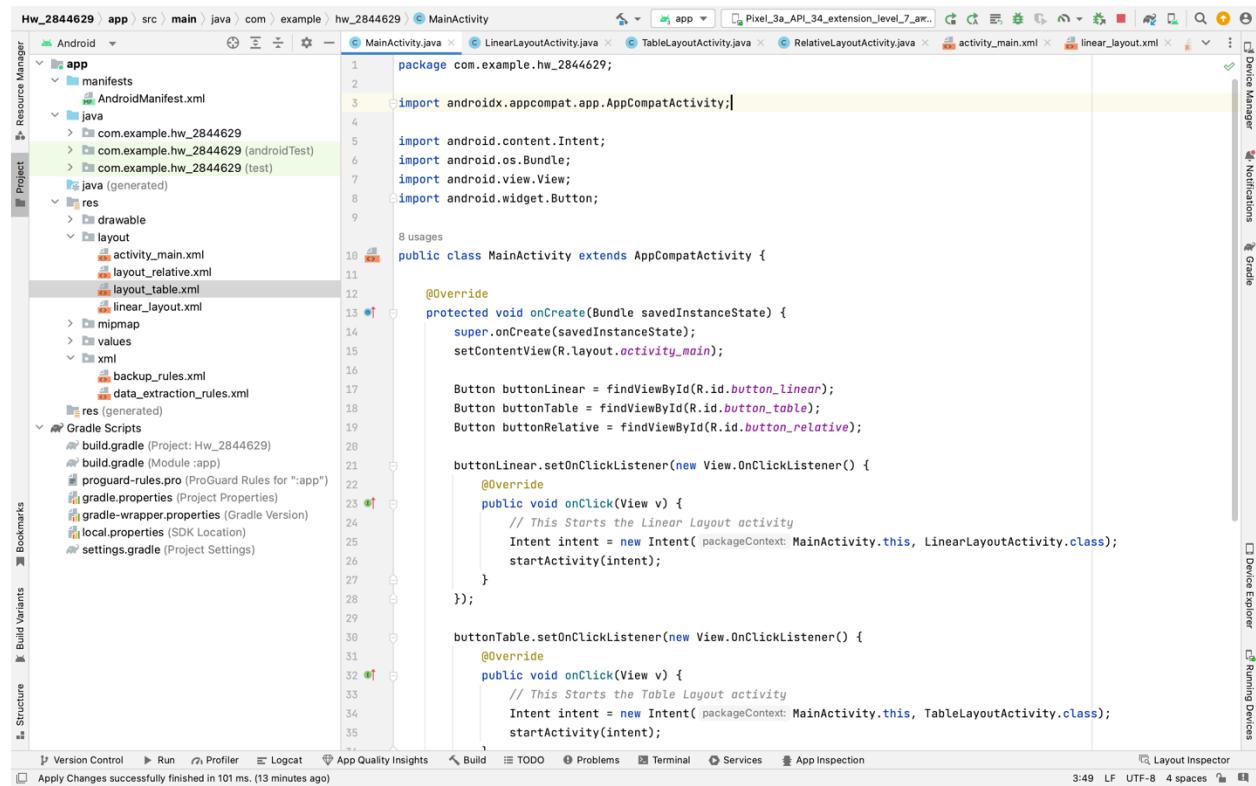


Create an app that showcases different types of layouts:

- In the main activity, add three buttons.
- Button-1: on click, it should start another activity with a linear layout as shown.
- Button-2: on click, it should start another activity with a table layout as shown.
- Button-3: on click, it should start another activity with a relative layout as shown.

Note: I also added a back button, so we don't have to close the application and run again for running three different layouts. I am using "OnClickListener" to use back button which when clicked will take back to the MainActivity from any of the three layouts(Linear Layout, Table layout, Relative Layout),which is a straightforward method for handing button clicks.

MainActivity.java : In this activity the three buttons “buttonLinear”, “buttonTable”, “buttonrelative” are initialized. Each of these buttons is associated with the method “OnClickListener”. An Intent is created to start a new activity when the button is clicked. The “LinearLayoutActivity” is started when the “buttonLinear” is clicked, same way “TableLayoutActivity”, “RelativeLayoutActivity” are started when “buttonTable”, “buttonRelative” is clicked. The “intent” calls take the user from the main activity to the selected layout activity, demonstrating basic activity switching and navigation in Android.



