EX 3b	Layout Switching, Form Submission and Feedback Using Android's XML Layouts

Aim

To demonstrate layout switching, form submission, and feedback using Android's XML layouts.

Definitions

Layout Switching

Layout switching is the process of changing the visible user interface on the screen while the application is running, typically in response to user interaction.

Form Submission

Form submission refers to the mechanism by which user-inputted data in a layout (e.g., username, password) is gathered and sent to the application logic for processing.

Feedback

Feedback is the visual, auditory, or haptic response provided to the user after they have interacted with the application, ensuring they know their action was registered.

Android XML Layout

An Android XML layout is a file written in Extensible Markup Language (XML) that defines the structure and design for a user interface in an Android application. These files describe the visual components such as buttons, text fields, and images and their arrangement within the app screen

Procedure

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

MainActivity.java

```
package xmllayouts.com.xmllayouts;


    import android.app.Activity;
    import android.content.Intent;
    import android.os.Bundle;
    import android.view.View;
    import android.widget.Button;
    import android.widget.EditText;
    import android.widget.Toast;

public class MainActivity extends Activity { // Extend android.app.Activity

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

    final EditText nameField = (EditText) findViewById(R.id.edit_text_name);
    final EditText emailField = (EditText) findViewById(R.id.edit_text_email);
    Button submitButton = (Button) findViewById(R.id.button_submit);

    submitButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String name = nameField.getText().toString().trim();
        String email = emailField.getText().toString().trim();

        if (!name.isEmpty() && !email.isEmpty()) {
          // Create an explicit Intent to switch to FeedbackActivity
          Intent intent = new Intent(MainActivity.this, FeedbackActivity.class);
          // Pass data to the next activity
          intent.putExtra("EXTRA_NAME", name);
          intent.putExtra("EXTRA_EMAIL", email);
          startActivity(intent); // Start the new activity
        } else {
          // Show feedback as a Toast message for invalid input
          Toast.makeText(MainActivity.this, "Please fill in all fields.",
```

```
Toast.LENGTH_SHORT).show();
            }
        }
    });
  }
}
```

3. Right click on xmllayouts.com. xmllayouts package → New → Java Class → Name: FeedbackActivity → ok. Type the following codes in FeedbackActivity.java,

FeedbackActivity.java

```
package xmllayouts.com.xmllayouts;


    import android.app.Activity;
    import android.os.Bundle;
    import android.view.View;
    import android.widget.Button;
    import android.widget.TextView;

public class FeedbackActivity extends Activity { // Extend android.app.Activity

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_feedback); // Load the feedback layout

    TextView feedbackTextView = (TextView) findViewById(R.id.text_view_feedback);
    Button backButton = (Button) findViewById(R.id.button_back);

    // Get data passed from the previous activity via Intent
    Bundle extras = getIntent().getExtras();
    if (extras != null) {
      String name = extras.getString("EXTRA_NAME");
      String email = extras.getString("EXTRA_EMAIL");
      String feedbackMessage = "Thank you, " + name + "! Your submission with email
" + email + " has been received.";
      feedbackTextView.setText(feedbackMessage);
    }

    backButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        finish(); // Close this activity and return to the previous one
      }
    });
  }
}
```

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

activity_main.xml

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

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Name:" />

   <EditText
      android:id="@+id/edit_text_name"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Enter your name" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Email:" />

   <EditText
      android:id="@+id/edit_text_email"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Enter your email" />

   <Button
      android:id="@+id/button_submit"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Submit" />

</LinearLayout>
```

5. Right click on Layout folder → New → Layout Resource File → File Name: activity_feedback → ok. Type the following codes in activity_feedback.xml,

activity_feedback.xml

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

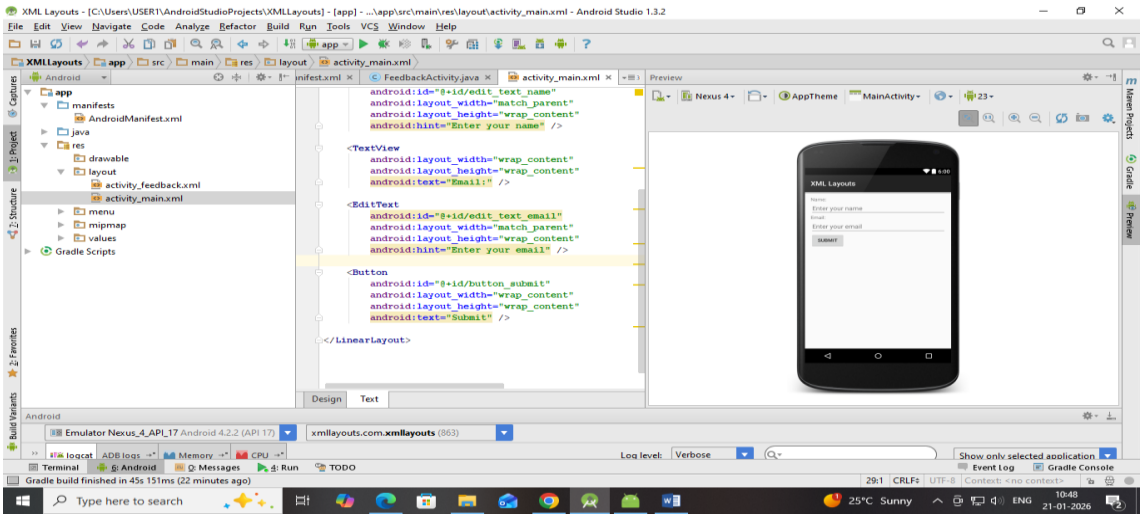
```
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/text_view_feedback"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="Feedback details will appear here." />

