well
    Toast.makeText(MainActivity.this, "Input field is empty!",
Toast.LENGTH_SHORT).show();
}
});
```

Output:



Practical No: 03

Aim: Toast

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/Div"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Toast"
        android:onClick="Toaster"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

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

MainActivity.java

```
package com.example.toast;

package com.example.toast;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View; // Required for the View parameter in the Toaster method
import android.widget.Toast; // Required for using the Toast class

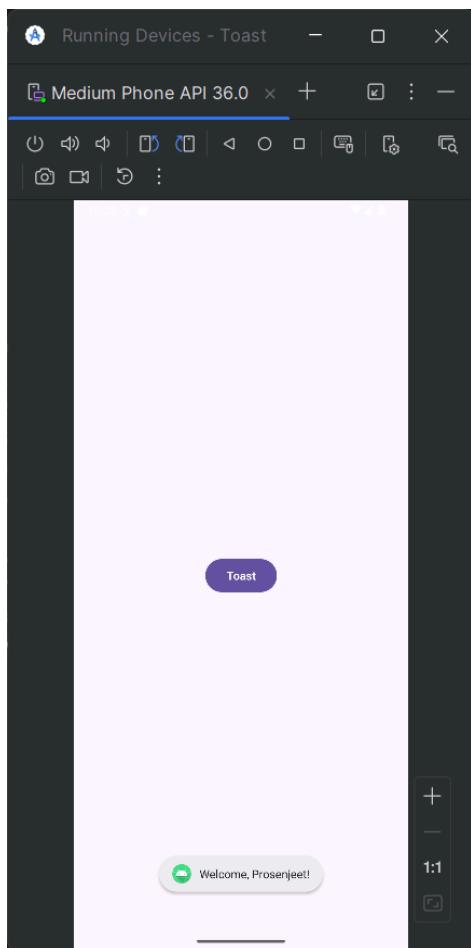
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}

public void Toaster(View view) {
    Toast.makeText(this, "Welcome, Prosenjeet!", Toast.LENGTH_LONG).show();
}
}
```

Output:



Practical No: 04

Aim: Simple calculator

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/num1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:hint="Enter first number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        android:layout_marginTop="96dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:autofillHints="" /> <EditText
        android:id="@+id/num2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:hint="Enter second number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/num1"
        app:layout_constraintHorizontal_bias="0.5"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:autofillHints="" /> <Button
```

```
    android:id="@+id/add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ADD"
    android:onClick="ADD" app:layout_constraintTop_toBottomOf="@+id/num2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/sub"
    android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/sub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SUB"
        android:onClick="SUB" app:layout_constraintTop_toBottomOf="@+id/num2"
        app:layout_constraintStart_toEndOf="@+id/add"
        app:layout_constraintEnd_toStartOf="@+id/Mul"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/Mul"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MUL"
        android:onClick="MUL" app:layout_constraintTop_toBottomOf="@+id/num2"
        app:layout_constraintStart_toEndOf="@+id/sub"
        app:layout_constraintEnd_toStartOf="@+id/Div"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/Div"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DIV"
        android:onClick="DIV" app:layout_constraintTop_toBottomOf="@+id/num2"
        app:layout_constraintStart_toEndOf="@+id/Mul"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp" />

    <TextView
        android:id="@+id/tv3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Result"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintTop_toBottomOf="@+id/add"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
android:layout_marginTop="16dp" />

<EditText
    android:id="@+id/result"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="none"
    android:focusable="false" android:longClickable="false" android:cursorVisible="false"
    app:layout_constraintTop_toBottomOf="@+id/tv3"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:autofillHints="" /> </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.simplecalculator;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button; // Not strictly needed if using findViewById and onClick XML
import android.widget.EditText;
import android.widget.TextView; // Not strictly needed if using findViewById and onClick XML
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    // No need to declare these as instance variables if you find them inside performOperation
    // private EditText num1, num2, result;
    // private Button addBtn, subBtn, mulBtn, divBtn;
    // The current implementation finds them on each click, which is fine for this simple app.

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // View initialization is not explicitly done here as operations are handled
        // via android:onClick directly on the button clicks
        // and views are found within performOperation.
    }
}
```

```
// Methods linked to android:onClick in XML
public void ADD(View view) {
    performOperation(view, Operation.ADD);
}

public void SUB(View view) {
    performOperation(view, Operation.SUBTRACT);
}

public void MUL(View view) {
    performOperation(view, Operation.MULTIPLY);
}

public void DIV(View view) {
    performOperation(view, Operation.DIVIDE);
}

// Central method to handle all arithmetic operations
private void performOperation(View view, Operation operation) {
    // Find the EditTexts and TextView on each button click
    // This is okay for a simple app, but for more complex UIs,
    // you'd typically initialize them once in onCreate.
    EditText num1EditText = findViewById(R.id.num1);
    EditText num2EditText = findViewById(R.id.num2);
    EditText resultEditText = findViewById(R.id.result); // Changed to EditText for result
    display

    // Get input text
    String num1Str = num1EditText.getText().toString();
    String num2Str = num2EditText.getText().toString();

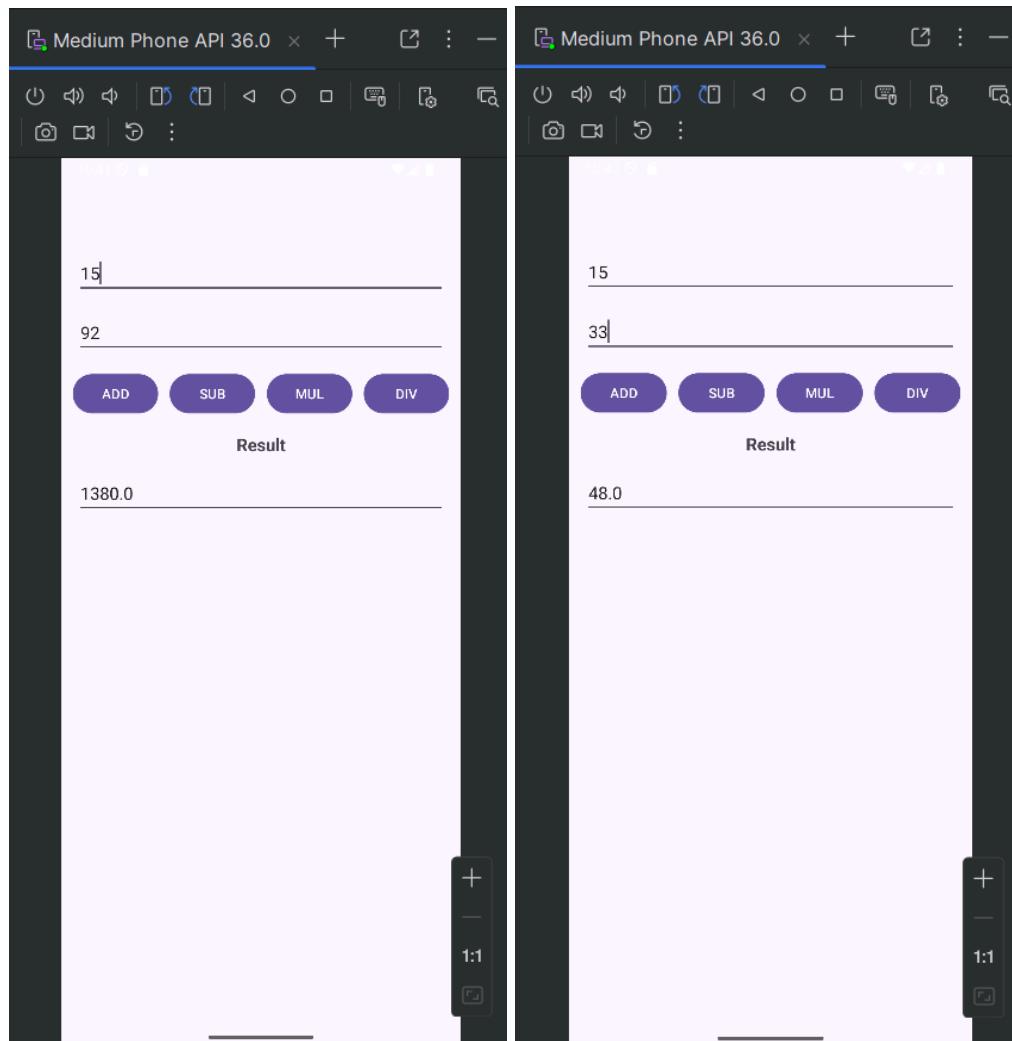
    // Input validation
    if (num1Str.isEmpty() || num2Str.isEmpty()) {
        Toast.makeText(this, "Please enter both numbers", Toast.LENGTH_SHORT).show();
        return; // Exit the method if input is missing
    }

    try {
        // Parse input to integers
        int x = Integer.parseInt(num1Str);
        int y = Integer.parseInt(num2Str);
        float result_num = 0; // Use float to handle division results accurately

        switch (operation) {
            case ADD:
                result_num = x + y;
                break;
        }
    }
}
```

```
case SUBTRACT:  
    result_num = x - y;  
    break;  
case MULTIPLY:  
    result_num = x * y;  
    break;  
case DIVIDE:  
    if (y != 0) {  
        result_num = (float) x / y; // Cast x to float for float division  
    } else {  
        Toast.makeText(this, "Cannot divide by zero", Toast.LENGTH_SHORT).show();  
        return; // Exit the method if division by zero  
    }  
    break;  
}  
// Display the result  
resultEditText.setText(String.valueOf(result_num));  
} catch (NumberFormatException e) {  
    // Handle cases where input is not a valid number  
    Toast.makeText(this, "Please enter valid numbers", Toast.LENGTH_SHORT).show();  
    e.printStackTrace(); // Print stack trace for debugging  
}  
}  
  
// Enum to clearly define operations  
private enum Operation {  
    ADD, SUBTRACT, MULTIPLY, DIVIDE  
}
```

Output:



Practical No: 05

Aim: Android program based on Intents

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <Button
            android:id="@+id/Div"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="Intenter"
            android:text="Next Page &gt;"
            tools:ignore="HardcodedText" /> </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent; // Required for using Intent
import android.os.Bundle;
import android.view.View; // Required for the View parameter in the Intenter method
```

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    /**  
     * This method is called when the Button with android:onClick="Intenter" is clicked.  
     * It starts a new Activity (MainActivity2).  
     * @param view The View (Button) that was clicked.  
     */  
    public void Intenter(View view) {  
        // Create an Intent:  
        // - 'this' refers to the current Context (MainActivity)  
        // - MainActivity2.class specifies the target Activity to start  
        Intent intent = new Intent(this, MainActivity2.class);  
        startActivity(intent); // Start the new Activity  
    }  
}
```

activity_main2.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:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity2">  
  
<TextView  
    android:id="@+id/textView"  
    android:layout_width="282dp"  
    android:layout_height="121dp"  
    android:text="Welcome to new page, Prosenjeet!" android:gravity="center"  
    android:textSize="34sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    tools:ignore="HardcodedText" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity2.java

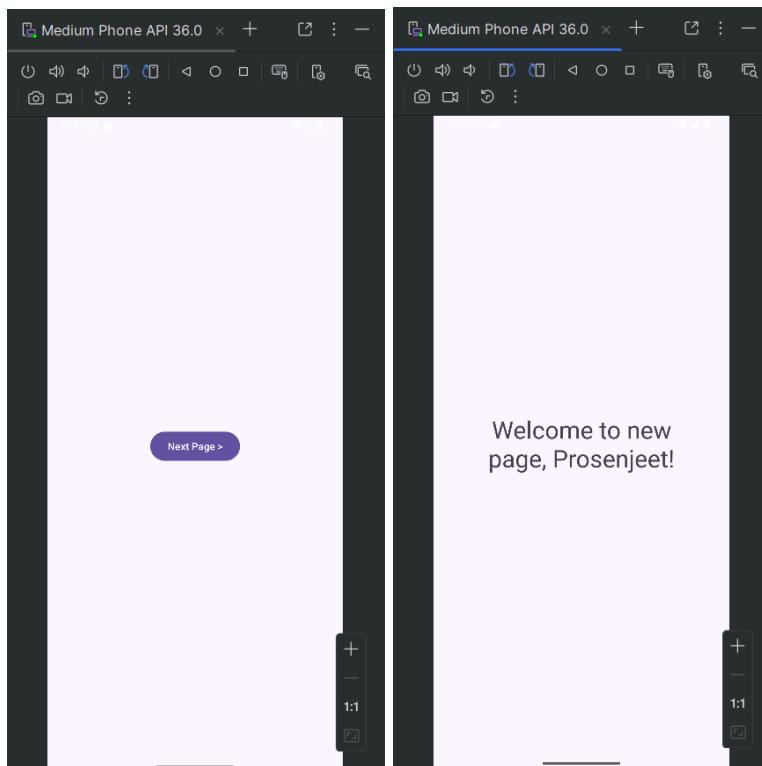
```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2); // Loads the layout for MainActivity2
    }
}
```

Output:



Practical No: 06

Aim: Android Program using various UI controls (Registration Form, Survey Form, etc..)

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Registration Form"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="24dp"
        tools:ignore="HardcodedText" />

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Name :"
        android:textSize="24sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv1"
        android:layout_marginTop="32dp"
        android:layout_marginStart="16dp"
        tools:ignore="HardcodedText" />

    <EditText
        android:id="@+id/et1"
        android:layout_width="0dp"
```

```
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Your Name"
    android:textSize="20sp"
    app:layout_constraintStart_toEndOf="@+id/tv_name"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="@+id/tv_name"
    app:layout_constraintBottom_toBottomOf="@+id/tv_name"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="16dp"
    android:autofillHints="name" />
```

```
<TextView
    android:id="@+id/tv_gender_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Gender :"
    android:textSize="24sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/et1"
    android:layout_marginTop="24dp"
    android:layout_marginStart="16dp"
    tools:ignore="HardcodedText" />
```

```
<RadioGroup
    android:id="@+id/radiogp1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintStart_toEndOf="@+id/tv_gender_label"
    app:layout_constraintTop_toTopOf="@+id/tv_gender_label"
    app:layout_constraintBottom_toBottomOf="@+id/tv_gender_label"
    android:layout_marginStart="8dp">
```

```
<RadioButton
    android:id="@+id/rb1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Female"
    android:textSize="20sp"
    tools:ignore="HardcodedText" />
```

```
<RadioButton
    android:id="@+id/rb2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:text="Male"
    android:textSize="20sp"
    android:layout_marginStart="16dp"
    tools:ignore="HardcodedText" />
</RadioGroup>

<TextView
    android:id="@+id/tv_course_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Courses :"
    android:textSize="24sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/radiogp1"
    android:layout_marginTop="24dp"
    android:layout_marginStart="16dp"
    tools:ignore="HardcodedText" />

<CheckBox
    android:id="@+id/cb1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="PHP"
    android:textSize="20sp"
    app:layout_constraintStart_toEndOf="@+id/tv_course_label"
    app:layout_constraintTop_toTopOf="@+id/tv_course_label"
    app:layout_constraintBottom_toBottomOf="@+id/tv_course_label"
    android:layout_marginStart="8dp"
    tools:ignore="HardcodedText" />

<CheckBox
    android:id="@+id/cb2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Java"
    android:textSize="20sp"
    app:layout_constraintStart_toEndOf="@+id/cb1"
    app:layout_constraintTop_toTopOf="@+id/cb1"
    app:layout_constraintBottom_toBottomOf="@+id/cb1"
    android:layout_marginStart="16dp"
    tools:ignore="HardcodedText" />

<RatingBar
    android:id="@+id/ratingbar1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
```

```
        android:stepSize="0.5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_course_label"
        android:layout_marginTop="32dp" />

    <Button
        android:id="@+id/b1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginBottom="32dp"
        tools:ignore="HardcodedText" />

