# Experiment3.1

**Student Name: Sanchit Singal** **UID: 21BCS1569**
**Branch: CSE** **Section/Group: 606/B**
**Semester: 6th** **Date of Performance: 29/03/24**
**Subject Name: Mobile App Development** **Subject Code: 21CSH-355**

1. **Aim:** Implement building blocks for Android Application using different layouts such as linear, relative and absolute.

2. **Objective:** The objective of implementing building blocks for an Android application using different layouts such as linear, relative, and absolute is to create a diverse and visually appealing user interface that accommodates various design requirements. Different layout types offer flexibility in organizing UI components, and understanding their usage is crucial for effective Android app development.

3. **Input/Apparatus Used:**

- **Android Studio:** The official IDE for Android development. Download and install Android Studio from the official website: Android Studio.
- **Android SDK:** The Android Software Development Kit (SDK) is essential for developing Android applications. Android Studio usually comes bundled with the SDK, but you may need to update it through the SDK Manager within Android Studio.
- **Java Development Kit (JDK):** Android apps are primarily written in Java or Kotlin. Make sure you have the Java Development Kit installed. Android Studio supports JDK. You can download it from the Oracle website: Java SE Downloads.
- **Android Virtual Device (AVD) or Physical Android Device:** You need a device to test your Android application. You can use an emulator (AVD) that comes with Android Studio or a physical Android device connected to your computer.

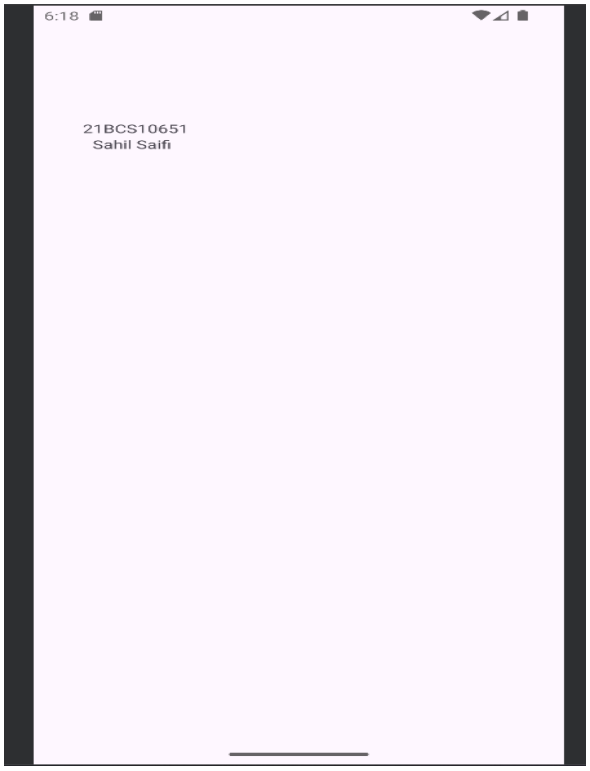
## 4. Procedure:

1) **Absolute Layout:** An Absolute Layout allows you to specify the exact location .i.e., X and Y coordinates of its children with respect to the origin at the top left corner of the layout. The absolute layout is less flexible and harder to maintain for varying sizes of screens that's why it is not recommended. Although Absolute Layout is deprecated now.

Some of the important Absolute Layout attributes are the following:

- android:id: It uniquely specifies the absolute layout
- android:layout_x: It specifies X-Coordinate of the Views (Possible values of this is in density-pixel or pixel)
- android:layout_y: It specifies Y-Coordinate of the Views (Possible values of this is in dp or px)