The screenshot shows the Android Studio interface with the project 'Hw_2844629' open. The Project tool window on the left shows the directory structure: app > manifests, java, res, layout, values, xml, and Gradle Scripts. The Java tool window on the right displays the MainActivity.java code. The code initializes three buttons (buttonLinear, buttonTable, buttonRelative) and sets their onClick listeners to start different activities (LinearLayoutActivity, TableLayoutActivity, and RelativeLayoutActivity) using Intent objects. The code is annotated with comments explaining the logic for each button's functionality.

```

package com.example.hw_2844629;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

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

        Button buttonLinear = findViewById(R.id.button_linear);
        Button buttonTable = findViewById(R.id.button_table);
        Button buttonRelative = findViewById(R.id.button_relative);

        buttonLinear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // This Starts the Linear Layout activity
                Intent intent = new Intent(MainActivity.this, LinearLayoutActivity.class);
                startActivity(intent);
            }
        });

        buttonTable.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // This Starts the Table Layout activity
                Intent intent = new Intent(MainActivity.this, TableLayoutActivity.class);
                startActivity(intent);
            }
        });

        buttonRelative.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // This Starts the Relative Layout activity
                Intent intent = new Intent(MainActivity.this, RelativeLayoutActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

The screenshot shows the Android Studio interface with the project 'hw_2844629' open. The left sidebar displays the project structure, including the app module with its sub-directories like manifests, java, and res. The main editor area contains the Java code for the MainActivity.java file, which implements click listeners for three different layout types: Linear Layout, Table Layout, and Relative Layout.

```
hw_2844629 app src main java com.example hw_2844629 MainActivity
```

```
    package com.example.hw_2844629;
    import android.os.Bundle;
    import android.view.View;
    import androidx.appcompat.app.AppCompatActivity;
    import androidx.constraintlayout.widget.ConstraintLayout;
    import androidx.core.widget.NestedScrollView;
    import androidx.recyclerview.widget.LinearLayoutManager;
    import androidx.recyclerview.widget.RecyclerView;
    import android.widget.Button;
    import android.widget.TableLayout;
    import android.widget.TableRow;
    import android.widget.TextView;
    import android.widget.Toast;
    import java.util.ArrayList;
    import java.util.List;

    public class MainActivity extends AppCompatActivity {
        NestedScrollView nestedScrollView;
        RecyclerView recyclerView;
        ConstraintLayout constraintLayout;
        Button buttonLinear, buttonTable, buttonRelative;
        TableLayout tableLayout;
        TableRow tableRow;
        TextView textViewTable;
        TextView textViewRelative;
        TextView textViewLinear;
        List<String> list = new ArrayList<>();

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

            nestedScrollView = findViewById(R.id.nestedScrollView);
            recyclerView = findViewById(R.id.recyclerView);
            constraintLayout = findViewById(R.id.constraintLayout);
            buttonLinear = findViewById(R.id.button_linear);
            buttonTable = findViewById(R.id.button_table);
            buttonRelative = findViewById(R.id.button_relative);
            tableLayout = findViewById(R.id.tableLayout);
            tableRow = findViewById(R.id.tableRow);
            textViewTable = findViewById(R.id.textViewTable);
            textViewRelative = findViewById(R.id.textViewRelative);
            textViewLinear = findViewById(R.id.textViewLinear);

            buttonLinear.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    // This Starts the Linear Layout activity
                    Intent intent = new Intent(getApplicationContext(), LinearLayoutActivity.class);
                    startActivity(intent);
                }
            });

            buttonTable.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    // This Starts the Table Layout activity
                    Intent intent = new Intent(getApplicationContext(), TableLayoutActivity.class);
                    startActivity(intent);
                }
            });

            buttonRelative.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    //This Starts the Relative Layout activity
                    Intent intent = new Intent(getApplicationContext(), RelativeLayoutActivity.class);
                    startActivity(intent);
                }
            });
        }
    }
```

MainActivity.xml : It defines the app layout using “ConstraintLayout”. It has three buttons that are equally spaced vertically: “button_linear”, “button_table”, and “button_relative”. Every button contains text describing its function and is configured to spread over the parent's width 0dp with limits to start and end of parent. The buttons are arranged in relation to one another as follows: “button_table” is the second button from the bottom, “button_linear” is the top button, and “button_relative” is the third button from the bottom. A vertically aligned menu of buttons is efficiently created by this arrangement.

The screenshot shows the Android Studio interface with the following details:

- Project Bar:** Hw_2844629 app src main res layout activity_main.xml
- Resource Manager:** Shows the project structure:
 - app: manifests (AndroidManifest.xml), java (com.example.hw_2844629, com.example.hw_2844629 (androidTest), com.example.hw_2844629 (test)), res (drawable, layout (activity_main.xml, layout_relative.xml, layout_table.xml, linear_layout.xml), mipmap, values), xml (backup_rules.xml, data_extraction_rules.xml), res (generated).
 - Gradle Scripts: build.gradle, build.gradle (Module: app), proguard-rules.pro, gradle.properties, gradle-wrapper.properties, local.properties, settings.gradle.
- Code Editor:** The code editor displays the XML for activity_main.xml, which contains three Button elements arranged vertically. Each button has constraints relative to the others and the parent layout.
- Bottom Navigation Bar:** Version Control, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Terminal, Services, App Inspection, Layout Inspector.