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

MainActivity.java

```
package com.example.uicomponents;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup; // Important: Use RadioGroup for managing RadioButtons
import android.widget.RatingBar;
import android.widget.Toast; // For user feedback

public class MainActivity extends AppCompatActivity {
    EditText e1;
    RadioGroup rgGender; // Use RadioGroup for proper gender selection handling
    RadioButton r1, r2; // Individual radio buttons
    CheckBox c1, c2;
    Button b1;
    RatingBar rb1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

// Initialize views by finding their IDs from activity_main.xml
e1 = findViewById(R.id.et1);
rgGender = findViewById(R.id.radiogp1); // Initialize RadioGroup
r1 = findViewById(R.id.rb1); // Initialize individual RadioButtons
r2 = findViewById(R.id.rb2);
c1 = findViewById(R.id.cb1);
c2 = findViewById(R.id.cb2);
b1 = findViewById(R.id.b1);
rb1 = findViewById(R.id.ratingbar1);

// Set up the button click listener
b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 1. Get Name
        String name = e1.getText().toString().trim(); // .trim() removes leading/trailing spaces
        if (name.isEmpty()) {
            Toast.makeText(MainActivity.this, "Please enter your name",
Toast.LENGTH_SHORT).show();
            return; // Stop execution if name is empty
        }

        // 2. Get Gender
        String gender = "Not Selected";
        int selectedGenderId = rgGender.getCheckedRadioButtonId(); // Get ID of selected
radio button
        if (selectedGenderId != -1) { // Check if any radio button is selected
            RadioButton selectedRadioButton = findViewById(selectedGenderId);
            gender = selectedRadioButton.getText().toString();
        } else {
            Toast.makeText(MainActivity.this, "Please select your gender",
Toast.LENGTH_SHORT).show();
            return;
        }

        // 3. Get Course (Multiple selection possible with CheckBoxes)
        StringBuilder courseBuilder = new StringBuilder();
        if (c1.isChecked()) {
            courseBuilder.append(c1.getText().toString());
        }
        if (c2.isChecked()) {
            if (courseBuilder.length() > 0) {
                courseBuilder.append(", ");
            } // Add comma if more than one course selected
        }
    }
})
```

```
        courseBuilder.append(c2.getText().toString());
    }
    String course = courseBuilder.toString();
    if (course.isEmpty()) {
        course = "None Selected"; // Default if no courses are checked
    }

    // 4. Get Rating
    String rating = String.valueOf(rb1.getRating());

    // Create Intent to start MainActivity2
    Intent intent = new Intent(MainActivity.this, MainActivity2.class);

    // Put data as Extras into the Intent
    intent.putExtra("NAME", name);
    intent.putExtra("GENDER", gender);
    intent.putExtra("COURSE", course);
    intent.putExtra("RATING", rating);

    // Start the new Activity
    startActivity(intent);
}
});
```

activity_main2.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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/tv_welcome_title" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome!"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```

```
        android:layout_marginTop="48dp"
        tools:ignore="HardcodedText" />

<TextView
    android:id="@+id/tv_name"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintTop_toBottomOf="@+id/tv_welcome_title"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="32dp"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    tools:text="Name : John Doe" /> <TextView
    android:id="@+id/tv_gender"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintTop_toBottomOf="@+id/tv_name"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    tools:text="Gender : Female" />

<TextView
    android:id="@+id/tv_Course"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintTop_toBottomOf="@+id/tv_gender"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    tools:text="Courses : PHP, Java" />

<TextView
    android:id="@+id/tv_rating"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintTop_toBottomOf="@+id/tv_Course"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    tools:text="Rating : 4.5 Stars" />

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

MainActivity2.java

```
package com.example.uicomponents;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.widget.TextView;

public class MainActivity2 extends AppCompatActivity {

    // Declare TextViews as instance variables for clarity
    TextView tvName, tvGender, tvCourse, tvRating;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        // Initialize TextViews
        tvName = findViewById(R.id.tv_name); // Corresponds to the tv_name in
        activity_main2.xml
        tvGender = findViewById(R.id.tv_gender);
        tvCourse = findViewById(R.id.tv_Course);
        tvRating = findViewById(R.id.tv_rating);

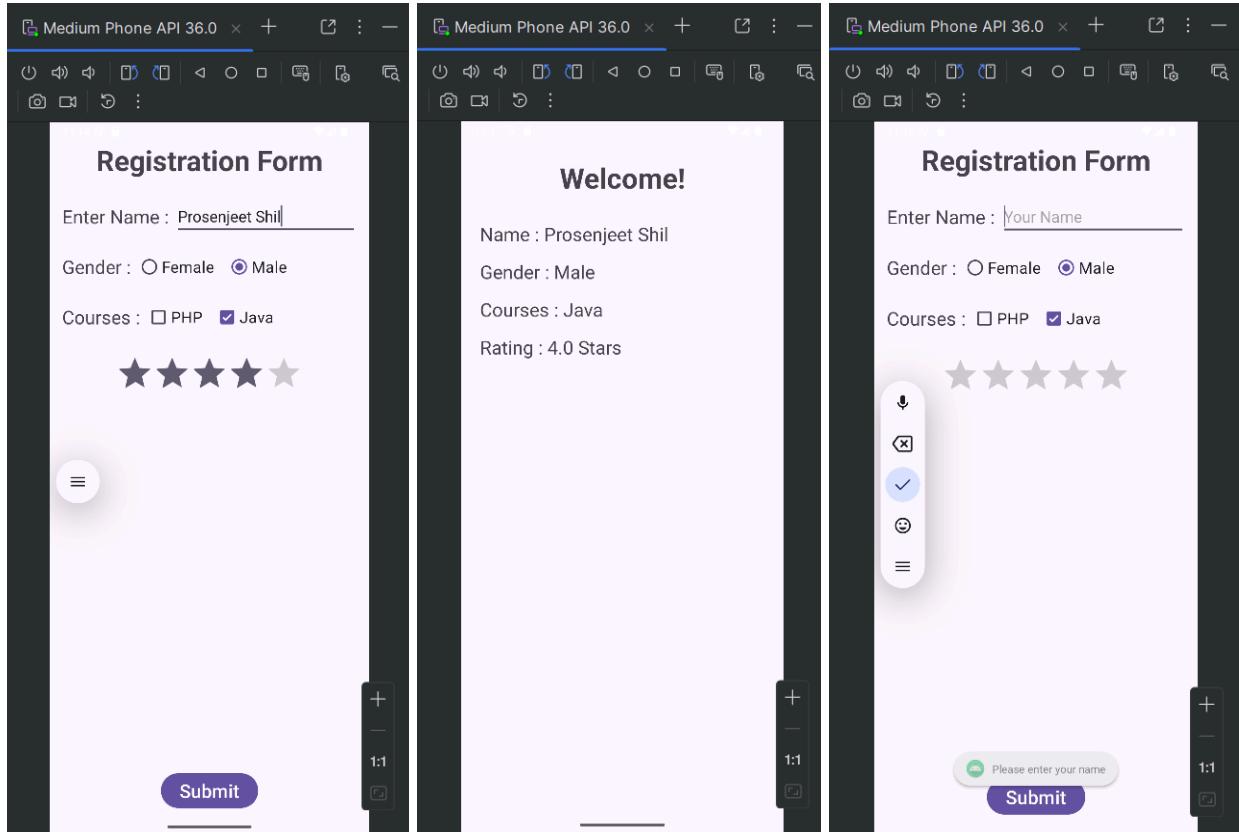
        // Get the Intent that started this Activity
        Intent intent = getIntent();

        // Retrieve the data using the same keys used in MainActivity
        String name = intent.getStringExtra("NAME");
        String gender = intent.getStringExtra("GENDER");
        String course = intent.getStringExtra("COURSE");
        String rating = intent.getStringExtra("RATING"); // Rating is passed as String from
        MainActivity

        // Set the retrieved data to the TextViews
        tvName.setText("Name : " + name);
```

```
tvGender.setText("Gender : " + gender);
tvCourse.setText("Courses : " + course);
tvRating.setText("Rating : " + rating + " Stars"); // Added "Stars" for clarity
}
}
```

Output:



Practical No: 07

Aim: Button Onclicklistener, Image View

Code:

activity_main.xml

```
<RelativeLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/iv2"
        android:layout_width="match_parent"
        android:layout_height="667dp"
        android:layout_above="@+id/bt1"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="0dp"
        android:layout_marginBottom="19dp"
        android:src="@drawable/ic_launcher_foreground" />

    <Button
        android:id="@+id/bt1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="change"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="-3dp"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

MainActivity.java

```
package com.example.buttononclicklistener;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
```

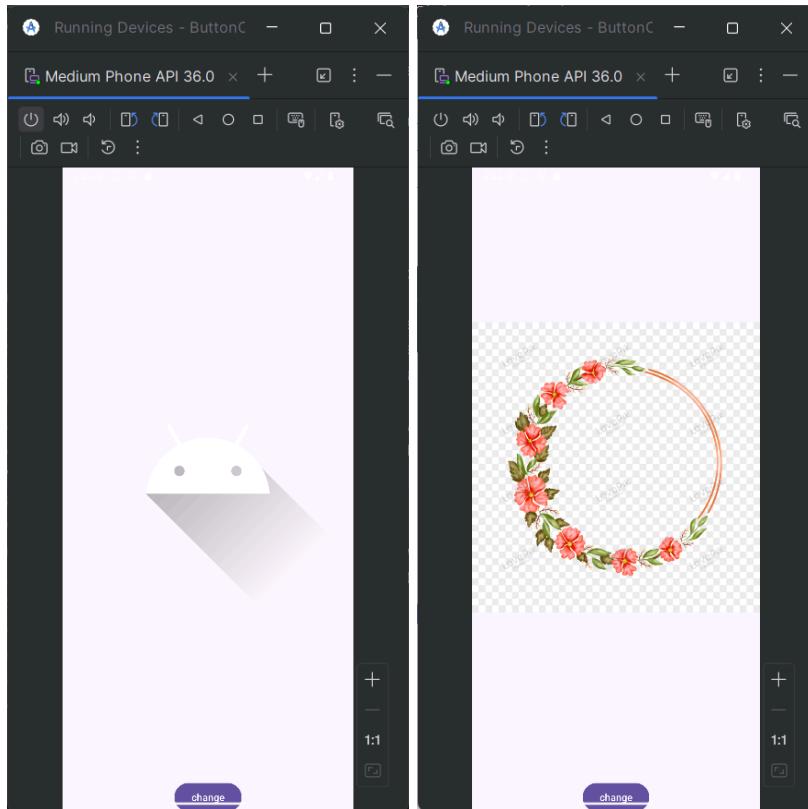
```
import android.widget.Button;
import android.widget.ImageView;

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

        Button btn1 = (Button) findViewById(R.id.bt1);
        ImageView iv = (ImageView) findViewById(R.id.iv2);

        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                iv.setImageResource(R.drawable.pic);
            }
        });
    }
}
```

Output:



Practical No: 08

Aim: Canvas (Draw Circle)

Code:

MainActivity.java

```
package com.example.canvas;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.content.Context;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new Mview(this)); // Sets the content view to an instance of our custom
Mview class
    }

    // A custom inner class that extends View to create a custom drawing surface
    private class Mview extends View {
        public Mview(Context context) {
            super(context);
        }

        @Override
        public void onDraw(Canvas canvas) {
            super.onDraw(canvas);

            // Get the width and height of the view
            int x = getWidth();
            int y = getHeight();

            int radius = 100; // Define the radius of the circle

            // Create a Paint object to define the drawing style and color
        }
    }
}
```

```
Paint p = new Paint();
p.setStyle(Paint.Style.FILL);

// Set the background color to white
p.setColor(Color.WHITE);
canvas.drawPaint(p);

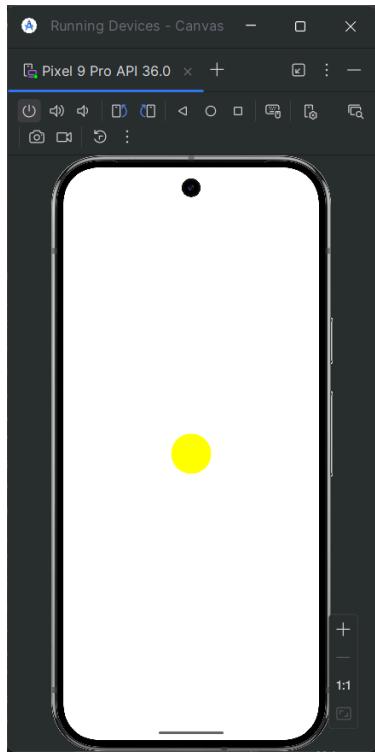
// Set the circle color to yellow
p.setColor(Color.parseColor("yellow"));

// Draw a circle in the center of the view with the defined radius
canvas.drawCircle(x / 2, y / 2, radius, p);

// This line draws a rectangle, but the coordinates are likely incorrect as provided
// canvas.drawRect(300, 250, 100, 100, p);
// I've commented this out as it may not produce the intended result.

    }
}
}
```

Output:



Practical No: 09

Aim: Alert Box

Code:

activity_main.xml

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

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="50dp"
        android:text="@string/hello_world"
        android:textSize="24sp"
        tools:ignore="HardcodedText" />

    <Button
        android:id="@+id/Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textViewTitle"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:text="@string/show_alert"
        tools:ignore="HardcodedText" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.alertdialog;

import androidx.appcompat.app.AlertDialog;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.DialogInterface;
import android.widget.Button;
import android.view.View;
import android.widget.Toast; // Added for a more user-friendly "NO" action

public class MainActivity extends AppCompatActivity {
    private Button buttonSbm; // Declared as an instance variable

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

        // Initialize the button here.
        buttonSbm = findViewById(R.id.Button);

        // Set the click listener on the button.
        onButtonClickListener();
    }

    public void onButtonClickListener() {
        buttonSbm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new AlertDialog.Builder instance.
                AlertDialog.Builder a_builder = new AlertDialog.Builder(MainActivity.this);

                // Set the message and title of the alert dialog.
                a_builder.setMessage("Do you Want to close this app?")
                    .setTitle("Alert!!!") // Title can be set here or after .create()
                    ..setCancelable(false); // Prevents the user from dismissing the dialog by tapping
                outside it.

                // Set the "YES" button (Positive action).
                a_builder.setPositiveButton("YES", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        // This code runs when the user clicks "YES".
                        finish(); // Closes the current Activity, effectively closing the app in this case.
                    }
                });

                // Set the "NO" button (Negative action).
                a_builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {

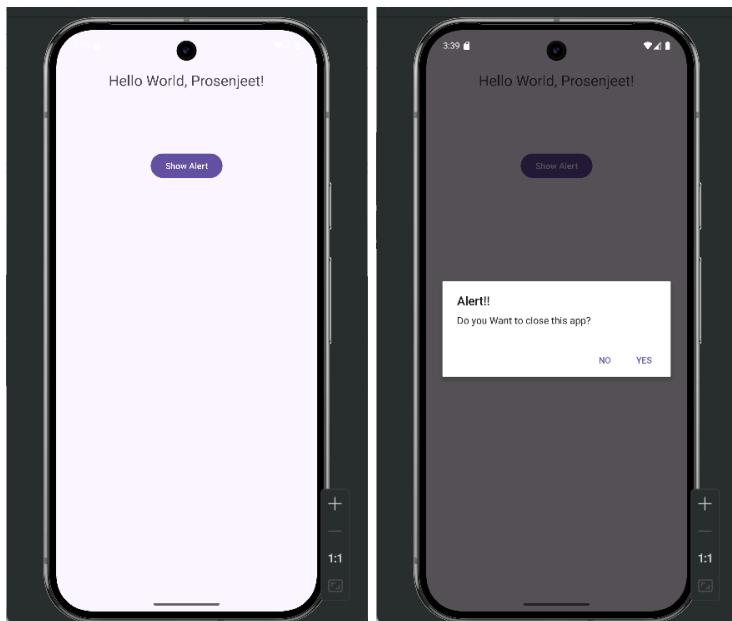
```

```
// This code runs when the user clicks "NO".  
// The dialog automatically dismisses when a button is clicked.  
// We can add a Toast here for user feedback.  
    Toast.makeText(MainActivity.this, "Action cancelled.",  
Toast.LENGTH_SHORT).show();  
    dialog.cancel(); // Dismisses the dialog (though it would have already done so)  
}  
});  
  
// Create the AlertDialog object and show it.  
AlertDialog alert = a_builder.create();  
alert.show();  
}  
});  
}  
}
```

strings.xml

```
<resources>  
<string name="app_name">AlertDialog</string>  
  
<string name="hello_world">Hello World, Prosenjeet!</string>  
<string name="show_alert">Show Alert</string>  
</resources>
```

Output:



Practical No: 10

Aim: File Handling (Read, Write, Delete)

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/txtcontent"
        android:layout_width="0dp"
        android:layout_height="65dp"
        android:hint="@string/enter_text_here"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="HardcodedText,LabelFor"
        android:autofillHints="" />

    <Button
        android:id="@+id/btnwrite"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="@string/write"
        app:layout_constraintEnd_toStartOf="@+id/btnread"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtcontent"
        tools:ignore="HardcodedText" />

    <Button
        android:id="@+id/btnread"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
```

```
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:text="@string/read"
        app:layout_constraintBottom_toBottomOf="@+id(btnwrite"
        app:layout_constraintEnd_toStartOf="@+id	btn_clear"
        app:layout_constraintStart_toEndOf="@+id	btnwrite"
        app:layout_constraintTop_toTopOf="@+id	btnwrite"
        tools:ignore="HardcodedText" />

<Button
    android:id="@+id/button_clear"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/clear"
    app:layout_constraintBottom_toBottomOf="@+id	btnread"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id	btnread"
    app:layout_constraintTop_toTopOf="@+id	btnread"
    tools:ignore="HardcodedText"
    android:onClick="clearFile" />

<Button
    android:id="@+id/button_delete"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/delete"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id	btnread"
    tools:ignore="HardcodedText"
    android:onClick="deleteFile" />

<TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:textSize="18sp"
    android:text="@string/no_file_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id	button_delete"
    tools:ignore="HardcodedText" />

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

MainActivity.java

