

Program No:	4
Roll No :	1545
Title of Program :	Linear layout
Objective :	Create a Registration form using Linear layout

## What is the use of linear layout?

**LinearLayout** is a layout container used in Android that arranges its child views either in a single horizontal row or a single vertical column. It's especially useful when you want to display elements sequentially—one after the other—in a straight line. The direction of arrangement is controlled by the `android:orientation` attribute, which can be set to either `"vertical"` (top to bottom) or `"horizontal"` (left to right)

### Key uses:

- Building simple user interfaces where elements follow a clear, linear order.
- Dividing space among child elements flexibly, either equally or proportionally, using weights.
- Quickly stacking UI components, such as buttons, text fields, or images.

## Attributes of Linear Layout

LinearLayout comes with several attributes that allow you to customize how its child views are positioned and sized. Some attributes are applied to the LinearLayout itself, while others are for its child views.

### Common Attributes

Attribute	Description
<code>android:id</code>	Assigns a unique ID to the layout.

<code>android:orientation</code>	Specifies the direction for child elements: " <code>horizontal</code> " or " <code>vertical</code> " (default: <code>horizontal</code> ).
<code>android:layout_width</code>	Sets the width of the layout ( <code>match_parent</code> , <code>wrap_content</code> , or specific dp value).
<code>android:layout_height</code>	Sets the height of the layout.
<code>android:gravity</code>	Sets the alignment of all children within <code>LinearLayout</code> (e.g., <code>center</code> , <code>top</code> , <code>bottom</code> ).
<code>android:weightSum</code>	Sets the total sum of layout weights for child elements.
<code>android:divider</code>	Adds drawable between child views as dividers.
<code>android:baselineAligned</code>	Boolean; aligns children by their text baselines for visual consistency.
<code>android:measureWithLargestChild</code>	If true, children with a weight will be given the size of the largest child.

## Important Child Attributes

Attribute	Description
<code>android:layout_weight</code>	Defines how much of the remaining space a child view should consume relative to siblings.
<code>android:layout_gravity</code>	Sets the alignment of a specific child view within the layout.
<code>android:layout_margin*</code>	Provides margins around views (e.g., <code>layout_marginLeft</code> , <code>layout_marginTop</code> ).

## Source Code:

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textSize="30sp"
        android:text="Registration Form"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:hint="Name"
        android:inputType="text"
        android:textSize="20sp"
        android:layout_marginTop="5dp"
        android:id="@+id/e1"
        />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:hint="Email"
        android:inputType="text"
        android:textSize="20sp"
        android:layout_marginTop="5dp"
        android:id="@+id/e2"
        />

    <Button
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_gravity="center"
        android:id="@+id/submit"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textSize="20sp"
        android:layout_marginTop="15dp"
        android:textColor="@color/black"
        android:id="@+id/t1"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textSize="20sp"
        android:layout_marginTop="15dp"
        android:textColor="@color/black"
        android:id="@+id/t2"
    />
</LinearLayout>
```

## MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

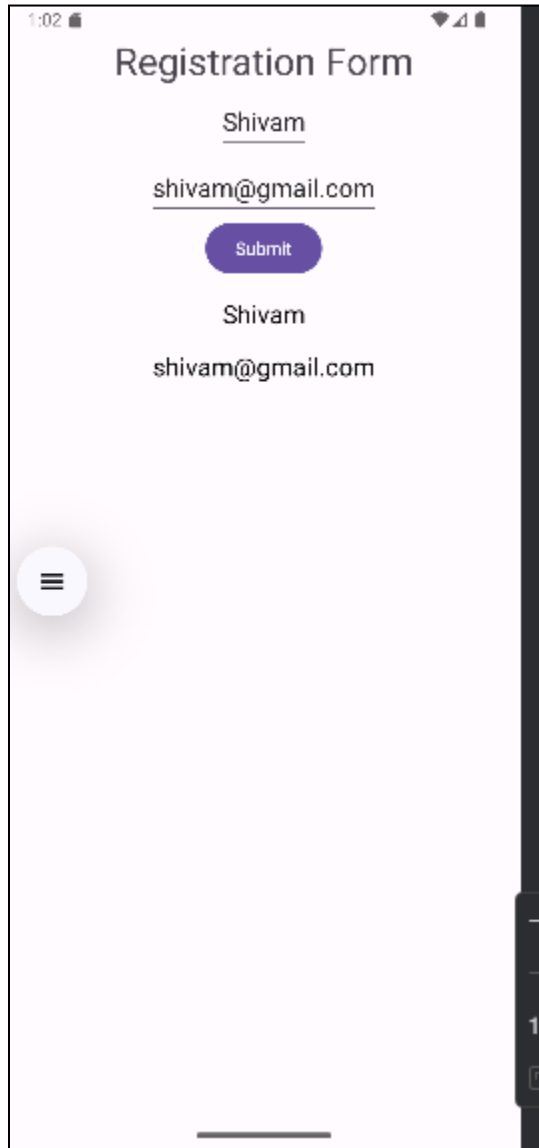
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    Button b1;
    EditText e1, e2;
    TextView t1, t2;

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

```
super.onCreate(savedInstanceState);
EdgeToEdge.enable(this);
setContentView(R.layout.activity_main);
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
    Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
    return insets;
});
b1 = findViewById(R.id.submit);
e1 = findViewById(R.id.e1);
e2 = findViewById(R.id.e2);
t1 = findViewById(R.id.t1);
t2 = findViewById(R.id.t2);

b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String str1 = e1.getText().toString();
        String str2 = e2.getText().toString();
        t1.setText(str1);
        t2.setText(str2);
    }
});
}
```

**Output:**



1:02

### Registration Form

Shivam

shivam@gmail.com

Submit

Shivam

shivam@gmail.com

☰

+

1