```

Hw_2844629 app src main res layout activity_main.xml
15     app:layout_constraintStart_toStartOf="parent"
16     app:layout_constraintEnd_toEndOf="parent"
17     app:layout_constraintHorizontal_bias="0.5"
18     app:layout_constraintVertical_bias="0.3"/>
19
20 <Button
21     android:id="@+id/button_table"
22     android:layout_width="0dp"
23     android:layout_height="wrap_content"
24     android:text="Button2-TL"
25     app:layout_constraintTop_toBottomOf="@+id/button_linear"
26     app:layout_constraintStart_toStartOf="parent"
27     app:layout_constraintEnd_toEndOf="parent"
28     app:layout_constraintHorizontal_bias="0.5"
29     app:layout_constraintVertical_bias="0.1"/>
30
31 <Button
32     android:id="@+id/button_relative"
33     android:layout_width="0dp"
34     android:layout_height="wrap_content"
35     android:text="Button_RL"
36     app:layout_constraintTop_toBottomOf="@+id/button_table"
37     app:layout_constraintStart_toStartOf="parent"
38     app:layout_constraintEnd_toEndOf="parent"
39     app:layout_constraintHorizontal_bias="0.5"
40     app:layout_constraintVertical_bias="0.1"/>
41
42 </androidx.constraintlayout.widget.ConstraintLayout>
43

```

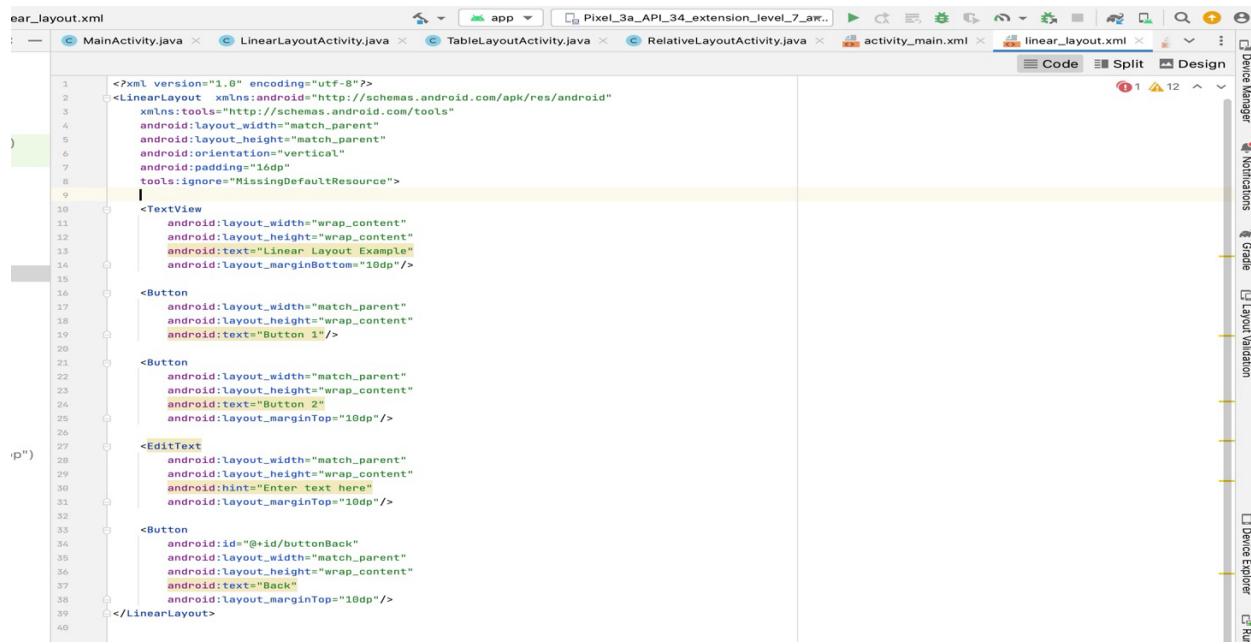
LinearLayoutActivity.java : To build up the activity's user interface, the “onCreate” function is overridden in the “LinearLayoutActivity” class, which extends “AppCompatActivity”. The activity's content view is initially configured to a layout specified in “R.layout.linear_layout” upon creation. The code creates a “OnClickListener” and initializes a “Button” with the ID buttonBack. The application starts a new “Intent” to return to “MainActivity” when the “buttonBack” is clicked. This code shows how to handle button clicks in Android to switch between activities in a basic pattern.