```
package com.example.filehandling;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class MainActivity extends AppCompatActivity {
    EditText et_Text;
    Button b_read, b_write;
    TextView tv_Text;
    Button b_delete; // Delete button is now declared correctly
    String file_name = "data.txt"; // A more descriptive filename is better

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

        // Initialize all the views
        et_Text = findViewById(R.id.txtcontent);
        b_read = findViewById(R.id.btnread);
        b_write = findViewById(R.id.btnwrite);
        tv_Text = findViewById(R.id.textView);
        b_delete = findViewById(R.id.btn_delete);

        // --- Set up the OnClickListener for the WRITE button ---
        b_write.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Get text from EditText and save it to the file
                String content = et_Text.getText().toString();
                saveFile(file_name, content);
            }
        });

        // --- Set up the OnClickListener for the READ button ---
    }

    private void saveFile(String fileName, String content) {
        try {
            FileOutputStream fos = new FileOutputStream(fileName);
            fos.write(content.getBytes());
            fos.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private String readFile(String fileName) {
        String content = "";
        try {
            FileInputStream fis = new FileInputStream(fileName);
            byte[] buffer = new byte[fis.available()];
            fis.read(buffer);
            content = new String(buffer);
            fis.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return content;
    }
}
```

```
b_read.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Read the content from the file and display it
        String content = readFile(file_name);
        tv_Text.setText(content);
    }
});

// The clear and delete buttons are handled via `android:onClick` in XML,
// so we don't need to define listeners for them here.
}

// --- Method to save data to a file in internal storage ---
public void saveFile(String fileName, String text) {
    try {
        FileOutputStream fos = openFileOutput(fileName, Context.MODE_PRIVATE);
        fos.write(text.getBytes());
        fos.close();
        Toast.makeText(this, "Text saved successfully!", Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(this, "Error saving file", Toast.LENGTH_SHORT).show();
    }
}

// --- Method to read data from a file in internal storage ---
public String readFile(String fileName) {
    String text = "";
    try {
        FileInputStream fis = openFileInput(fileName);
        int size = fis.available();
        byte[] buffer = new byte[size];
        fis.read(buffer);
        fis.close();
        text = new String(buffer);
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(this, "Error reading file or file not found.",
Toast.LENGTH_SHORT).show();
    }
    return text;
}

// --- Method to clear the EditText content (linked via android:onClick) ---
public void clearFile(View view) {
    et_Text.getText().clear();
```

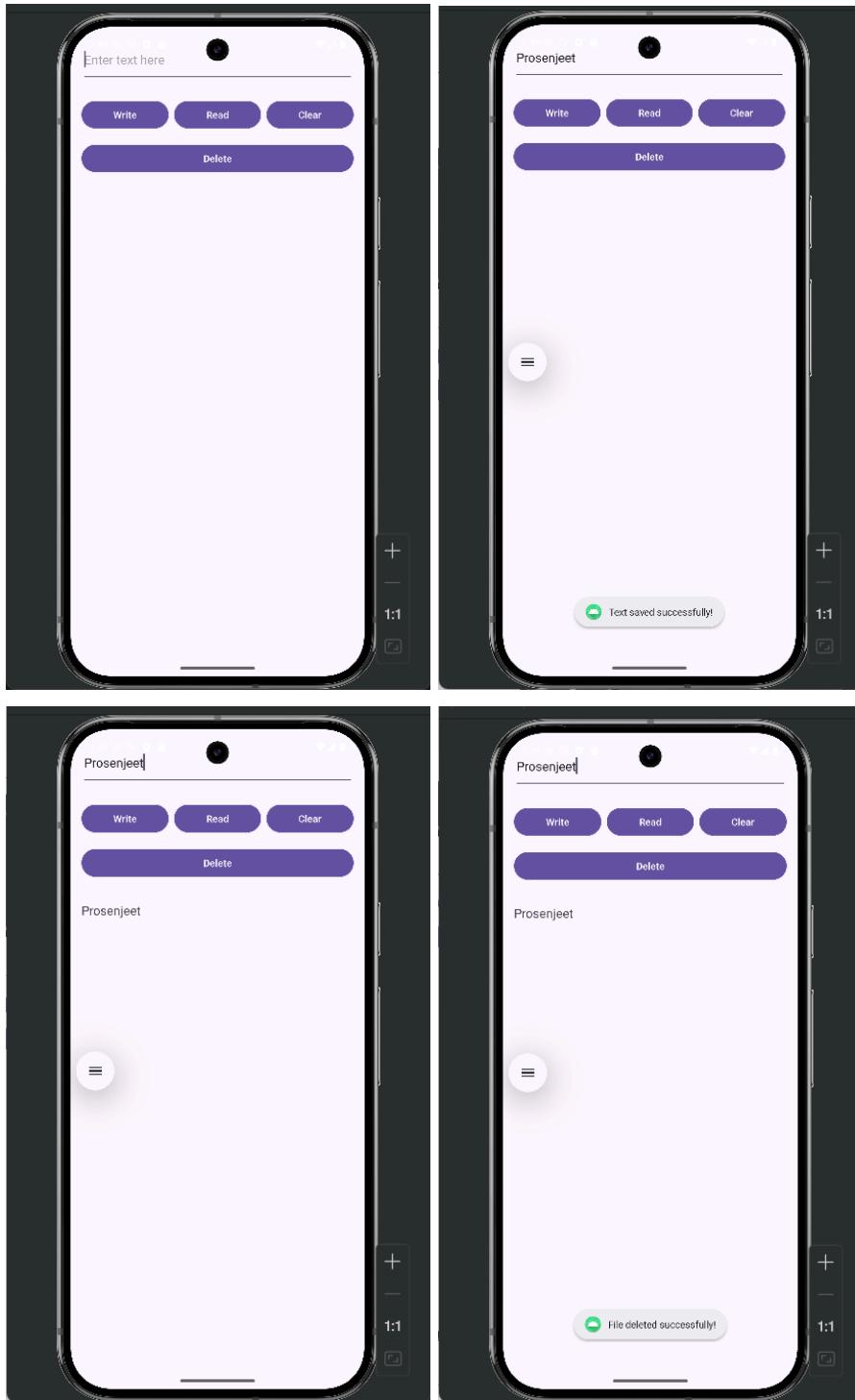
```
}

// --- Method to delete the file (linked via android:onClick) ---
public void deleteFile(View view) {
    File file = new File(getFilesDir(), file_name);
    if (file.exists()) {
        file.delete();
        Toast.makeText(this, "File deleted successfully!", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(this, "File not found.", Toast.LENGTH_SHORT).show();
    }
}
```

strings.xml

```
<resources>
    <string name="app_name">FileHandling</string>
    <string name="enter_text_here">Enter text here</string>
    <string name="write">Write</string>
    <string name="read">Read</string>
    <string name="clear">Clear</string>
    <string name="delete">Delete</string>
    <string name="no_file_content">No file content</string>
</resources>
```

Output:



Practical No: 11 & 12

Aim: Wifi, Bluetooth (Enable/Disable)

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/wifi_enable_disable"
        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/switch1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_chainStyle="packed"
        app:layout_constraintVertical_bias="0.5" />

    <Switch
        android:id="@+id/switch1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="36dp"
        android:textOff="@string/off"
        android:textOn="@string/on"
        app:layout_constraintBottom_toBottomOf="@+id/tv1"
        app:layout_constraintStart_toEndOf="@+id/tv1"
        app:layout_constraintTop_toTopOf="@+id/tv1" />

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:layout_marginTop="32dp"
    android:text="@string/bluetooth_enable_disable"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv1" />

<Switch
    android:id="@+id/switch3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="36dp"
    android:textOff="@string/off"
    android:textOn="@string/on"
    app:layout_constraintBottom_toBottomOf="@+id/tv"
    app:layout_constraintStart_toEndOf="@+id/tv"
    app:layout_constraintTop_toTopOf="@+id/tv" />

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

MainActivity.java

```
package com.example.connectivity;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.bluetooth.BluetoothAdapter;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.wifi.WifiManager;
import android.os.Build;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private static final int BLUETOOTH_PERMISSION_REQUEST_CODE = 1;
    private static final int BLUETOOTH_ENABLE_REQUEST_CODE = 2;

    Switch wifi, bluetooth;
    WifiManager wifiManager;
```

```
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

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

    wifiManager = (WifiManager)
getApplicationContext().getSystemService(Context.WIFI_SERVICE);
    wifiiii = findViewById(R.id.switch1);
    bluetoooot = findViewById(R.id.switch3);

    // Set initial state for Wi-Fi
    if (wifiManager != null) {
        wifiiii.setChecked(wifiManager.isWifiEnabled());
    }

    // Set initial state for Bluetooth
    if (bluetoothAdapter != null) {
        bluetoooot.setChecked(bluetoothAdapter.isEnabled());
    }

    // --- Set OnCheckedChangeListener for WIFI ---
    wifiiii.setOnCheckedChangeListener((buttonView, isChecked) -> {
        if (wifiManager == null) {
            Toast.makeText(this, "Wi-Fi not supported on this device.",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (isChecked) {
            wifiManager.setWifiEnabled(true);
            Toast.makeText(this, "WIFI ON", Toast.LENGTH_SHORT).show();
        } else {
            wifiManager.setWifiEnabled(false);
            Toast.makeText(this, "WIFI Off", Toast.LENGTH_SHORT).show();
        }
    });

    // --- Set OnCheckedChangeListener for BLUETOOTH ---
    bluetoooot.setOnCheckedChangeListener((buttonView, isChecked) -> {
        if (bluetoothAdapter == null) {
            Toast.makeText(getApplicationContext(), "Bluetooth not supported on this device.",
Toast.LENGTH_SHORT).show();
            return;
        }
        if (isChecked) {
```

```
// Check for BLUETOOTH_CONNECT permission on Android 12+
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S &&
    ContextCompat.checkSelfPermission(this,
        android.Manifest.permission.BLUETOOTH_CONNECT) != PackageManager.PERMISSION_GRANTED) {

    // Request the permission from the user
    ActivityCompat.requestPermissions(this, new
        String[]{android.Manifest.permission.BLUETOOTH_CONNECT},
        BLUETOOTH_PERMISSION_REQUEST_CODE);

} else {
    // Permission is already granted or not needed (on older versions)
    if (!bluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new
            Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, BLUETOOTH_ENABLE_REQUEST_CODE);
    }
}
} else {
    // Disable Bluetooth
    bluetoothAdapter.disable();
    Toast.makeText(getApplicationContext(), "Bluetooth Turned Off",
        Toast.LENGTH_SHORT).show();
}
});

}

// This method handles the result of the runtime permission request
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
    @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == BLUETOOTH_PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
            PackageManager.PERMISSION_GRANTED) {
            // Permission granted, now attempt to enable Bluetooth
            if (bluetoothAdapter != null && !bluetoothAdapter.isEnabled()) {
                Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, BLUETOOTH_ENABLE_REQUEST_CODE);
            }
        } else {
            // Permission denied, inform the user
            Toast.makeText(this, "Bluetooth permission is required to enable Bluetooth.",
                Toast.LENGTH_SHORT).show();
            bluetoot.setChecked(false); // Reset the switch to reflect the denial
        }
    }
}
```

```
        }  
    }  
  
    // This method handles the result of the Bluetooth enable intent  
    @Override  
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
        super.onActivityResult(requestCode, resultCode, data);  
        if (requestCode == BLUETOOTH_ENABLE_REQUEST_CODE) {  
            if (resultCode == RESULT_OK) {  
                // Bluetooth was successfully enabled  
                Toast.makeText(this, "Bluetooth is ON", Toast.LENGTH_SHORT).show();  
            } else {  
                // User denied enabling Bluetooth  
                Toast.makeText(this, "Failed to turn on Bluetooth.", Toast.LENGTH_SHORT).show();  
                bluetoooot.setChecked(false); // Reset switch state  
            }  
        }  
    }  
}
```

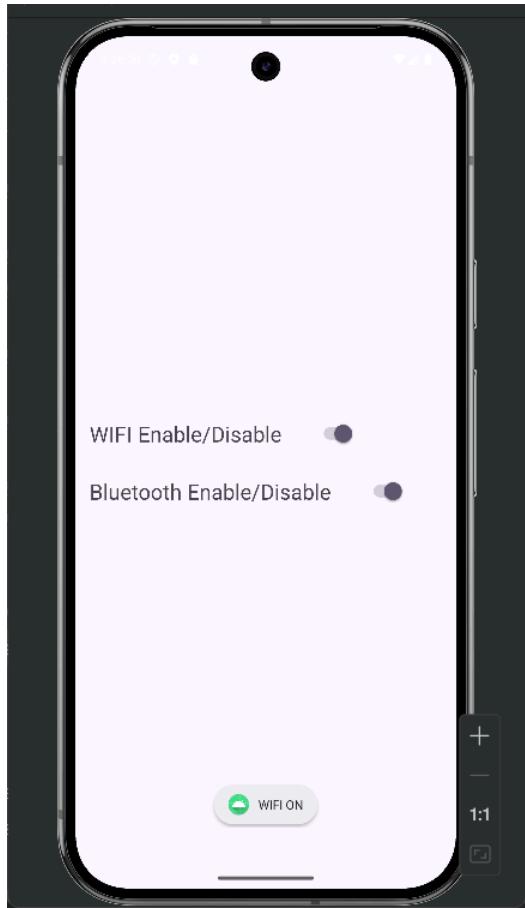
strings.xml

```
<resources>  
    <string name="app_name">Connectivity</string>  
    <string name="wifi_enable_disable">WIFI Enable/Disable</string>  
    <string name="bluetooth_enable_disable">Bluetooth Enable/Disable</string>  
    <string name="on">ON</string>  
    <string name="off">OFF</string>  
</resources>
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.BLUETOOTH"  
    android:maxSdkVersion="30" />  
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"  
    android:maxSdkVersion="30" />  
    <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />  
  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />  
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
```

Output:



Practical No: 13

Aim: Create an application to CRUD operations in SQLite

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="@string/contact_us_title"
        android:textColor="?attr/colorPrimary"
        android:textSize="24sp"
        android:textStyle="bold"
        app:fontFamily="serif"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:hint="@string/enter_your_name"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5"
        tools:ignore="Autofill" />

    <EditText
```

```
    android:id="@+id/editTextEmailAddress"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:hint="@string/enter_your_email_address"
    android:inputType="textEmailAddress"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextName"
    tools:ignore="Autofill" />
```

```
<EditText
    android:id="@+id/editTextMessage"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:hint="@string/enter_your_message"
    android:inputType="textMultiLine"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextEmailAddress"
    tools:ignore="Autofill" />
```

```
<Button
    android:id="@+id/btnsave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/save"
    app:layout_constraintEnd_toStartOf="@+id(btnupdate)"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextMessage" />
```

```
<Button
    android:id="@+id	btnupdate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/update"
    app:layout_constraintEnd_toStartOf="@+id/btndel"
    app:layout_constraintStart_toEndOf="@+id/btnsave"
    app:layout_constraintTop_toBottomOf="@+id/editTextMessage" />
```

```
<Button
    android:id="@+id/btndel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="16dp"
    android:text="@string/delete"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id(btnupdate"
    app:layout_constraintTop_toBottomOf="@+id/editTextMessage" />

<Button
    android:id="@+id/btnselect"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/view"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnsave" />

<Button
    android:id="@+id/btnselectperticular"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="@string/search"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnselect" />

<EditText
    android:id="@+id/editTextSearch"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="@string/search_hint"
    android:inputType="text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btnselectperticular"
    tools:ignore="Autofill" />

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

MainActivity.java

```
package com.example.crudssqlite;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
```

```
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    // Class variables for the database and UI elements
    SQLiteDatabase db;
    EditText editSearchContact, editName, editEmailAddress, editMessage;
    Button btnSave, btnDelete, btnModify, btnView, btnSearch;

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

        // Create or open the database. It will be created if it doesn't exist.
        db = openOrCreateDatabase("ContactDb", Context.MODE_PRIVATE, null);

        // Create the Contact table if it doesn't exist.
        // cld is the primary key and will auto-increment.
        db.execSQL("CREATE TABLE IF NOT EXISTS Contact(cld INTEGER PRIMARY KEY
        AUTOINCREMENT, Name VARCHAR, Mail VARCHAR, CMessage VARCHAR);");

