Name : Kanchan Adhikrao Jadhav
Roll No : 8239
Subject : Android Development.

Practical -06

Aim:- Create an android application to with interactive user interface different layout.

1.Linear Layout

Linear layout is a simple layout used in android for layout designing. In the Linear layout all the elements are displayed in linear fashion means all the childs/elements of a linear layout are displayed according to its orientation. The value for orientation property can be either horizontal or vertical.

Types Of Linear Layout Orientation

There are two types of linear layout orientation:

1. Vertical
2. Horizontal

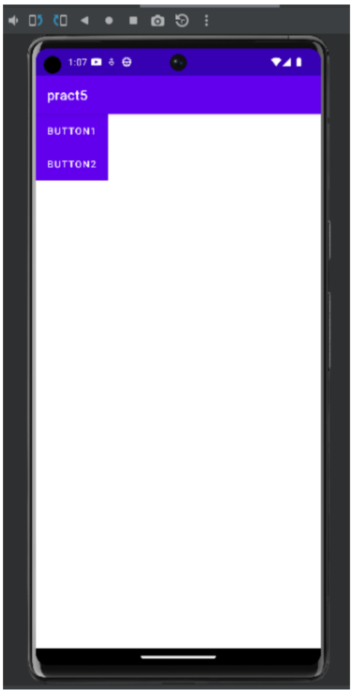
1.Vertical: In this all the child are arranged vertically in a line one after the other. In below code snippets we have specified orientation “vertical” so the childs/views of this layout are displayed vertically.

Activity_main.xml:-

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

android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="vertical"> <!-- Horizontal Orientation set -->

<!-- Child Views(In this case 2 Button) are here -->

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Button1"
   android:id="@+id/button"
   android:background="#358a32" />

<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button2"
    android:id="@+id/button2"
    android:background="#0058b6" />
</LinearLayout>
```

Output :-



2. Horizontal:

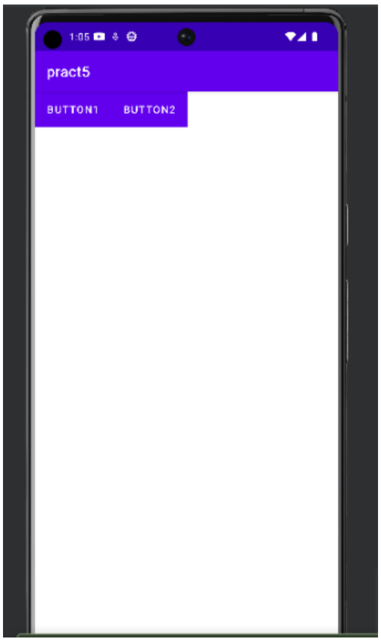
In this all the child are arranged horizontally in a line one after the other. In below code snippets we have specified orientation “horizontal” so the childs/views of this layout are displayed horizontally.

Activity_main.xml:-

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

android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"> <!-- Horizontal Orientation set -->
```

```
<!-- Child Views(In this case 2 Button) are here -->

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Button1"
   android:id="@+id/button"
   android:background="#358a32" />

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Button2"
   android:id="@+id/button2"
   android:background="#0058b6" />
</LinearLayout>
```

Output:-



2.Relative Layout:-

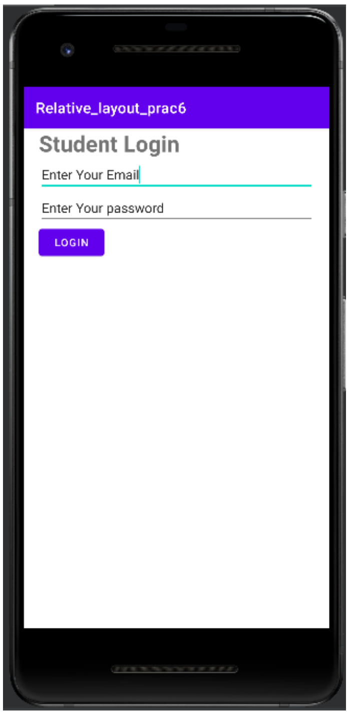
Android RelativeLayout enables you to specify how child views are positioned relative to each other. The position of each view can be specified as relative to sibling elements or relative to the parent.

Name : Kanchan Adhikrao Jadhav
Roll No : 8239
Subject : Android Development.

Activity_main.xml:-

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp">

    <TextView
        android:id="@+id/heading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student Login"
        android:textStyle="bold"
        android:textSize="30dp"
        android:hint="@string/app_name" />
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/heading">
        <EditText
            android:id="@+id/email"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Enter Your Email"
            android:hint="@string/app_name" />
        <EditText
            android:id="@+id/password"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Enter Your password"
            android:hint="@string/app_name" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:id="@+id/button" />
</LinearLayout>
</RelativeLayout>
```

Output :-



Relative_layout_prac6

Student Login

Enter Your Email

Enter Your password

LOGIN

3.Frame Layout

Android Frame layout is a ViewGroup subclass that is used to specify the position of multiple views placed on top of each other to represent a single view screen. Generally, we can say FrameLayout simply blocks a particular area on the screen to display a single view. Here, all the child views or elements are added in stack format means the most recently added child will be shown on the top of the screen. But, we can add multiple children's views and control their positions only by using gravity attributes in FrameLayout.

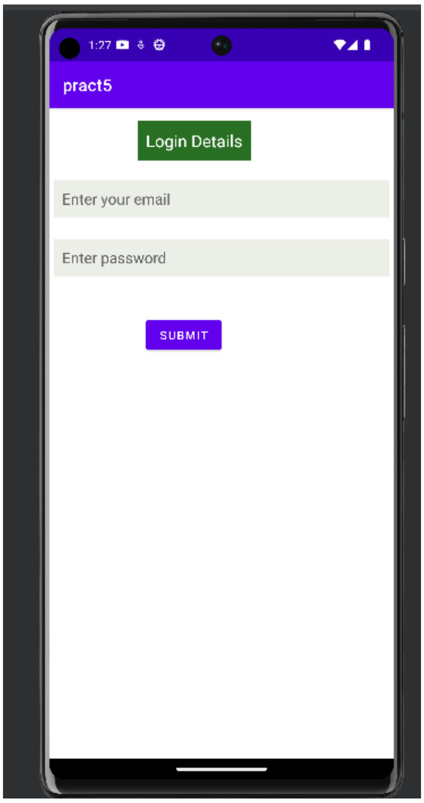
Activity_main.xml:-

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