```

Hw_2844629 app src main java com example hw_2844629 MainActivity.java
1 package com.example.hw_2844629;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.widget.Button;
6 import android.view.View;
7
8 import androidx.appcompat.app.AppCompatActivity;
9
10 public class LinearLayoutActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.linear_layout);
16
17         Button buttonBack = findViewById(R.id.buttonBack);
18         buttonBack.setOnClickListener(new View.OnClickListener() {
19             @Override
20             public void onClick(View v) {
21                 Intent intent = new Intent(getApplicationContext(), MainActivity.class);
22                 startActivity(intent);
23             }
24         });
25     }
26 }
27
28

```

Linear_layout.xml : This XML layout uses a vertically oriented “LinearLayout” to define the user interface for an Android activity. After a “TextView” with the text "Linear Layout Example," two generic buttons, a “EditText” for text input, and a "Back" button with the ID “@+id/buttonBack” are included. All buttons and the EditText extend to fill the full width of the parent container, and all elements are stacked vertically with uniform margin spacing.

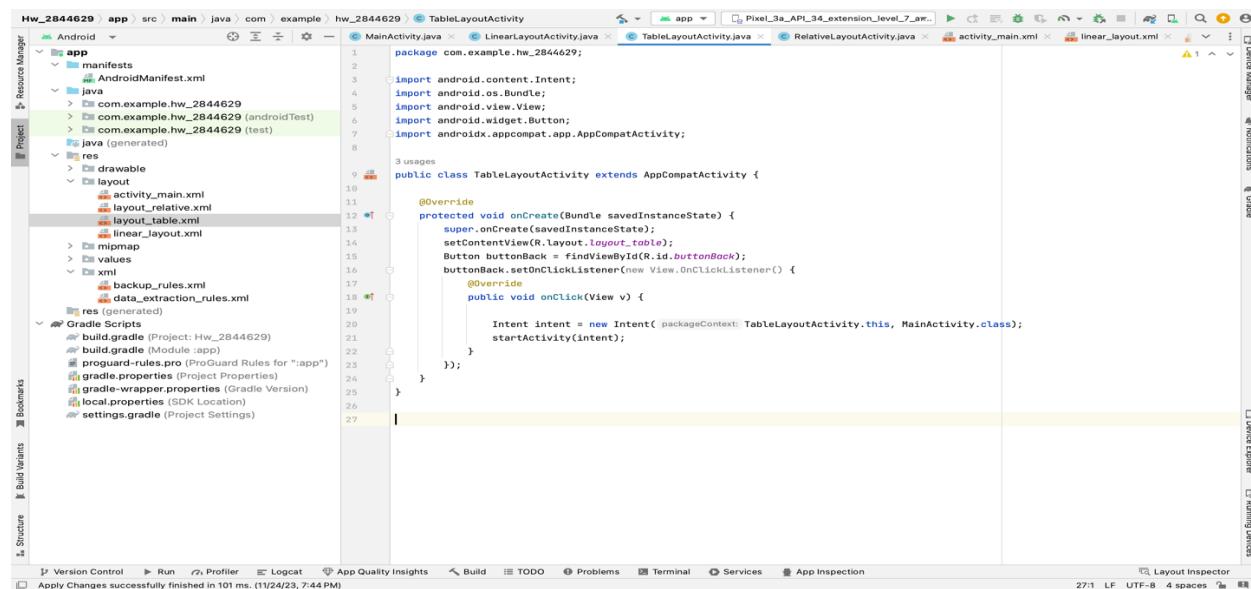


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:padding="16dp"
    tools:ignore="MissingDefaultResource">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Linear Layout Example"
        android:layout_marginBottom="10dp"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button 1"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button 2"
        android:layout_marginTop="10dp"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text here"
        android:layout_marginTop="10dp"/>
    <Button
        android:id="@+id/buttonBack"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Back"
        android:layout_marginTop="10dp"/>
</LinearLayout>