        // Initialize all the views by finding their IDs from the XML layout.
        editSearchContact = findViewById(R.id.editTextSearch);
        editName = findViewById(R.id.editTextName);
        editEmailAddress = findViewById(R.id.editTextEmailAddress);
        editMessage = findViewById(R.id.editTextMessage);
        btnSave = findViewById(R.id.btnsave);
       .btnDelete = findViewById(R.id.btndel);
        btnModify = findViewById(R.id.btnupdate);
        btnView = findViewById(R.id.btnselect);
        btnSearch = findViewById(R.id.btnselectperticular);

        // Set the click listeners for all the buttons.
        btnSave.setOnClickListener(this);
       .btnDelete.setOnClickListener(this);
        btnModify.setOnClickListener(this);
        btnView.setOnClickListener(this);
        btnSearch.setOnClickListener(this);
    }

    // Helper method to display a Toast message.
```

```
private void showMessage(Context context, String message) {
    Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
}

// The main click handler for all buttons.
@Override
public void onClick(View v) {
    int id = v.getId(); // Get the ID of the clicked button

    if (id == R.id.btnsave) {
        saveData();
    } else if (id == R.id.btnupdate) {
        updateData();
    } else if (id == R.id.btndel) {
        deleteData();
    } else if (id == R.id.btnselect) {
        selectAllData();
    } else if (id == R.id.btnselectparticular) {
        selectParticularData();
    }
}

// --- CRUD Operation Methods ---

private void saveData() {
    // Check if any of the required fields are empty.
    if (isAnyFieldEmpty(editName, editEmailAddress, editMessage)) {
        showMessage(this, "Please enter all values");
        return;
    }

    // SQL INSERT statement with '?' to prevent SQL injection.
    String sql = "INSERT INTO Contact(Name, Mail, CMessage) VALUES(?, ?, ?)";
    // Execute the SQL statement with the values from the EditTexts.
    db.execSQL(sql, new Object[]{
        editName.getText().toString(),
        editEmailAddress.getText().toString(),
        editMessage.getText().toString()
    });

    clearFields();
    showMessage(this, "Response Noted, Thanks!!");
}

private void updateData() {
    // Check if the search field is empty.
    if (editSearchContact.getText().toString().trim().isEmpty()) {
```

```
showMessage(this, "Enter Name in the Search field to update");
return;
}

// Check if the record exists before updating.
Cursor c = db.rawQuery("SELECT * FROM Contact WHERE Name=?", new
String[]{editSearchContact.getText().toString()});
if (c.moveToFirst()) {
    // SQL UPDATE statement
    String sql = "UPDATE Contact SET Name=?, Mail=?, CMessage=? WHERE Name=?";
    db.execSQL(sql, new Object[]{
        editName.getText().toString(),
        editEmailAddress.getText().toString(),
        editMessage.getText().toString(),
        editSearchContact.getText().toString()
    });
    clearFields();
    editSearchContact.getText().clear();
    showMessage(this, "Record Modified");
} else {
    showMessage(this, "Invalid Name or Record not found");
}
c.close();
}

private void deleteData() {
    // Check if the search field is empty.
    if (editSearchContact.getText().toString().trim().isEmpty()) {
        showMessage(this, "Please enter Name in the Search field to delete");
        return;
    }

    // Check if the record exists before deleting.
    Cursor c = db.rawQuery("SELECT * FROM Contact WHERE Name=?", new
String[]{editSearchContact.getText().toString()});
    if (c.moveToFirst()) {
        // SQL DELETE statement
        db.execSQL("DELETE FROM Contact WHERE Name=?", new
String[]{editSearchContact.getText().toString()});
        clearFields();
        editSearchContact.getText().clear();
        showMessage(this, "Record Deleted");
    } else {
        showMessage(this, "Invalid Name or Record not found");
    }
    c.close();
}
```

```
private void selectAllData() {
    // Get all records from the Contact table.
    Cursor c = db.rawQuery("SELECT * FROM Contact", null);
    if (c.getCount() == 0) {
        showMessage(this, "No records found");
        return;
    }

    StringBuilder buffer = new StringBuilder();
    while (c.moveToFirst()) {
        buffer.append("Name: ").append(c.getString(1)).append("\n");
        buffer.append("Mail: ").append(c.getString(2)).append("\n");
        buffer.append("Message: ").append(c.getString(3)).append("\n\n");
    }
    // Display all records in a single Toast.
    // Note: This may be problematic for a large number of records.
    showMessage(this, buffer.toString());
    c.close();
}

private void selectParticularData() {
    // Check if the search field is empty.
    if (editSearchContact.getText().toString().trim().isEmpty()) {
        showMessage(this, "Enter Name in the Search field");
        return;
    }

    // Select a specific record by name.
    Cursor c = db.rawQuery("SELECT * FROM Contact WHERE Name=?", new
String[]{editSearchContact.getText().toString()});
    if (c.moveToFirst()) {
        // Populate the EditText fields with the found data.
        editTextName.setText(c.getString(1));
        editTextEmail.setText(c.getString(2));
        editTextMessage.setText(c.getString(3));
        showMessage(this, "Record found");
    } else {
        showMessage(this, "Invalid Name or Record not found");
    }
    c.close();
}

// --- Helper Methods ---
private boolean isEmptyField(EditText... fields) {
    for (EditText field : fields) {
        if (field.getText().toString().trim().isEmpty()) {
```

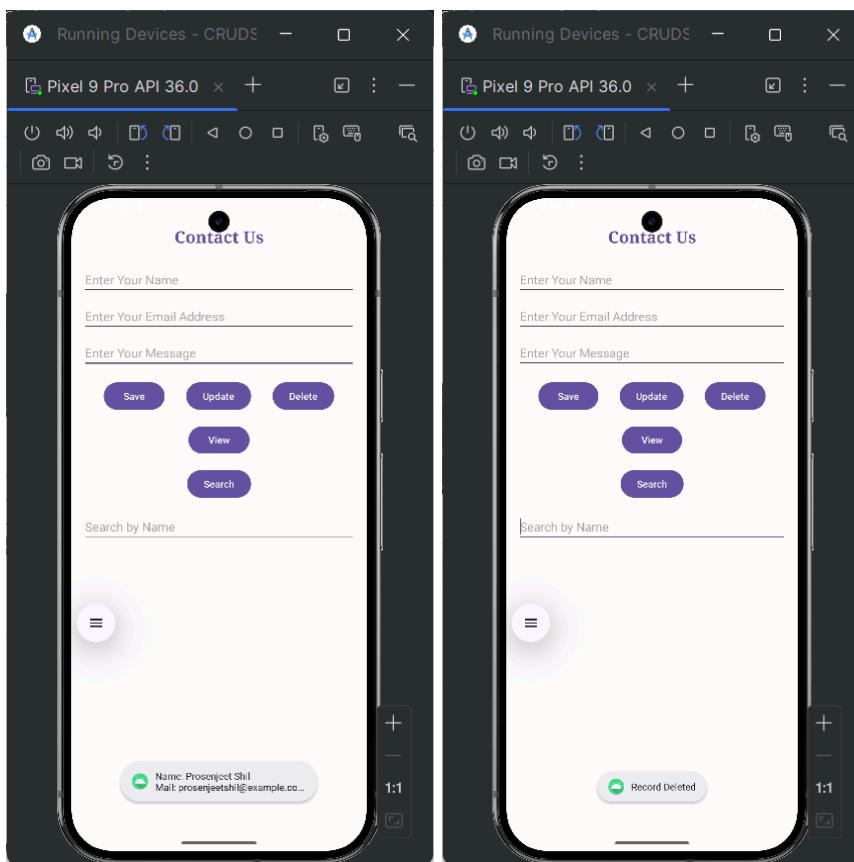
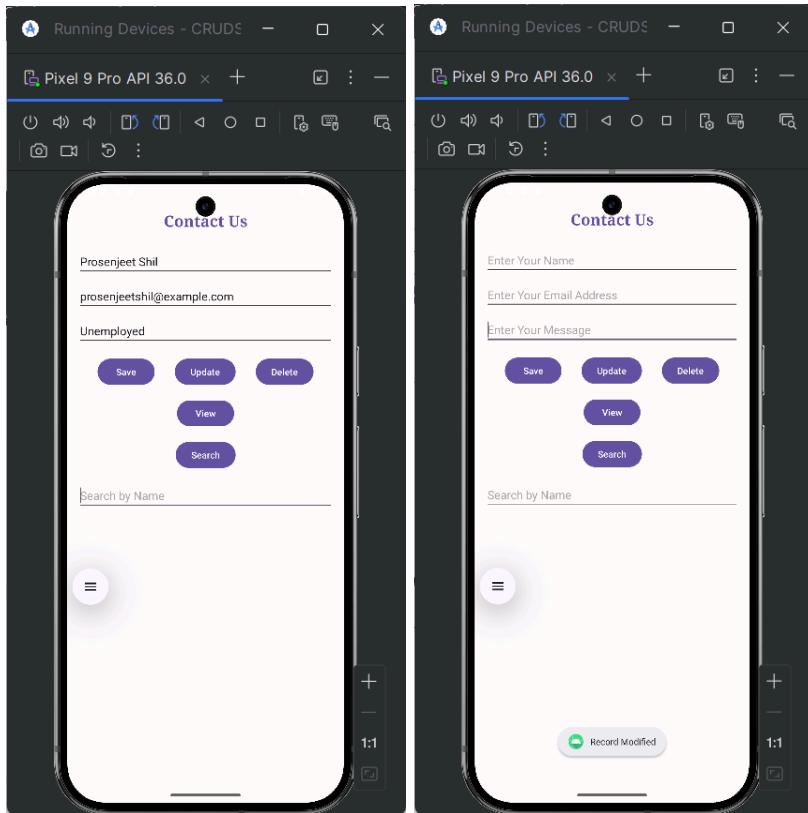
```
        return true;
    }
}
return false;
}

private void clearFields() {
    editName.getText().clear();
    editEmailAddress.getText().clear();
    editMessage.getText().clear();
}
}
```

strings.xml

```
<resources>
    <string name="app_name">Practical_13_48_SQLite</string>
    <string name="contact_us_title">Contact Us</string>
    <string name="enter_your_name">Enter Your Name</string>
    <string name="enter_your_email_address">Enter Your Email Address</string>
    <string name="enter_your_message">Enter Your Message</string>
    <string name="save">Save</string>
    <string name="update">Update</string>
    <string name="delete">Delete</string>
    <string name="view">View</string>
    <string name="search">Search</string>
    <string name="search_hint">Search by Name</string>
</resources>
```

Output:



Practical No: 14

Aim: Create an application for animation (Fade, Rotate, Move and blink)

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp"
        app:srcCompat="@drawable/pic" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/imageView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:onClick="move"
        android:text="Move" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:onClick="fade"
        android:text="Fade" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="8dp"
    android:onClick="rotate"
    android:text="Rotate" />

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button3"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="8dp"
    android:onClick="blink"
    android:text="Blink" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.animation;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {
    private ImageView imageView;

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

    // Method called by the "Move" button
    public void move(View view) {
        startAnimation(R.anim.move);
```

```
}

// Method called by the "Fade" button
public void fade(View view) {
    startAnimation(R.anim.fade_in);
}

// Method called by the "Rotate" button
public void rotate(View view) {
    startAnimation(R.anim.rotate);
}

// Method called by the "Blink" button
public void blink(View view) {
    startAnimation(R.anim.blink);
}

// Helper method to load and start an animation
private void startAnimation(int animationResource) {
    Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
animationResource);
    imageView.startAnimation(animation);
}
}
```

move.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">
    <translate
        android:fromXDelta="0%p"
        android:toXDelta="50%p"
        android:duration="1000" />
</set>
```

fade_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true"
    android:interpolator="@android:anim/linear_interpolator">
    <alpha
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
```

```
        android:duration="1000" />  
</set>
```

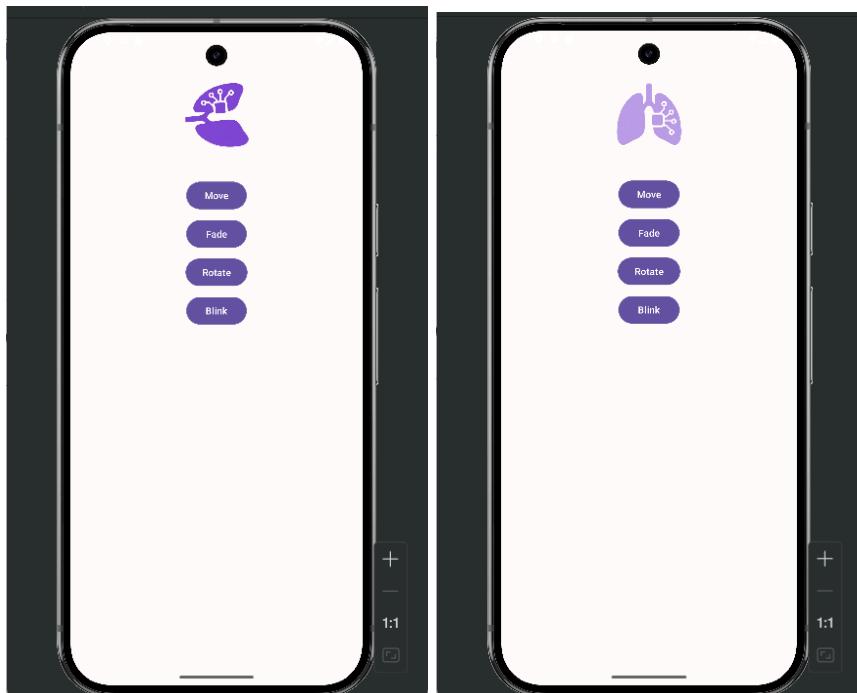
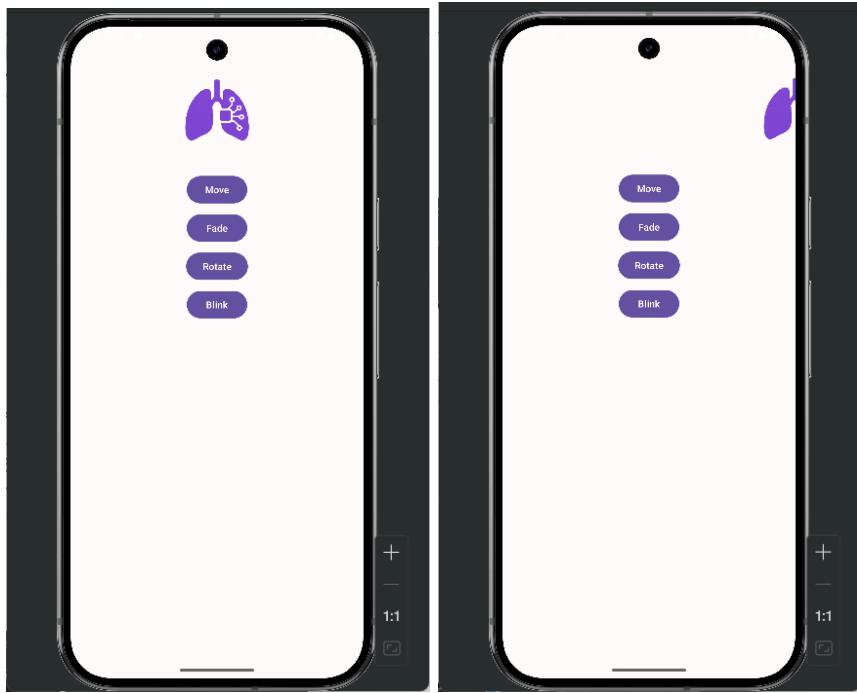
rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
      android:fillAfter="true"  
      android:interpolator="@android:anim/linear_interpolator">  
    <rotate  
        android:fromDegrees="0"  
        android:toDegrees="360"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:duration="1000" />  
</set>
```

blink.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
      android:fillAfter="true"  
      android:interpolator="@android:anim/linear_interpolator">  
    <alpha  
        android:fromAlpha="0.0"  
        android:toAlpha="1.0"  
        android:duration="500"  
        android:repeatCount="infinite"  
        android:repeatMode="reverse" />  
</set>
```

Output:



Practical No: 15

Aim: Shared reference

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView_emailId"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/email_id"
        android:textSize="24sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_chainStyle="packed"
        app:layout_constraintBottom_toTopOf="@+id/textView_password"/>

    <EditText
        android:id="@+id/editText_email"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:hint="Email"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView_emailId"
        app:layout_constraintTop_toTopOf="@+id/textView_emailId"
        app:layout_constraintBottom_toBottomOf="@+id/textView_emailId"
        tools:ignore="Autofill,HardcodedText"/>