## Output:

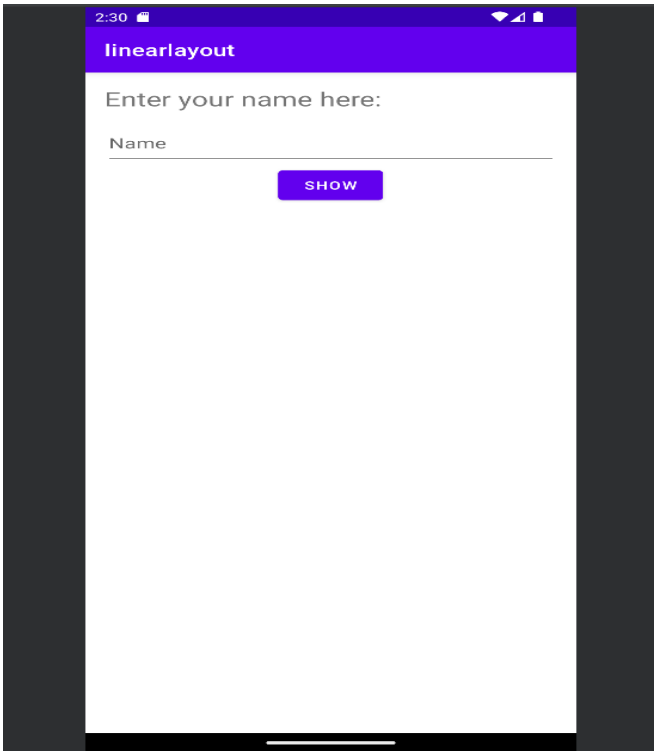


2) **Linear Layout:** Android LinearLayout is a ViewGroup subclass, used to provide child View elements one by one either in a particular direction either horizontally or vertically based on the orientation property. We can specify the linear layout orientation using android:orientation attribute.

All the child elements arranged one by one in multiple rows and multiple columns.

- Horizontal list: One row, multiple columns.
- Vertical list: One column, multiple rows.

## Output:



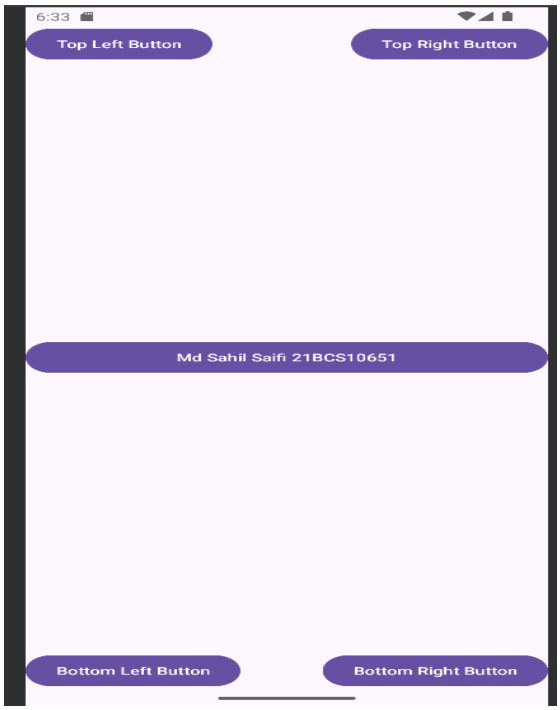

3) **Relative Layout:** The relative layout is used to arrange the child views in a proper order which means arranging the child objects relative to each other. Generally, if we create an application using a linear layout that consists of 5 buttons. Even if we specify weight and gravity properties they will not be relatively arranged. To arrange them in a proper order we need to use the relative layout. To arrange them we need some advanced properties. Basically, we use layout_width, layout_height, layout_text properties while creating an

application using the linear layout. But we need some more advanced properties which are supported by relative layout. There are so many properties that are supported by relative layout.

Some of the most used properties are listed below

- layout_alignParentTop
- layout_alignParentBottom
- layout_alignParentRight
- layout_alignParentLeft
- layout_centerHorozontal
- layout_centerVertical
- layout_above
- layout_below

## Output:



## 5. Learning Outcomes:

1. I have learned the process of installing Android Studio, a tool for Android app development.
2. I understand the importance of configuring SDKs and virtual devices for a smooth development environment.
3. I now understand the significance of testing applications on a virtual device, ensuring a well-prepared development setup.