```

TableLayoutActivity.java : The activity's layout is set to "R.layout.layout_table" by the onCreate function of the "TableLayoutActivity" class, which extends "AppCompatActivity". Initialization of a Button with the ID "buttonBack" and setting of a "OnClickListener". This button illustrates basic button interaction and activity navigation in application by creating an intent and using it to return to "MainActivity" when pressed.



```

package com.example.hw_2844629;

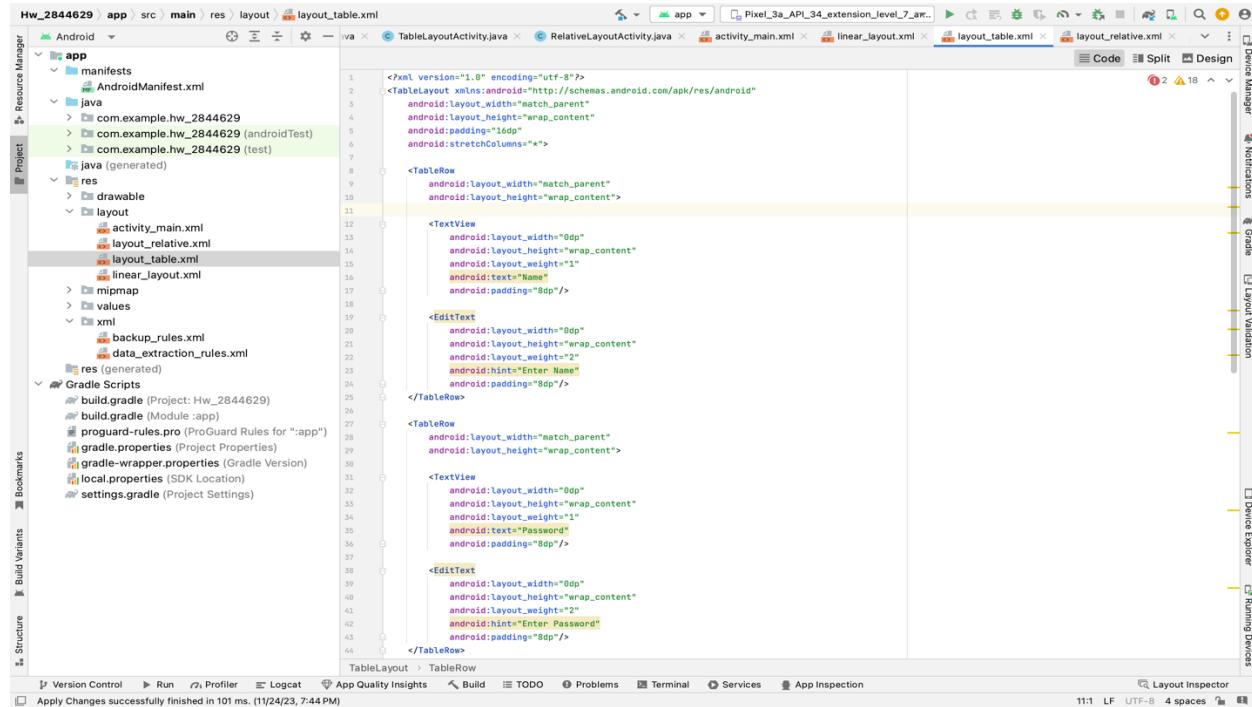
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class TableLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_table);
        Button buttonBack = findViewById(R.id.buttonBack);
        buttonBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

Layout_table.xml : This provides a table row with "TableRow" characteristics for every row, enabling TextView and EditText input to scale proportionately. A Checkbox and two Button elements are also included in the layout; these take up the whole width of the row because of the "layout_span" argument. A form-like user interface is intended for this layout.



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:stretchColumns="*"

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Name"
            android:padding="8dp"/>

        <EditText
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Name"
            android:padding="8dp"/>

    </TableRow>

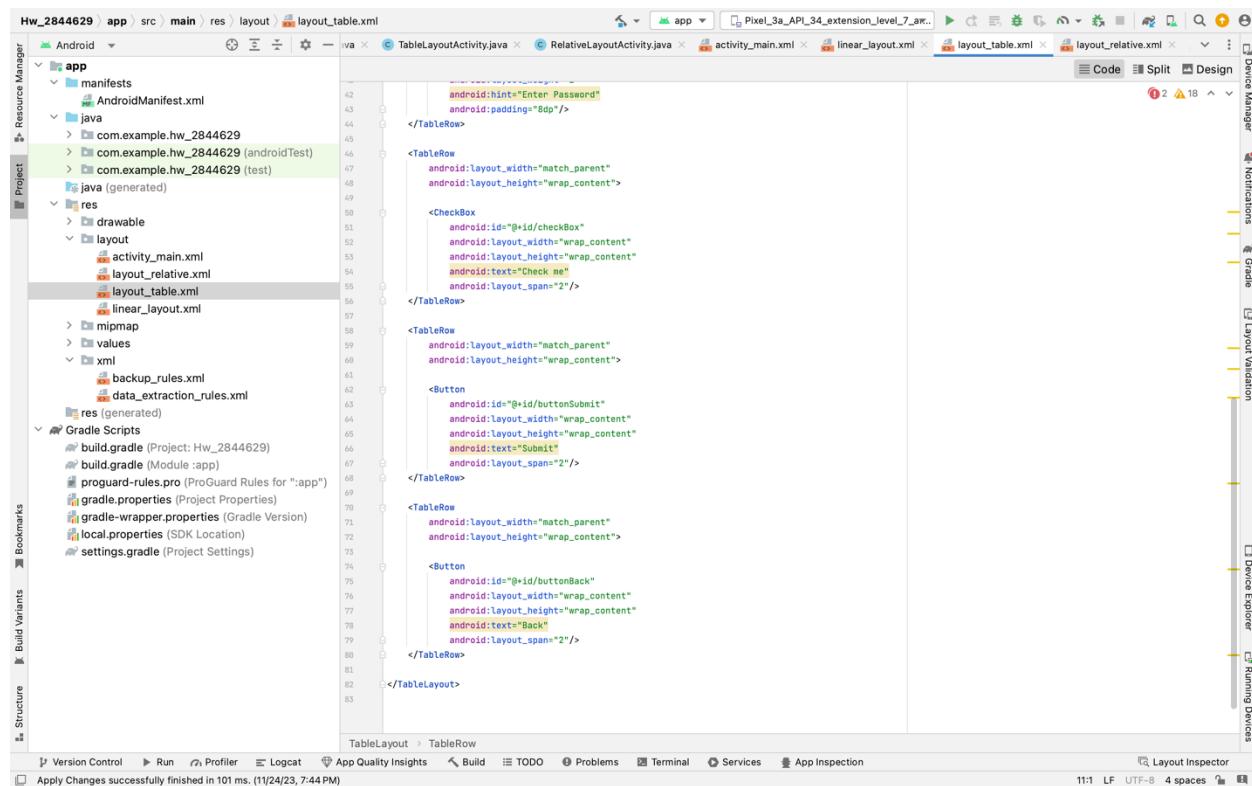
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Password"
            android:padding="8dp"/>

        <EditText
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Password"
            android:padding="8dp"/>

    </TableRow>