    <TextView
        android:id="@+id/textView_password"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="@string/password"
    android:textSize="24sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView_emailId"
    app:layout_constraintBottom_toTopOf="@+id/checkBox"/>
```

```
<EditText
    android:id="@+id/editText_password"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView_password"
    app:layout_constraintTop_toTopOf="@+id/textView_password"
    app:layout_constraintBottom_toBottomOf="@+id/textView_password"
    tools:ignore="Autofill,HardcodedText"/>
```

```
<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="@string/remember_me"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView_password"
    app:layout_constraintBottom_toTopOf="@+id/button_signIn"/>
```

```
<Button
    android:id="@+id/button_signIn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:onClick="Save"
    android:text="@string/sign_in"
    android:textSize="18sp"
    app:layout_constraintEnd_toStartOf="@+id/button_getData"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/checkBox"/>
```

```
<Button
    android:id="@+id/button_getData"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="24dp"
        android:onClick="Get"
        android:text="@string/get_data"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/button_signIn"
        app:layout_constraintTop_toBottomOf="@+id/checkBox"/>

<Button
    android:id="@+id/button_clear"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:onClick="clear"
    android:text="@string/clear"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_signIn" />

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

MainActivity.java

```
package com.example.sharedpreferences;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.CheckBox;
import android.widget.EditText; // Use EditText instead of TextView
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    // Class variables
    SharedPreferences sharedpreferences;
    EditText password, email;
    CheckBox remember;

    public static final String MY_PREFERENCES = "my_pref";
    public static final String EMAIL_KEY = "emailKey";
```

```
public static final String PASSWORD_KEY = "passwordKey";

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

    // Initialize views
    email = findViewById(R.id.editText_email);
    password = findViewById(R.id.editText_password);
    remember = findViewById(R.id.checkBox);

    // Get the SharedPreferences object
    sharedPreferences = getSharedPreferences(MY_PREFERENCES,
Context.MODE_PRIVATE);

    // Check if credentials are saved and pre-fill fields
    if (sharedPreferences.contains(EMAIL_KEY)) {
        email.setText(sharedPreferences.getString(EMAIL_KEY, ""));
    }
    if (sharedPreferences.contains(PASSWORD_KEY)) {
        password.setText(sharedPreferences.getString(PASSWORD_KEY, ""));
    }
}

// Method called by the "Get Data" button
public void Get(View view) {
    // Get credentials from SharedPreferences and display them
    if (sharedPreferences.contains(EMAIL_KEY) &&
sharedPreferences.contains(PASSWORD_KEY)) {
        email.setText(sharedPreferences.getString(EMAIL_KEY, ""));
        password.setText(sharedPreferences.getString(PASSWORD_KEY, ""));
        Toast.makeText(this, "Data retrieved from preferences", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(this, "No data found in preferences", Toast.LENGTH_SHORT).show();
    }
}

// Method called by the "Sign In" button
public void Save(View view) {
    // Check if the "Remember Me" checkbox is checked
    if (remember.isChecked()) {
        // Get the text from the EditText fields
        String userEmail = email.getText().toString();
        String userPassword = password.getText().toString();

        // Get a SharedPreferences.Editor to modify the data
    }
}
```

```
SharedPreferences.Editor editor = sharedPreferences.edit();

// Put the email and password into the editor
editor.putString(EMAIL_KEY, userEmail);
editor.putString(PASSWORD_KEY, userPassword);

// Commit the changes
editor.apply(); // Use apply() for asynchronous saving

    Toast.makeText(this, "Value stored in shared preference",
Toast.LENGTH_SHORT).show();
} else {
    Toast.makeText(this, "Remember Me is not checked", Toast.LENGTH_SHORT).show();
}
}

// Method called by the "Clear" button
public void clear(View view) {
    // Clear the text fields
    email.setText("");
    password.setText("");

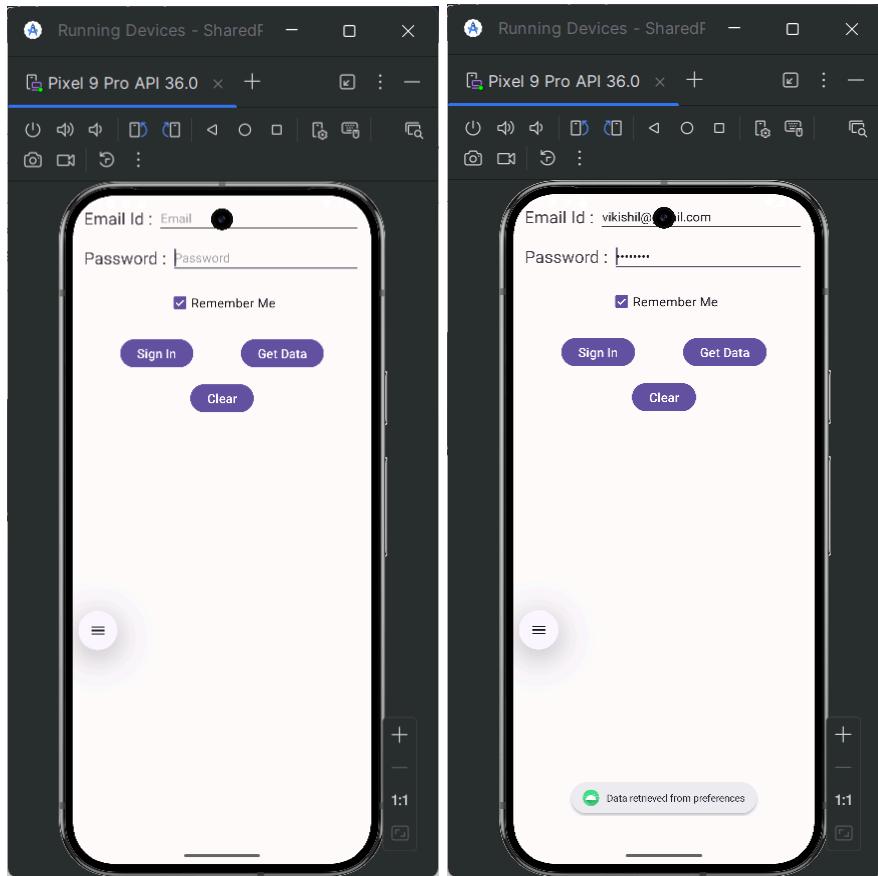
    // Clear the data from SharedPreferences as well
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear();
    editor.apply();

    Toast.makeText(this, "Cleared fields and data", Toast.LENGTH_SHORT).show();
}
}
```

strings.xml

```
<resources>
    <string name="app_name">Practical_15_48_SharedPreferences</string>
    <string name="email_id">Email Id :</string>
    <string name="password">Password :</string>
    <string name="remember_me">Remember Me</string>
    <string name="sign_in">Sign In</string>
    <string name="clear">Clear</string>
    <string name="get_data">Get Data</string>
    <string name="value_stored">Value stored in shared preference</string>
</resources>
```

Output:



Practical No: 16

Aim: Image Download

Code:

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:text="Web Page Image"
        android:textSize="25sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="HardcodedText" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_margin="32dp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        tools:ignore="ContentDescription" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="32dp"
```

```
    android:text="Download"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="HardcodedText" />

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

MainActivity.java

```
package com.example.imagedownload;

import android.content.Context;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask; // Modern approach to background tasks
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Objects;

public class MainActivity extends AppCompatActivity {
    ImageView imageView;
    Button b1;
    AlertDialog progressDialog;

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

        imageView = findViewById(R.id.imageView);
        b1 = findViewById(R.id.button);
```

```
b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Check for internet connection first
        if (checkInternetConnection()) {
            // Use a valid image URL for the practical
            String imageUrl =
"https://hips.hearstapps.com/hmg-prod/images/2022-chevrolet-corvette-z06-1607016574.jp
g";
            // Start the AsyncTask to download the image
            new DownloadImageTask().execute(imageUrl);
        } else {
            Toast.makeText(MainActivity.this, "No internet connection!",
Toast.LENGTH_LONG).show();
        }
    }
});

// A helper method to check for a valid internet connection
private boolean checkInternetConnection() {
    ConnectivityManager connec = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetwork = connec.getActiveNetworkInfo();
    return activeNetwork != null && activeNetwork.isConnectedOrConnecting();
}

// An AsyncTask to handle the background download and UI updates
private class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = ProgressDialog.show(MainActivity.this, "", "Downloading Image...");
    }

    @Override
    protected Bitmap doInBackground(String... urls) {
        Bitmap bitmap = null;
        try {
            URL url = new URL(urls[0]);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setDoInput(true);
            connection.connect();
            InputStream input = connection.getInputStream();
            bitmap = BitmapFactory.decodeStream(input);
        }
    }
}
```

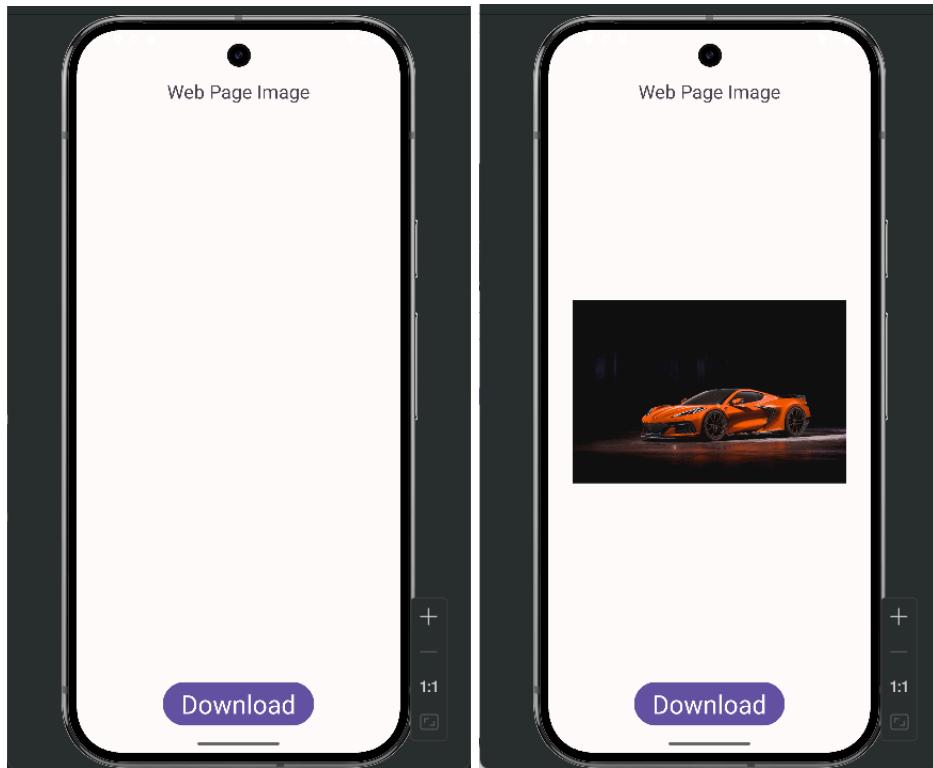
```
        } catch (IOException e) {
            e.printStackTrace();
        }
        return bitmap;
    }

@Override
protected void onPostExecute(Bitmap result) {
    super.onPostExecute(result);
    if (progressDialog != null && progressDialog.isShowing()) {
        progressDialog.dismiss();
    }
    if (result != null) {
        imageView.setImageBitmap(result);
        Toast.makeText(MainActivity.this, "Download successful!",
Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(MainActivity.this, "Failed to download image.",
Toast.LENGTH_SHORT).show();
    }
}
}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Output:



Practical No: 17

Aim: Video Player

Code:

Add the Video Resource:

- In the "Project" pane, right-click the `res` folder.
- Select New > Android Resource Directory.
- In the dialog, set Directory name to `raw` and Resource type to `raw`.
- Click OK.
- Now, copy your video file (e.g., `please.mp4`) into the new `res/raw` folder. Make sure the file name is all lowercase, without spaces or special characters.

activity_main.xml

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

    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Simple Video Player"
        android:textSize="24sp"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

    <VideoView
        android:id="@+id/videoView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textViewTitle"
        android:layout_marginTop="32dp" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.videoplayer;

import androidx.appcompat.app.AppCompatActivity;
import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {
    VideoView videoView;
    MediaController mediaController;

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

        // Initialize VideoView and MediaController
        videoView = findViewById(R.id.videoView1);
        mediaController = new MediaController(this);

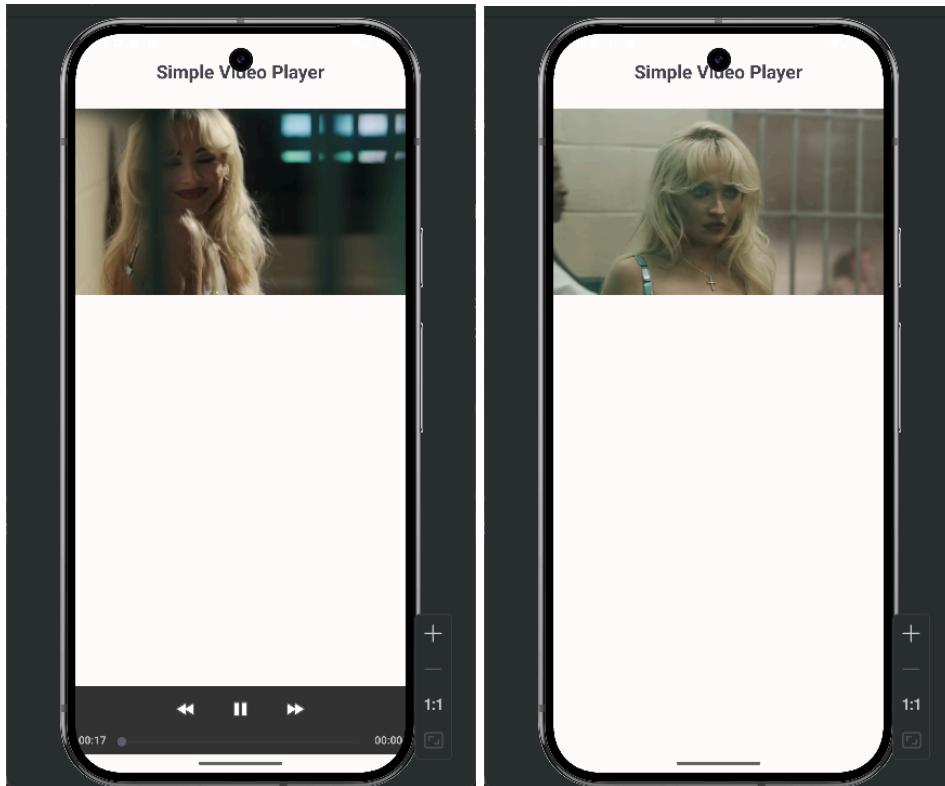
        try {
            // Construct the URI for the video resource. R.raw.tom corresponds to
            res/raw/please.mp4
            Uri videoUri = Uri.parse("android.resource://" + getPackageName() + "/" +
R.raw.please);

            // Set the video URI to the VideoView
            videoView.setVideoURI(videoUri);

            // Attach the MediaController to the VideoView
            videoView.setMediaController(mediaController);
            mediaController.setAnchorView(videoView);

            // Start video playback
            videoView.start();
        } catch (Exception e) {
            // Handle cases where the video file might not be found
            e.printStackTrace();
            Toast.makeText(this, "Error: Video file not found or could not be loaded.",
Toast.LENGTH_LONG).show();
        }
    }
}
```

Output:



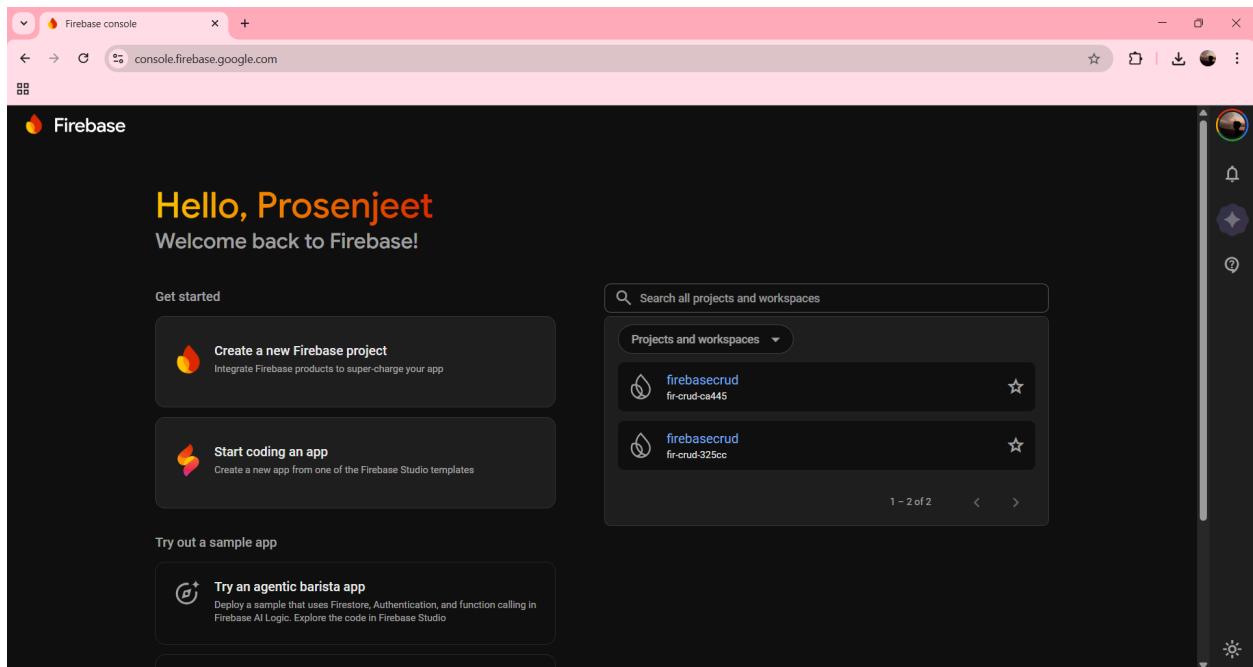
Practical No: 18

Aim: FIREBASE

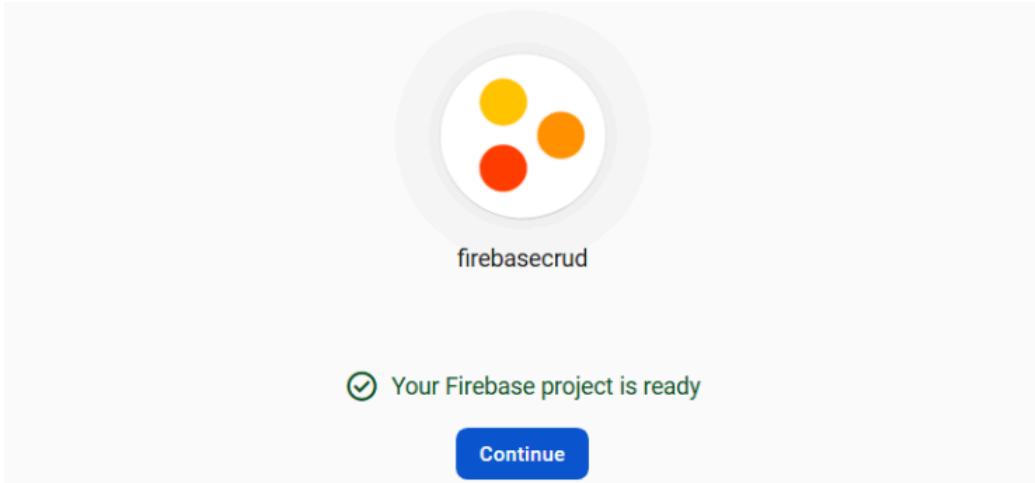
To use Firebase Realtime Database in your Android project, you need to follow these Firebase Setup Prerequisites:

1. Create Firebase Project

1. Go to: <https://console.firebaseio.google.com>

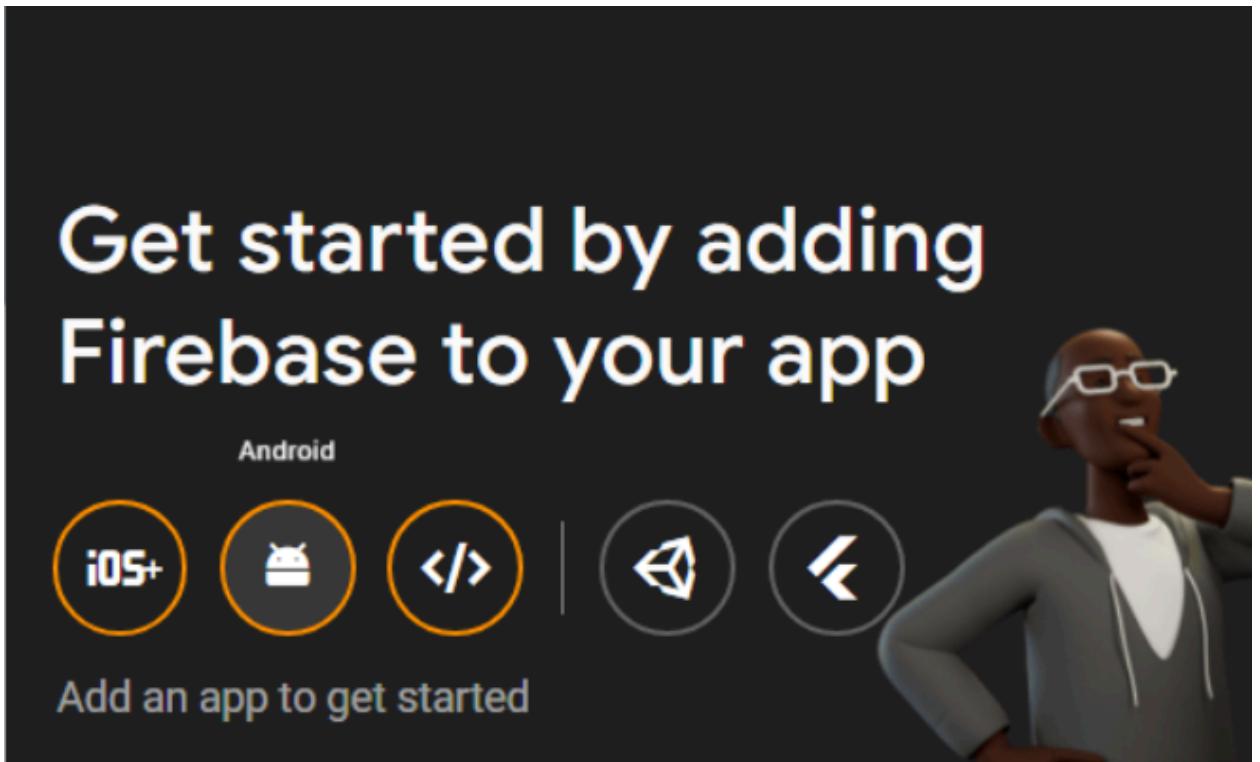


2. Click "Add Project".
3. Enter a **Project Name**, click **Continue**.
4. Enable or disable **Google Analytics** (optional), then click **Create Project**.
5. Wait for Firebase to set up your project.



2. Register Your Android App

1. Inside your Firebase project dashboard, click "Add app" → **Android icon**.



2. Enter the following details:

- **Android package name** (e.g., com.example.firebaseio) – **Required**
- **App nickname** – (Optional)

- SHA-1 – (Optional for Realtime Database; Required for Firebase Auth)
3. Click "Register App".

3. Download and Add `google-services.json`

1. After app registration, click "**Download google-services.json**".
2. Move this file into your Android project under:

```
app/  
└── google-services.json
```

4. Add Firebase SDK to Your Project

Project-level `build.gradle` (`ProjectName/build.gradle`):

```
buildscript {  
    dependencies {  
        classpath 'com.google.gms:google-services:4.4.1'  
    }  
}
```

App-level `build.gradle` (`app/build.gradle`):

```
plugins {  
    id 'com.android.application'  
    id 'com.google.gms.google-services' // Add this line  
}  
  
dependencies {  
    // Firebase Realtime Database and Auth  
    implementation 'com.google.firebaseio:firebase-database:20.3.0'  
    implementation 'com.google.firebase:firebase-auth:22.3.0'  
}
```

5. Sync Project with Gradle

- Click "**Sync Now**" in Android Studio when prompted.

6. Enable Realtime Database in Firebase Console

1. Go to the [Firebase Console](#) → Select your project.

2. Navigate to Build > Realtime Database.
3. Click "Create Database".
4. Choose a region, and start in **test mode** for development.

Test Mode Rules (for development only):

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

5. Click "**Publish**" to save the rules.

Code:

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:layout_width="match_parent"  
  android:layout_height="match_parent"  
  tools:context=".MainActivity">  
  
<LinearLayout  
  android:padding="16dp"  
  android:orientation="vertical"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:fitsSystemWindows="true">  
  
  <!-- Input Field for Name -->  
  <!-- Removed outdated 'android:background' that caused the error -->  
  <EditText  
    android:id="@+id/etName"  
    android:hint="Name"  
    android:padding="12dp"  
    android:layout_marginBottom="8dp"
```

```
    android:inputType="textPersonName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<!-- Input Field for Email -->
<!-- Removed outdated 'android:background' that caused the error -->
<EditText
    android:id="@+id/etEmail"
    android:hint="Email"
    android:padding="12dp"
    android:layout_marginBottom="16dp"
    android:inputType="textEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<!-- Buttons for CRUD Operations -->
<Button
    android:id="@+id	btnAdd"
    android:text="Add User"
    android:layout_marginBottom="8dp"
    android:backgroundTint="#6200EE"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id	btnUpdate"
    android:text="Update User"
    android:layout_marginBottom="8dp"
    android:backgroundTint="#03DAC5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id	btnDelete"
    android:text="Delete User"
    android:layout_marginBottom="8dp"
    android:backgroundTint="#CF6679"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id	btnViewAll"
    android:text="View All Users"
    android:layout_marginBottom="16dp"
    android:backgroundTint="#FF9800"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

```
<!-- ListView to display users -->
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:divider="#DDDDDD"
    android:dividerHeight="1dp"/>

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

MainActivity.java

```
package com.example.firebaseiocrud;

import android.os.Bundle;
import android.view.View;
import android.widget.*;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.analytics.FirebaseAnalytics;
import com.google.firebase.database.*;

import java.util.*;

public class MainActivity extends AppCompatActivity {
    EditText etName, etEmail;
    Button btnAdd, btnUpdate, btnDelete, btnViewAll;
    ListView listView;

    FirebaseAnalytics firebaseAnalytics;
    DatabaseReference dbRef;
    List<User> userList;
    String selectedUserId = "";

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

        // Initialize Firebase services
        firebaseAnalytics = FirebaseAnalytics.getInstance(this);
```

```
dbRef = FirebaseDatabase.getInstance().getReference("users");
userList = new ArrayList<>();

// Initialize views
etName = findViewById(R.id.etName);
etEmail = findViewById(R.id.etEmail);
btnAdd = findViewById(R.id.btnAdd);
btnUpdate = findViewById(R.id.btnUpdate);
btnDelete = findViewById(R.id.btnDelete);
btnViewAll = findViewById(R.id.btnViewAll);
listView = findViewById(R.id.listView);

// Button actions
btnAdd.setOnClickListener(view -> {
    addUser();
    logEvent("add_user");
});

btnUpdate.setOnClickListener(view -> {
    updateUser();
    logEvent("update_user");
});

btnDelete.setOnClickListener(view -> {
    deleteUser();
    logEvent("delete_user");
});

btnViewAll.setOnClickListener(view -> {
    retrieveUsers();
});

// List item click listener to load data for update/delete
listView.setOnItemClickListener((adapterView, view, i, l) -> {
    User user = userList.get(i);
    etName.setText(user.getName());
    etEmail.setText(user.getEmail());
    selectedUserId = user.getId();
});
}

/**
 * Logs an event to Firebase Analytics.
 * @param action The specific action performed (e.g., "add_user").
 */
private void logEvent(String action) {
    Bundle bundle = new Bundle();
```

```
bundle.putString(FirebaseAnalytics.Param.METHOD, action);
firebaseAnalytics.logEvent(FirebaseAnalytics.Event.SELECT_CONTENT, bundle);
}

/**
 * Adds a new user record to the Firebase Realtime Database.
 */
private void addUser() {
    String name = etName.getText().toString().trim();
    String email = etEmail.getText().toString().trim();

    if (name.isEmpty() || email.isEmpty()) {
        Toast.makeText(this, "Enter name and email", Toast.LENGTH_SHORT).show();
        return;
    }

    // Generate a unique ID for the new user
    String id = dbRef.push().getKey();

    User user = new User(id, name, email);

    // Save the user object to the database
    if (id != null) {
        dbRef.child(id).setValue(user);
        Toast.makeText(this, "User Added", Toast.LENGTH_SHORT).show();
        etName.setText("");
        etEmail.setText("");
    } else {
        Toast.makeText(this, "Error generating user ID", Toast.LENGTH_SHORT).show();
    }
}

/**
 * Updates the currently selected user's record in the database.
 */
private void updateUser() {
    if (!selectedUserId.isEmpty()) {
        String name = etName.getText().toString().trim();
        String email = etEmail.getText().toString().trim();

        if (name.isEmpty() || email.isEmpty()) {
            Toast.makeText(this, "Name and Email cannot be empty",
Toast.LENGTH_SHORT).show();
            return;
        }

        User user = new User(selectedUserId, name, email);
    }
}
```

```
dbRef.child(selectedUserId).setValue(user);
Toast.makeText(this, "User Updated", Toast.LENGTH_SHORT).show();

// Clear fields and ID after update
etName.setText("");
etEmail.setText("");
selectedUserId = "";
} else {
    Toast.makeText(this, "Select a user to update", Toast.LENGTH_SHORT).show();
}
}

/***
 * Deletes the currently selected user's record from the database.
 */
private void deleteUser() {
    if (!selectedUserId.isEmpty()) {
        dbRef.child(selectedUserId).removeValue();
        Toast.makeText(this, "User Deleted", Toast.LENGTH_SHORT).show();

        // Clear fields and ID after delete
        etName.setText("");
        etEmail.setText("");
        selectedUserId = "";
    } else {
        Toast.makeText(this, "Select a user to delete", Toast.LENGTH_SHORT).show();
    }
}

/***
 * Fetches all users from the database and displays them in the ListView.
 */
private void retrieveUsers() {
    dbRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            userList.clear();
            for (DataSnapshot postSnapshot : snapshot.getChildren()) {
                User user = postSnapshot.getValue(User.class);
                if (user != null) {
                    userList.add(user);
                }
            }
        }

        List<String> displayList = new ArrayList<>();
        for (User user : userList) {
            // FIX: Added missing semicolon here
        }
    });
}
```

```
        displayList.add("Name: " + user.getName() + "\nEmail: " + user.getEmail());
    }

    ArrayAdapter<String> adapter = new ArrayAdapter<>(MainActivity.this,
android.R.layout.simple_list_item_1, displayList);
    listView.setAdapter(adapter);
    Toast.makeText(MainActivity.this, "Data displayed on device",
Toast.LENGTH_SHORT).show();
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Toast.makeText(MainActivity.this, "Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
}
});
```

User.java

```
package com.example.firebaseiocrud;

public class User {
    private String id;
    private String name;
    private String email;

    // Required default constructor for Firebase
    public User() {}

    public User(String id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    public String getId() {
        return id;
    }

    public String getName() {
        return name;
    }

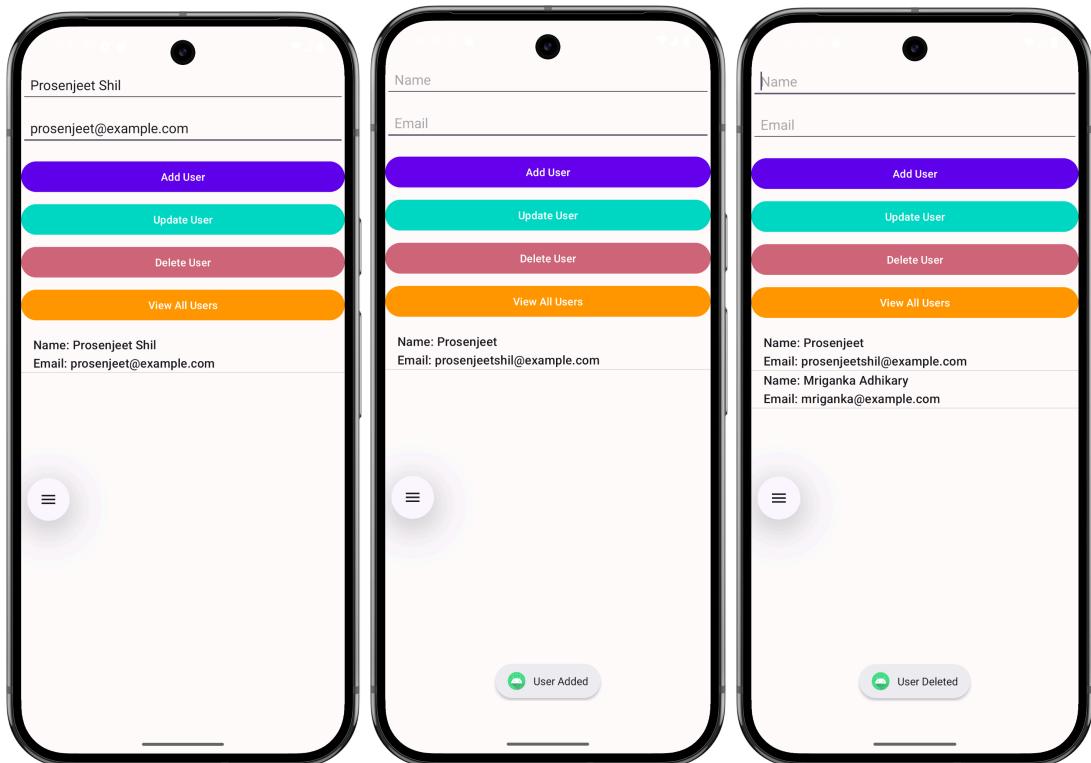
    public String getEmail() {
```

```
return email;  
}  
  
public void setId(String id) {  
    this.id = id;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```

Output:



Practical No: 19

Aim: Create a flutter Application to display the message.