```
 android:layout_height="match_parent"
 android:orientation="vertical"
 android:padding="5dp">
 <TextView
     android:id="@+id/txtvw1"
     android:layout_width="wrap_content"
android:layout_height="wrap_content"
 android:layout_marginLeft="100dp"
 android:layout_marginTop="10dp"
 android:background="#286F24"
 android:padding="10dp"
 android:text="Login Details"
 android:textColor="#FFFFFF"
 android:textSize="20sp" />
 <EditText
 android:id="@+id/editText1"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_centerHorizontal="true"
 android:layout_marginTop="80dp"
 android:background="#ECEEE8"
 android:hint="Enter your email"
 android:padding="10dp" />

 <EditText
 android:id="@+id/editText2"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_centerHorizontal="true"
 android:layout_marginTop="150dp"
 android:background="#ECEEE8"
 android:hint="Enter password"
```

```
    android:padding="10dp" />

    <Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="110dp"
   android:layout_marginTop="240dp"
    android:text="Submit" />
</FrameLayout>
```

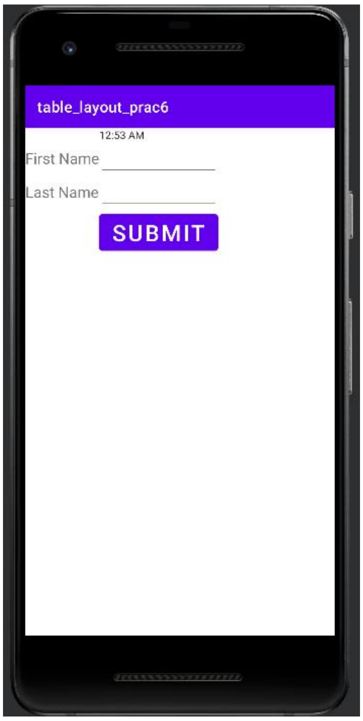
Output :-



4.Table Layout

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

    <TableRow
        android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextClock
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textClock"
        android:layout_column="2" />

</TableRow>

<TableRow>

    <TextView
        android:text="First Name"
        android:textSize="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1" />

    <EditText
        android:width="200px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow>

    <TextView
        android:text="Last Name"
        android:textSize="20dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1" />

    <EditText
        android:width="100px"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">


</TableRow>

<TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>
    <TableRow
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Submit"
            android:textSize="30dp"
            android:id="@+id/button"
            android:layout_column="2" />
    </TableRow>

</TableLayout>
```

Output :-



5. Constraint layout

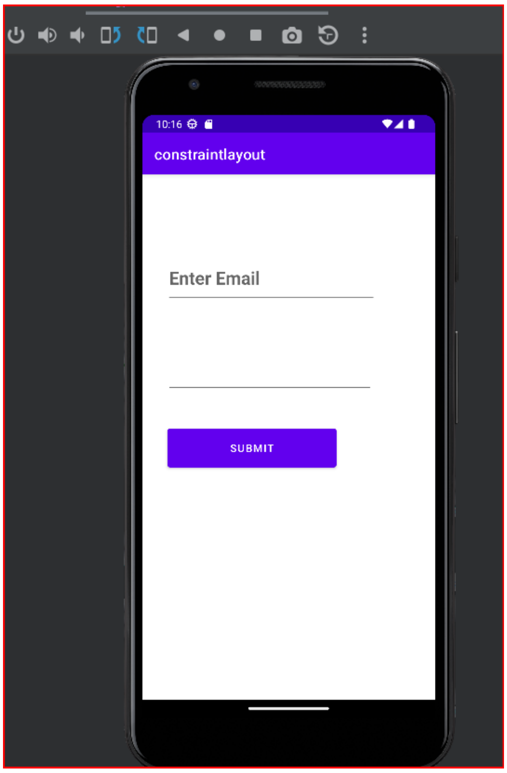
Activity_main.xml:-

```
<?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/editTextTextEmailAddress"
    android:layout_width="282dp"
    android:layout_height="69dp"
    android:ems="10"
    android:hint="Enter Email"
    android:inputType="textEmailAddress"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.286"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.164" />

<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="277dp"
    android:layout_height="56dp"
    android:ems="10"
    android:inputType="textPassword"
    android:textColor="#191818"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.276"
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.367" />

    <Button
        android:id="@+id/button"
        android:layout_width="227dp"
        android:layout_height="64dp"
        android:text="Submit"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.201"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.523" />

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

Output :-


10:16
constraintlayout
Enter Email
SUBMIT