```



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:stretchColumns="*"

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="Enter Password"
            android:padding="8dp"/>

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <CheckBox
            android:id="@+id/checkBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Check me"
            android:layout_span="2"/>

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/buttonSubmit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Submit"
            android:layout_span="2"/>

    </TableRow>

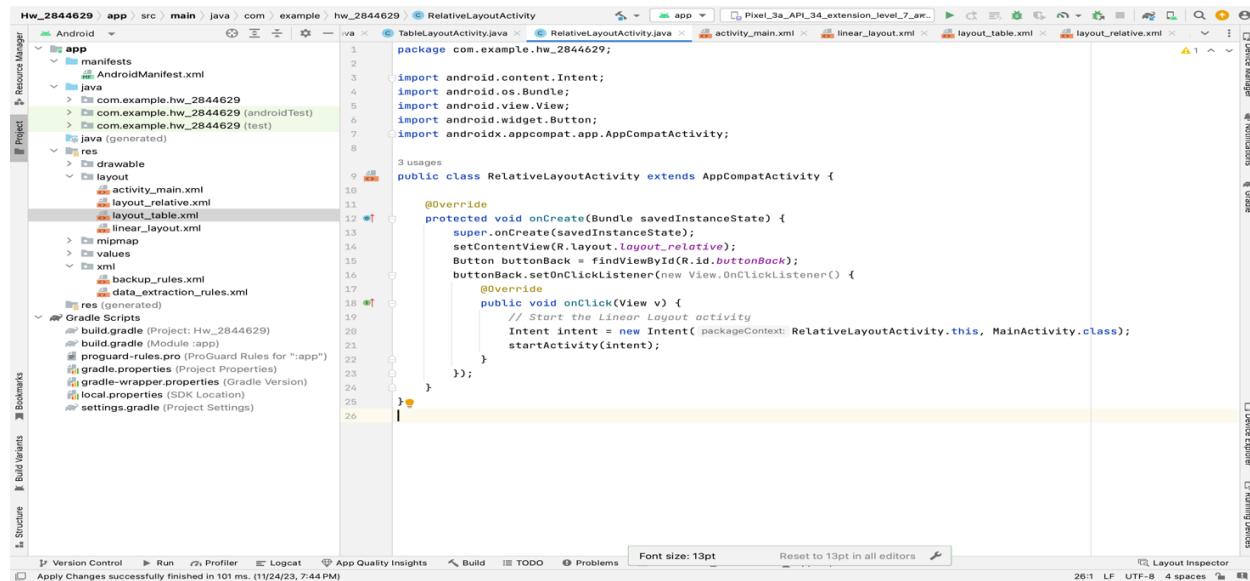
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/buttonBack"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Back"
            android:layout_span="2"/>

    </TableRow>

```

RelativeLayoutActivity.java : This class of an application the code sets up the user interface from “R.layout.layout_relative” upon creation. A Button with the ID "buttonBack" is initialized, and a "OnClickListener" is set on it. A simple pattern for managing button clicks and switching between activities inside the app is shown when the buttonBack is hit, causing the application to establish an intent to return to the "MainActivity".

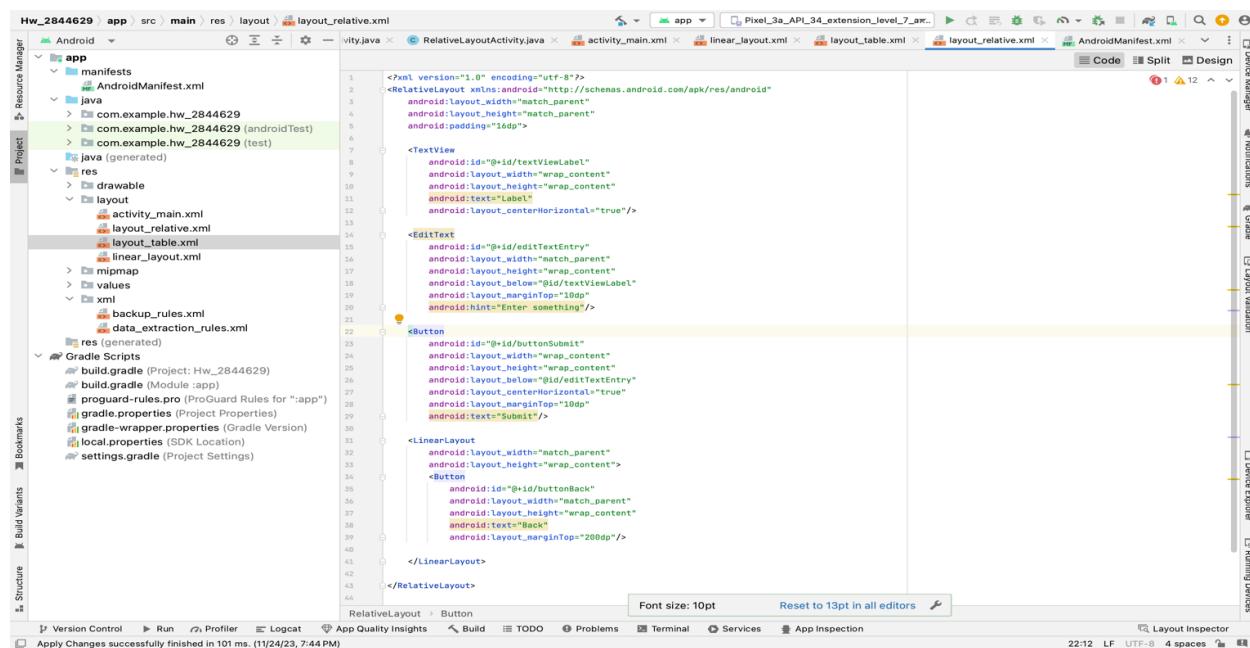


```

1 package com.example.hw_2844629;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import androidx.appcompat.app.AppCompatActivity;
8
9
10 public class RelativeLayoutActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.layout_relative);
16         Button buttonBack = findViewById(R.id.buttonBack);
17         buttonBack.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View v) {
20                 // Start the Linear Layout activity
21                 Intent intent = new Intent(getApplicationContext(), MainActivity.class);
22                 startActivity(intent);
23             }
24         });
25     }
26

```

Layout_relative.xml : the “RelativeLayout” is used as a root to for flexible positioning of the UI elements relative to each other. An "EditText" is positioned directly below a "TextView" that is positioned. The "EditText" is centered, and a Submit button is below it. To further demonstrate the use of both linear and relative layouts for adaptable UI design in an Android application, a LinearLayout with a Back button has been added at the bottom.



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:padding="10dp">
6
7     <TextView
8         android:id="@+id/textViewLabel"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Label"
12        android:layout_centerHorizontal="true"/>
13
14     <EditText
15         android:id="@+id/editTextEntry"
16         android:layout_width="match_parent"
17         android:layout_height="wrap_content"
18         android:layout_below="@+id/textViewLabel"
19         android:layout_marginTop="10dp"
20         android:hint="Enter something"/>
21
22     <Button
23         android:id="@+id/buttonSubmit"
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:layout_below="@+id/editTextEntry"
27         android:layout_centerHorizontal="true"
28         android:layout_marginTop="10dp"
29         android:text="Submit"/>
30
31     <LinearLayout
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content">
34         <Button
35             android:id="@+id/buttonBack"
36             android:layout_width="match_parent"
37             android:layout_height="wrap_content"
38             android:text="Back"
39             android:layout_marginTop="20dp"/>
40     </LinearLayout>
41
42 </RelativeLayout>
43
44

```

Androidmanifest.xml : it defines the basic configuration of the for the application. The "AndroidManifest.xml" file specifies the fundamental settings for an Android application, including four activities: TableLayoutActivity, RelativeLayoutActivity, MainActivity, and LinearLayoutActivity with MainActivity as the entry point.

The screenshot shows the Android Studio interface with the project 'Hw_2844629' open. The left sidebar displays the project structure, including the 'app' module with its sub-directories: 'AndroidManifest.xml', 'java', 'res', and 'gradle'. The 'res/layout' directory contains three XML files: 'activity_main.xml', 'layout_relative.xml', and 'layout_table.xml'. The 'res/xml' directory contains two XML files: 'backup_rules.xml' and 'data_extraction_rules.xml'. The main editor window shows the 'AndroidManifest.xml' file's code:

```

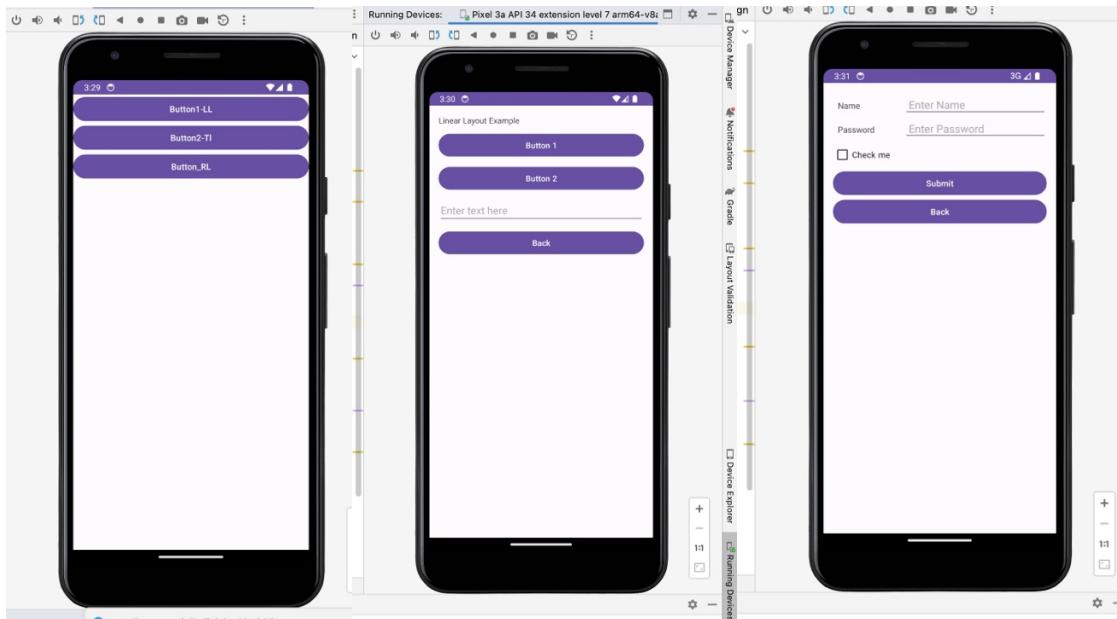
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Hw_2844629"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".LinearLayoutActivity" />
        <activity android:name=".TableLayoutActivity" />
        <activity android:name=".RelativeLayoutActivity" />
    </application>
</manifest>

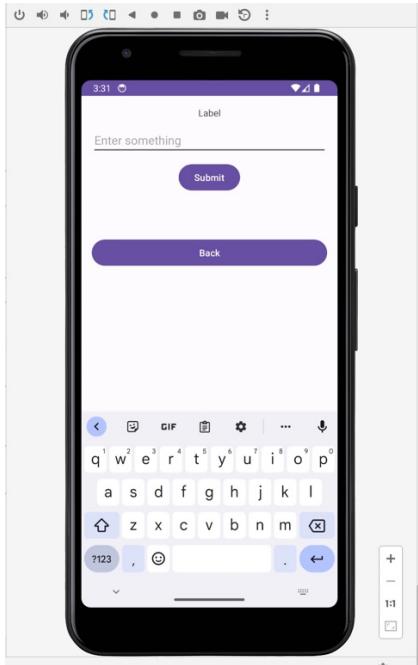
```

The bottom status bar indicates the device is connected to 'Pixel_3a API 34 extension level_7_arm64-v8a [emulator-5554]'.

Below are the screenshots of MainActivity and 3 layouts.



Screenshot 1(MainActivity). Screenshot 2(LinearLayout). Screenshot 3(Table Layout).



Screenshot 4(RelativeLayout).