Code:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Message Display',
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Message App'),
        ),
        body: const Center(
          child: Text(
            '125 PROSENJEET SHIL',
            style: TextStyle(
              fontSize: 32,
              fontWeight: FontWeight.bold,
              color: Colors.black,
            ),
          ),
        ),
      );
  }
}
```

Output:



Practical No: 20

Aim: Create a Flutter Application to display the Elevated Button.

Code:

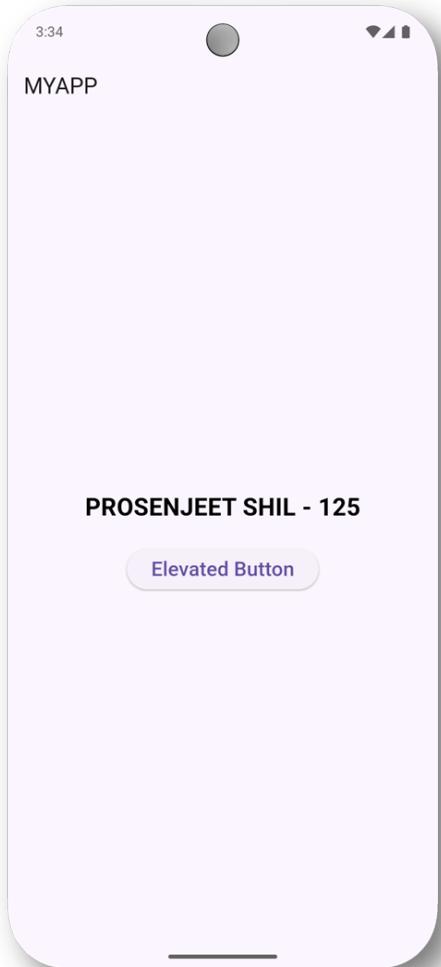
```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('MYAPP'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              // Display your name and roll number
              const Text(
                'PROSENJEET SHIL - 125',
                style: TextStyle(
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                  color: Colors.black,
                ),
              ),
              const SizedBox(height: 20),

              // Elevated Button
              ElevatedButton(
                onPressed: () {
                  // Add button action here if needed
                },
                child: const Text(
                  "Elevated Button",
                  style: TextStyle(fontSize: 20.0),
                ),
              ),
            ],
          ),
        ),
      ),
    );
}
```

```
    ),  
    ],  
    ),  
    ),  
    ),  
    );  
}  
}
```

Output:



Practical No: 21

Aim: Create a Flutter Application to display an Image.

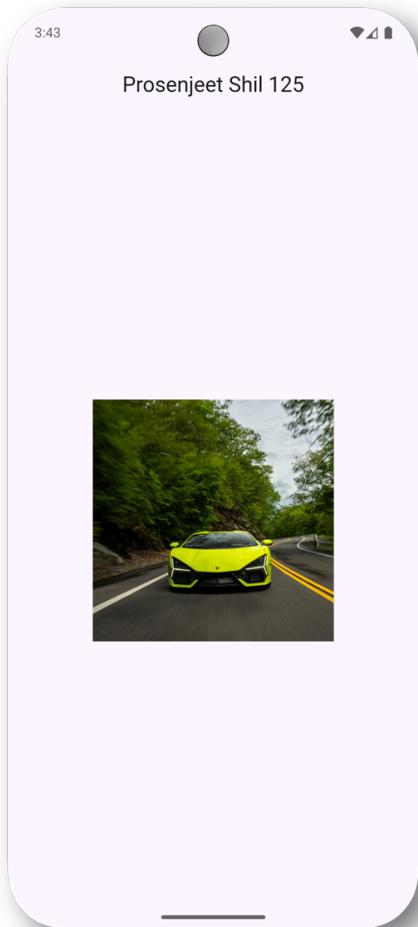
Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Prosenjeet Shil 125'),
          centerTitle: true,
        ),
        body: Center(
          child: Image.network(
            'https://img.freepik.com/free-photo/landscape-with-lake-sunset_395237-259.jpg?semt=ais_hybrid&w=740&q=80',
            width: 250,
            height: 250,
            fit: BoxFit.cover,
          ),
        ),
      );
  }
}
```

Output:



Practical No: 22

Aim: Create a Flutter Application to Display a form.

Code:

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Form Example',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: const MyForm(),
    );
  }
}

class MyForm extends StatefulWidget {
  const MyForm({super.key});

  @override
  _MyFormState createState() => _MyFormState();
}

class _MyFormState extends State<MyForm> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  String _name = "";
  String _email = "";

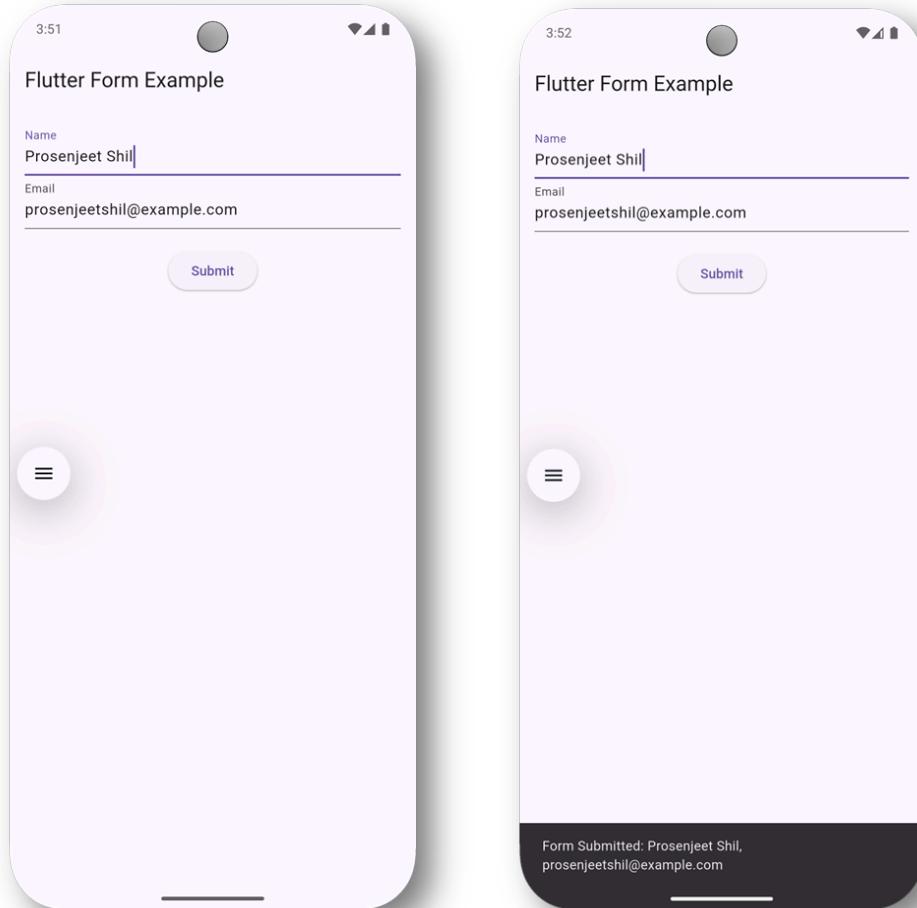
  void _submitForm() {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      print('Name: $_name');
      print('Email: $_email');
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Form Submitted: $_name, $_email')),
      );
    }
  }
}
```

```
    );
  }
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Flutter Form Example'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          children: <Widget>[
            TextFormField(
              decoration: const InputDecoration(labelText: 'Name'),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your name.';
                }
                return null;
              },
              onSaved: (value) {
                _name = value!;
              },
            ),
            TextFormField(
              decoration: const InputDecoration(labelText: 'Email'),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your email.';
                }
                if (!RegExp(r'\S+@\S+\.\S+').hasMatch(value)) {
                  return 'Please enter a valid email.';
                }
                return null;
              },
              onSaved: (value) {
                _email = value!;
              },
            ),
            const SizedBox(height: 20.0),
            ElevatedButton(
              onPressed: _submitForm,
              child: const Text('Submit'),
            ),
          ],
        ),
      ),
    ),
  );
}
```

```
        ),  
        ],  
        ),  
        ),  
        );  
    }  
}
```

Output:



Practical No: 23

Aim: Flutter Program based on Stateful and Stateless Widgets.

Code:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

// StatelessWidget: The root of the application
class MyApp extends StatelessWidget {
  const MyApp({super.key});

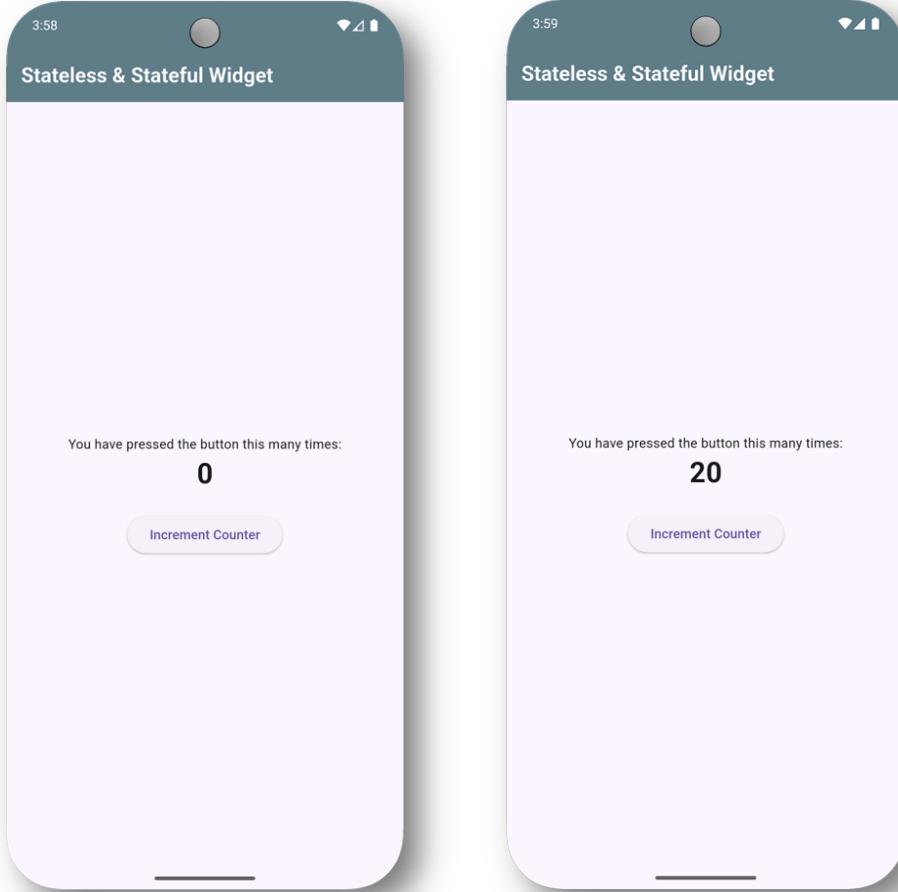
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Stateless vs Stateful Demo',
      home: const HomePage(),
    );
  }
}

// StatelessWidget: Represents the UI layout
class HomePage extends StatelessWidget {
  const HomePage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          'Stateless & Stateful Widget',
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.white,
          ),
        ),
        backgroundColor: Colors.orange,
      ),
      body: const Center(
        child: CounterWidget(), // Our StatefulWidget
      ),
    );
  }
}
```

```
        ),  
    );  
}  
}  
  
// StatefulWidget: Holds the state (e.g., counter)  
class CounterWidget extends StatefulWidget {  
    const CounterWidget({super.key});  
  
    @override  
    _CounterWidgetState createState() => _CounterWidgetState();  
}  
  
// The state class that changes over time  
class _CounterWidgetState extends State<CounterWidget> {  
    int _counter = 0;  
  
    void _incrementCounter() {  
        setState(() {  
            _counter++; // Update the state  
        });  
    }  
  
    @override  
    Widget build(BuildContext context) {  
        return Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: [  
                const Text('You have pressed the button this many times:'),  
                Text(  
                    '$_counter',  
                    style: const TextStyle(  
                        fontSize: 30,  
                        fontWeight: FontWeight.bold,  
                    ),  
                ),  
                const SizedBox(height: 20),  
                ElevatedButton(  
                    onPressed: _incrementCounter,  
                    child: const Text('Increment Counter'),  
                ),  
                ],  
            );  
    }  
}
```

Output:



Practical No: 24

Aim: Flutter Program using List.

Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'List Example',
      theme: ThemeData(primarySwatch: Colors.green),
      home: const FruitListScreen(),
    );
  }
}

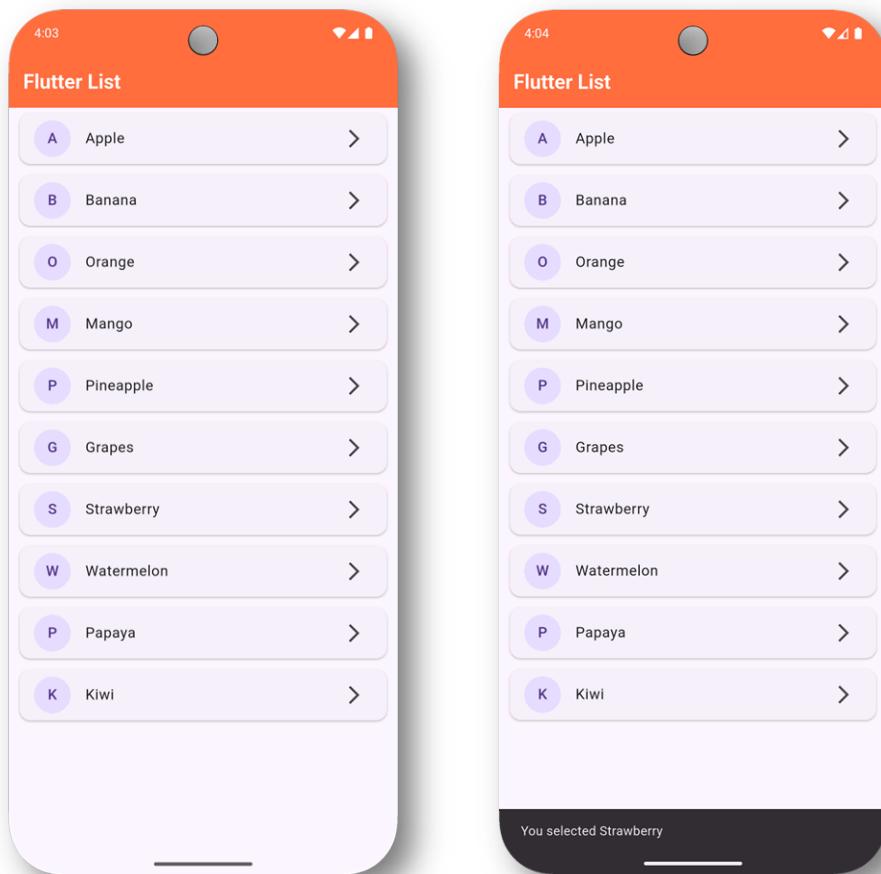
class FruitListScreen extends StatelessWidget {
  const FruitListScreen({super.key});

  final List<String> fruits = const [
    'Apple',
    'Banana',
    'Orange',
    'Mango',
    'Pineapple',
    'Grapes',
    'Strawberry',
    'Watermelon',
    'Papaya',
    'Kiwi',
  ];
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
```