    <Button
        android:id="@+id/button_back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go Back" />

</LinearLayout>
```

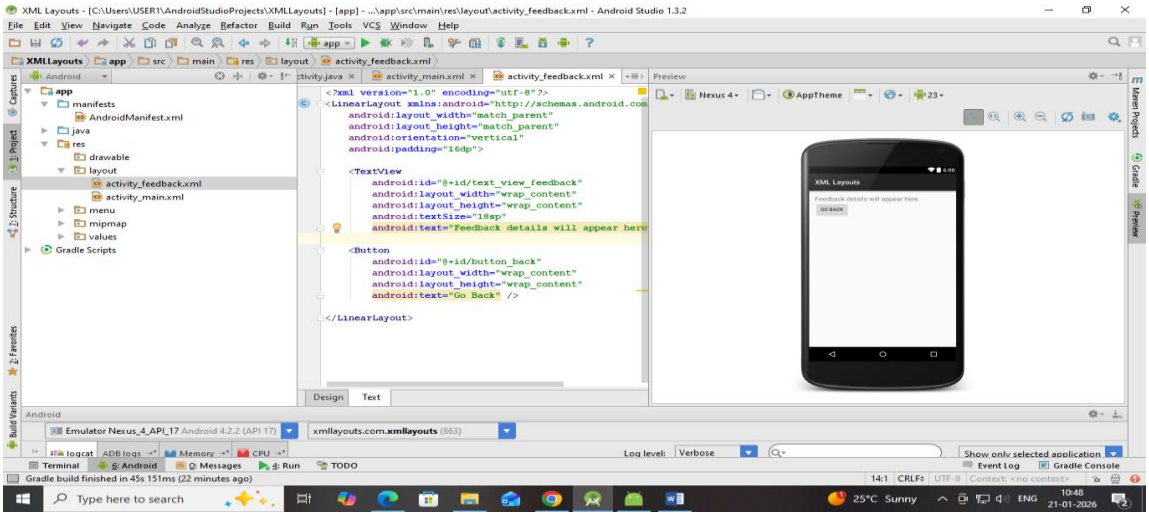
6. Open AndroidManifest.xml under app/manifests and type the following codes to add activity in the manifest file,

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="xmllayouts.com.xmllayouts" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".FeedbackActivity" />
    </application>

</manifest>
```

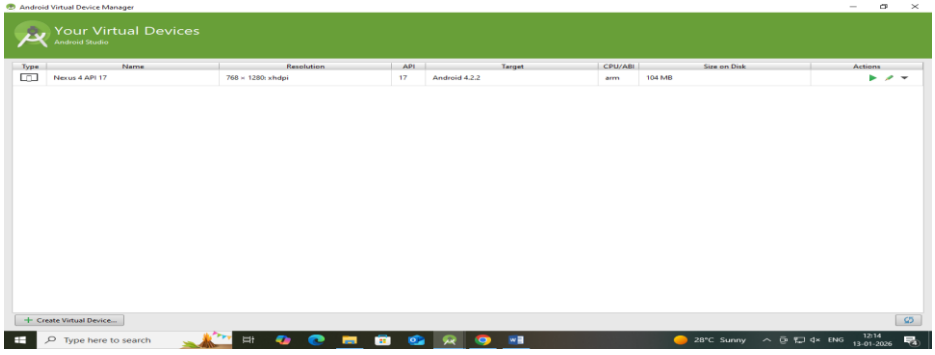
7. The design of the application will be as follows:





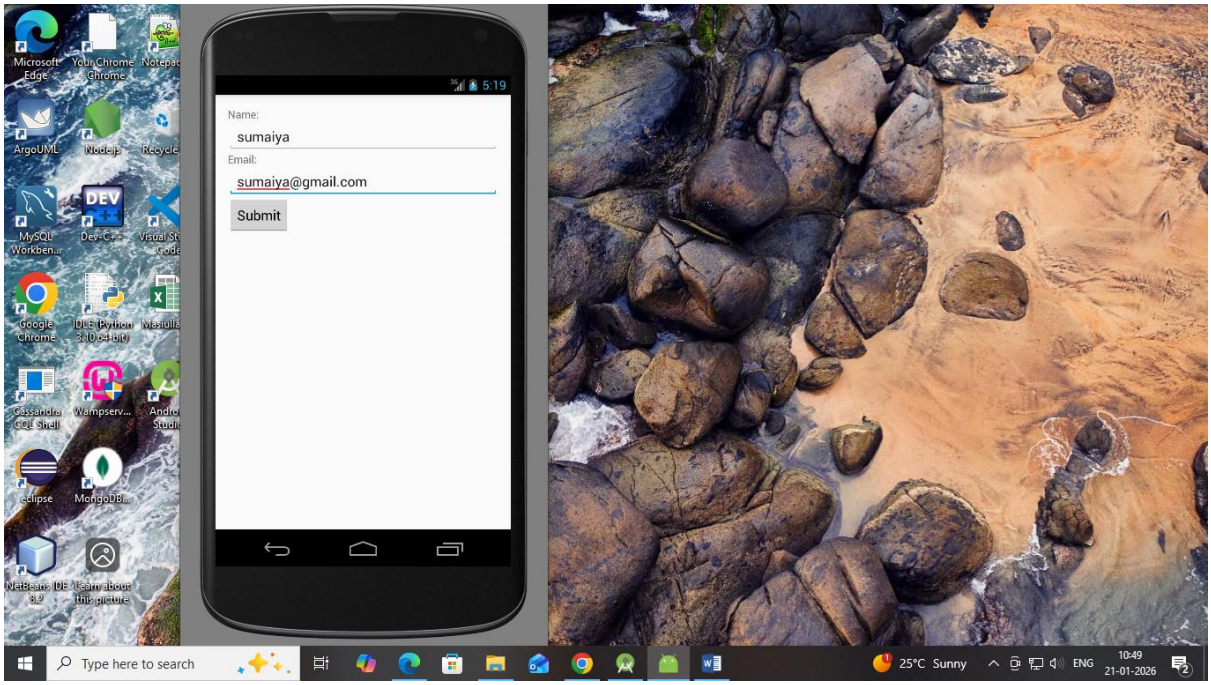
8. Go to Tools → android → AVD Manager → click "+ create a virtual device" → select "phone" from category → select "Nexus 4" from the list → click "next" → select Release name: Jelly Bean, API Level: 17, ABI: armeabi-v7a, Target: Android 4.2.2 from the list → click "next" → Choose orientation "portrait"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

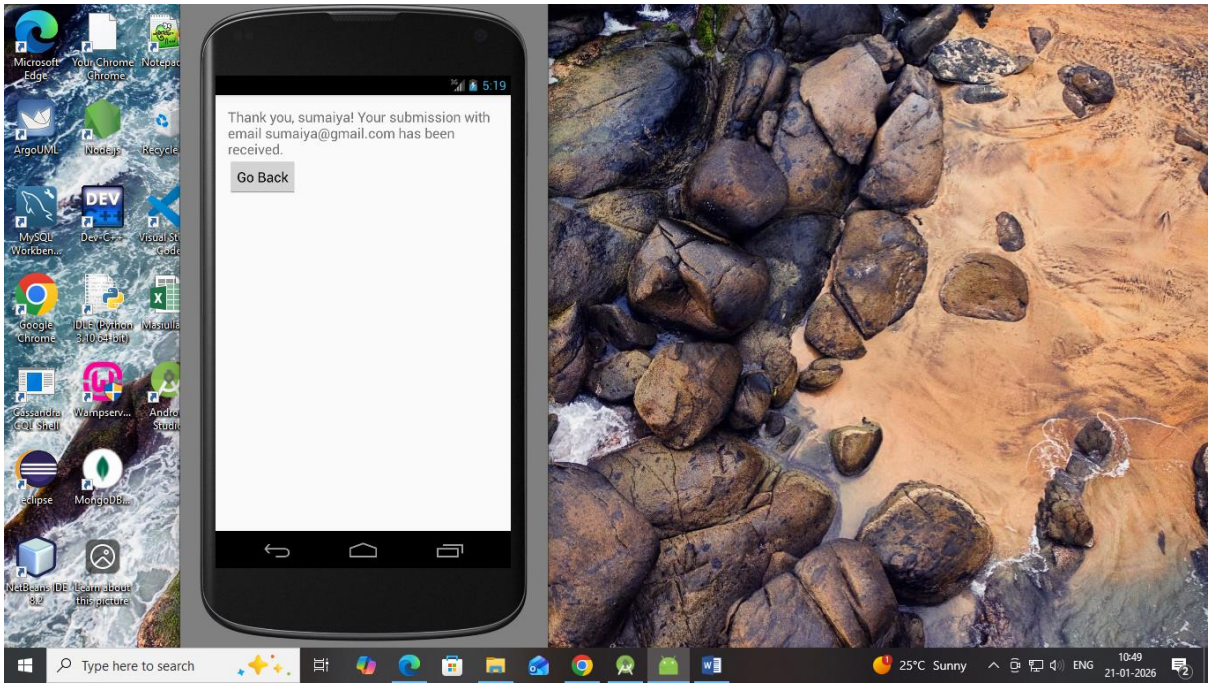
The following window will appear after configuring AVD:



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

Output





Result

Thus, layout switching, form submission, and feedback have been successfully demonstrated using Android’s XML layouts.