```
title: const Text(  
  'Flutter List',  
  style: TextStyle(  
    fontWeight: FontWeight.bold,  
    color: Colors.white,  
  ),  
,  
  backgroundColor: Colors.lightBlue,  
,  
body: ListView.builder(  
  itemCount: fruits.length,  
  itemBuilder: (context, index) {  
    return Card(  
      margin: const EdgeInsets.symmetric(vertical: 6, horizontal: 12),  
      child: ListTile(  
        leading: CircleAvatar(  
          child: Text(fruits[index][0]),  
        ),  
        title: Text(fruits[index]),  
        trailing: const Icon(Icons.arrow_forward_ios),  
        onTap: () {  
          ScaffoldMessenger.of(context).showSnackBar(  
            SnackBar(content: Text('You selected ${fruits[index]}')),  
          );  
        },  
      );  
    },  
  );  
}
```

Output:



Practical No: 25

Aim: Flutter Program using TextField, Check Box, Buttons, Drop down, Switch etc.

Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Form Example',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: const FormScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class FormScreen extends StatefulWidget {
  const FormScreen({super.key});

  @override
  _FormScreenState createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> {
  String name = "";
  bool agreeToTerms = false;
  bool notificationsEnabled = false;
  String selectedGender = 'Male';
  String result = "";
  final TextEditingController nameController = TextEditingController();

  final List<String> genderOptions = ['Male', 'Female', 'Other'];

  void handleSubmit() {
    setState(() {
      name = nameController.text;
    });
  }
}
```

```
result =  
    'Name: $name\nGender: $selectedGender\nAgreed to Terms: ${agreeToTerms ? "Yes" :  
"No"}\nNotifications: ${notificationsEnabled ? "Enabled" : "Disabled"}';  
});  
}  
  
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: const Text(  
                'User Info Form',  
                style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),  
            ),  
            backgroundColor: Colors.lightBlue,  
        ),  
        body: SingleChildScrollView(  
            padding: const EdgeInsets.all(16.0),  
            child: Column(  
                crossAxisAlignment: CrossAxisAlignment.start,  
                children: [  
                    TextField(  
                        controller: nameController,  
                        decoration: const InputDecoration(  
                            labelText: 'Enter your name',  
                            border: OutlineInputBorder(),  
                        ),  
                    ),  
                    const SizedBox(height: 16),  
                    DropdownButtonFormField<String>(  
                        value: selectedGender,  
                        items: genderOptions  
                        .map(  
                            (gender) => DropdownMenuItem(  
                                value: gender,  
                                child: Text(gender),  
                            ),  
                        )  
                        .toList(),  
                        onChanged: (value) {  
                            setState(() {  
                                selectedGender = value!;  
                            });  
                        },  
                        decoration: const InputDecoration(  
                            labelText: 'Select Gender',  
                            border: OutlineInputBorder(),  
                        ),  
                    ),  
                ],  
            ),  
        ),  
    );  
}
```

```
        ),
        ),
        const SizedBox(height: 16),
        CheckboxListTile(
            title: const Text('I agree to the terms and conditions'),
            value: agreeToTerms,
            onChanged: (value) {
                setState(() {
                    agreeToTerms = value!;
                });
            },
        ),
        SwitchListTile(
            title: const Text('Enable Notifications'),
            value: notificationsEnabled,
            onChanged: (value) {
                setState(() {
                    notificationsEnabled = value;
                });
            },
        ),
        Center(
            child: ElevatedButton(
                onPressed: handleSubmit,
                child: const Text('Submit'),
            ),
        ),
        const SizedBox(height: 20),
        if (result.isNotEmpty)
            Text(
                result,
                style:
                    const TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            ),
        ],
    ),
);
}
}
```

Output:

The image displays two side-by-side screenshots of a mobile application interface, both titled "User Info Form".

Left Screenshot:

- Text input field: "Enter your name" - "Prosenjeet Shil"
- Select dropdown: "Select Gender" - "Male"
- Text input field: "I agree to the terms and conditions" - checked (indicated by a blue checkmark)
- Switch: "Enable Notifications" - On (blue switch)
- Submit button: A white button labeled "Submit" with a blue outline.

Right Screenshot:

- Text input field: "Enter your name" - "Prosenjeet Shil"
- Select dropdown: "Select Gender" - "Male"
- Text input field: "I agree to the terms and conditions" - checked (indicated by a blue checkmark)
- Switch: "Enable Notifications" - On (blue switch)
- Submitted Data Summary:
 - Name: Prosenjeet Shil
 - Gender: Male
 - Agreed to Terms: Yes
 - Notifications: Enabled
- Submit button: A white button labeled "Submit" with a blue outline.

Practical No: 26

Aim: Program to demonstrate the use of SQFlite Database using flutter.

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'database_helper.dart';
import 'user.dart';

void main() {
    WidgetsFlutterBinding.ensureInitialized();
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            title: 'Flutter SQFlite Demo',
            theme: ThemeData(primarySwatch: Colors.blue),
            home: const UserListScreen(),
        );
    }
}

class UserListScreen extends StatefulWidget {
    const UserListScreen({super.key});

    @override
    State<UserListScreen> createState() => _UserListScreenState();
}

class _UserListScreenState extends State<UserListScreen> {
    final DatabaseHelper _dbHelper = DatabaseHelper();
    List<User> _users = [];

    @override
    void initState() {
        super.initState();
```

```
_loadUsers();  
}  
  
Future<void> _loadUsers() async {  
    final users = await _dbHelper.getUsers();  
    setState(() {  
        _users = users;  
    });  
}  
  
Future<void> _addUser() async {  
    final newUser = User(  
        name: 'New User ${DateTime.now().millisecond}',  
        age: 30,  
    );  
    await _dbHelper.insertUser(newUser);  
    _loadUsers();  
}  
  
Future<void> _updateUser(User user) async {  
    final updatedUser = User(  
        id: user.id,  
        name: '${user.name} (Updated)',  
        age: user.age + 1,  
    );  
    await _dbHelper.updateUser(updatedUser);  
    _loadUsers();  
}  
  
Future<void> _deleteUser(int id) async {  
    await _dbHelper.deleteUser(id);  
    _loadUsers();  
}  
  
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: const Text('Flutter SQFlite User List'),  
            backgroundColor: Colors.lightBlue,  
            actions: [  
                IconButton(  
                    icon: const Icon(Icons.add),  
                    onPressed: _addUser,  
                ),  
            ],  
        ),  
    );  
}
```

```
body: _users.isEmpty
? const Center(
  child: Text(
    "No users found. Tap + to add one.",
    style: TextStyle(fontSize: 16),
  ),
)
: ListView.builder(
  itemCount: _users.length,
  itemBuilder: (context, index) {
    final user = _users[index];
    return Card(
      margin: const EdgeInsets.symmetric(
        horizontal: 12, vertical: 6),
      child: ListTile(
        title: Text(user.name),
        subtitle: Text('Age: ${user.age}'),
        trailing: Row(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            IconButton(
              icon: const Icon(Icons.edit, color: Colors.orange),
              onPressed: () => _updateUser(user),
            ),
            IconButton(
              icon: const Icon(Icons.delete, color: Colors.red),
              onPressed: () => _deleteUser(user.id!),
            ),
          ],
        ),
      );
    );
  },
);
```

user.dart

```
class User {
  final int? id;
  final String name;
  final int age;

  User({this.id, required this.name, required this.age});
```

```
Map<String, dynamic> toMap() {
    return {
        'id': id,
        'name': name,
        'age': age,
    };
}

factory User.fromMap(Map<String, dynamic> map) {
    return User(
        id: map['id'],
        name: map['name'],
        age: map['age'],
    );
}

@Override
String toString() => 'User{id: $id, name: $name, age: $age}';
}
```

database_helper.dart

```
import 'dart:async';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'user.dart';

class DatabaseHelper {
    static final DatabaseHelper _instance = DatabaseHelper._internal();
    static Database? _database;

    factory DatabaseHelper() => _instance;

    DatabaseHelper._internal();

    Future<Database> get database async {
        _database ??= await _initDatabase();
        return _database!;
    }

    Future<Database> _initDatabase() async {
        final dbPath = await getDatabasesPath();
        final path = join(dbPath, 'user_database.db');

        return await openDatabase(
```

```
path,
version: 1,
onCreate: (db, version) async {
  await db.execute(
    ...
    CREATE TABLE users(
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT,
      age INTEGER
    )
    ...
  );
},
);
}

// Insert a user
Future<int> insertUser(User user) async {
  final db = await database;
  return await db.insert(
    'users',
    user.toMap(),
    conflictAlgorithm: ConflictAlgorithm.replace,
  );
}

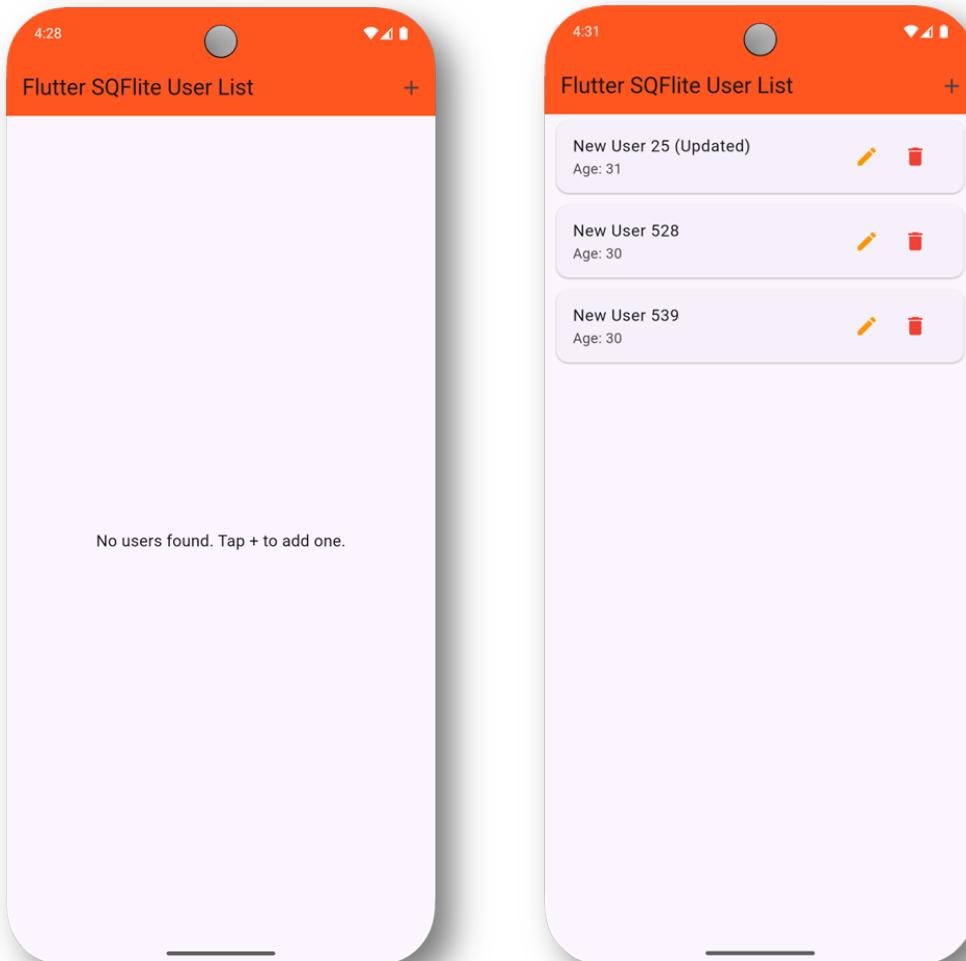
// Retrieve all users
Future<List<User>> getUsers() async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query('users');
  return List.generate(maps.length, (i) => User.fromMap(maps[i]));
}

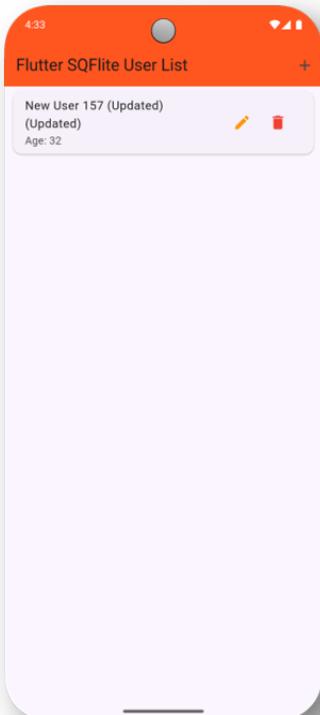
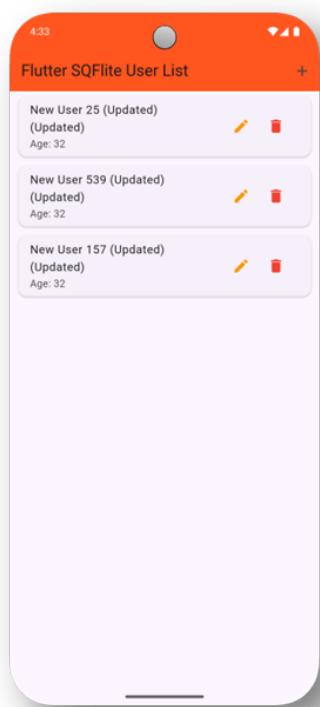
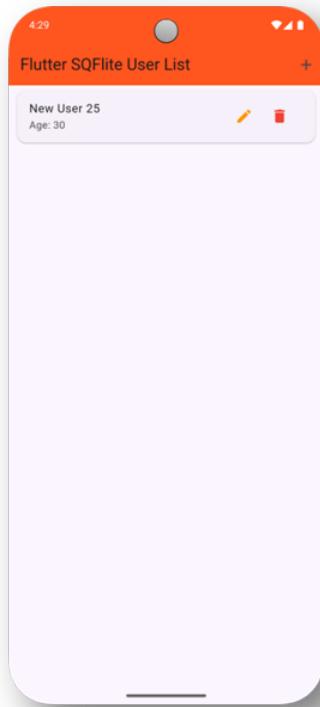
// Update a user
Future<int> updateUser(User user) async {
  final db = await database;
  return await db.update(
    'users',
    user.toMap(),
    where: 'id = ?',
    whereArgs: [user.id],
  );
}

// Delete a user
Future<int> deleteUser(int id) async {
  final db = await database;
```

```
return await db.delete(  
  'users',  
  where: 'id = ?',  
  whereArgs: [id],  
);  
}  
}
```

Output:





Practical No: 27

Aim: Program to demonstrate the use of RESTAPI.

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'post.dart';
import 'post_service.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter REST API',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: const PostsPage(),
    );
  }
}

class PostsPage extends StatefulWidget {
  const PostsPage({super.key});

  @override
  State<PostsPage> createState() => _PostsPageState();
}

class _PostsPageState extends State<PostsPage> {
  late Future<List<Post>> futurePosts;

  @override
  void initState() {
    super.initState();
    futurePosts = fetchPosts();
  }
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Posts')),
    body: FutureBuilder<List<Post>>(
      future: futurePosts,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(child: CircularProgressIndicator());
        } else if (snapshot.hasError) {
          return Center(child: Text(' Error: ${snapshot.error}'));
        } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
          return const Center(child: Text(' No posts found'));
        } else {
          final posts = snapshot.data!;
          return ListView.builder(
            itemCount: posts.length,
            itemBuilder: (context, index) {
              final post = posts[index];
              return Card(
                margin: const EdgeInsets.symmetric(vertical: 6, horizontal: 12),
                child: ListTile(
                  title: Text(
                    post.title,
                    style: const TextStyle(fontWeight: FontWeight.bold),
                  ),
                  subtitle: Text(post.body),
                ),
              );
            },
          );
        }
      },
    );
}
```

post.dart

```
class Post {
  final int userId;
  final int id;
  final String title;
  final String body;
```

```
Post({  
    required this.userId,  
    required this.id,  
    required this.title,  
    required this.body,  
});  
  
factory Post.fromJson(Map<String, dynamic> json) {  
    return Post(  
        userId: json['userId'],  
        id: json['id'],  
        title: json['title'],  
        body: json['body'],  
    );  
}  
}  
}
```

post_service.dart

```
import 'dart:convert';  
import 'package:http/http.dart' as http;  
import 'post.dart';  
  
Future<List<Post>> fetchPosts() async {  
    try {  
        final response = await http  
            .get(  
                Uri.parse('https://jsonplaceholder.typicode.com/posts'),  
                headers: {'Content-Type': 'application/json'},  
            )  
            .timeout(const Duration(seconds: 10));  
  
        if (response.statusCode == 200) {  
            final List<dynamic> jsonData = json.decode(response.body);  
            return jsonData.map((item) => Post.fromJson(item)).toList();  
        } else {  
            throw Exception(  
                'Failed to load posts (Code: ${response.statusCode})',  
            );  
        }  
    } catch (e) {  
        throw Exception('Error fetching posts: $e');  
    }  
}
```

Output:

