**Experiment 1**

**Title: Install Android Studio with Latest Configuration in your System.**

**Aim:** Installation Steps

1. **Install Android Studio on Windows**
2. Open the folder where you downloaded and saved the Android Studio installation file.
3. Double-click the downloaded file.
4. If you see a User Account Control dialog about allowing the installation to make changes to your computer, click Yes to confirm the installation. (Figure 1)
5. Click Next to start the installation. (Figure 2)
6. Accept the default installation settings for all steps.
7. Click Finish when the installation is done to launch Android Studio. (Figure 3)
8. Choose your preference of light or dark theme when Android Studio first launches. Screenshots in this course use the light theme, but choose whichever one you prefer. (Figure 4)
9. During the installation, the setup wizard downloads and installs additional components and tools needed for Android app development. This may take some time depending on your internet speed. During this time, you may see a User Account Control dialog for Windows
10. You may also receive a Windows Security Alert about adb.exe. Click Allow Access,if needed, to continue the installation. (Figure 6)
11. When the download and installation completes, click Finish. (Figure 7)

**Screenshots:**

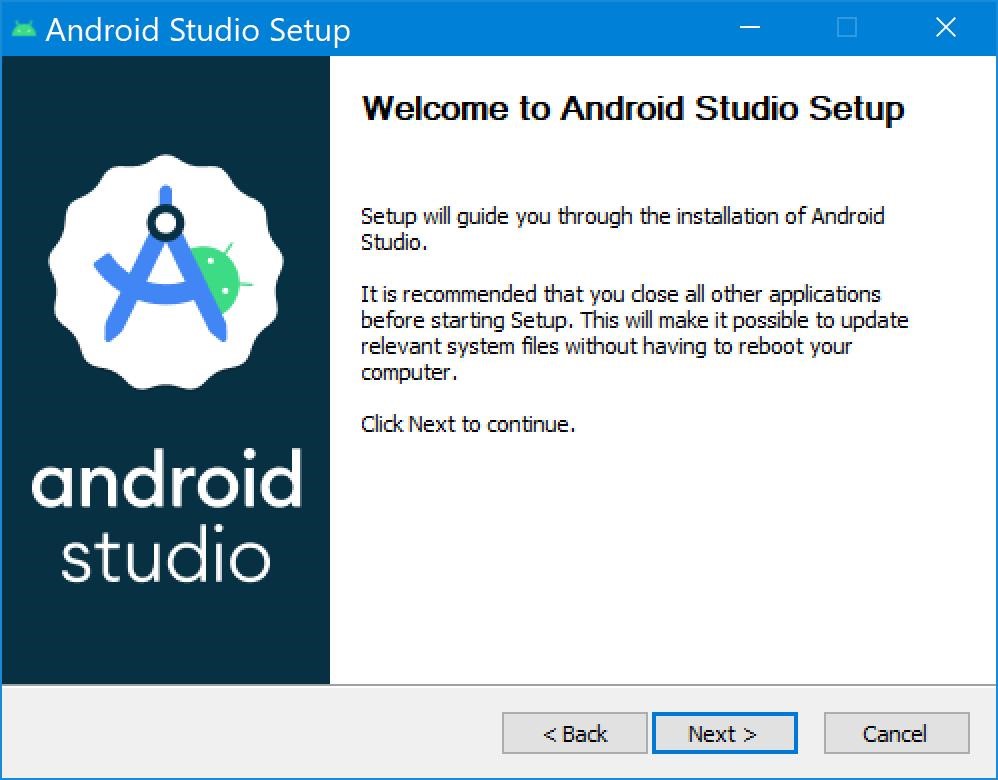
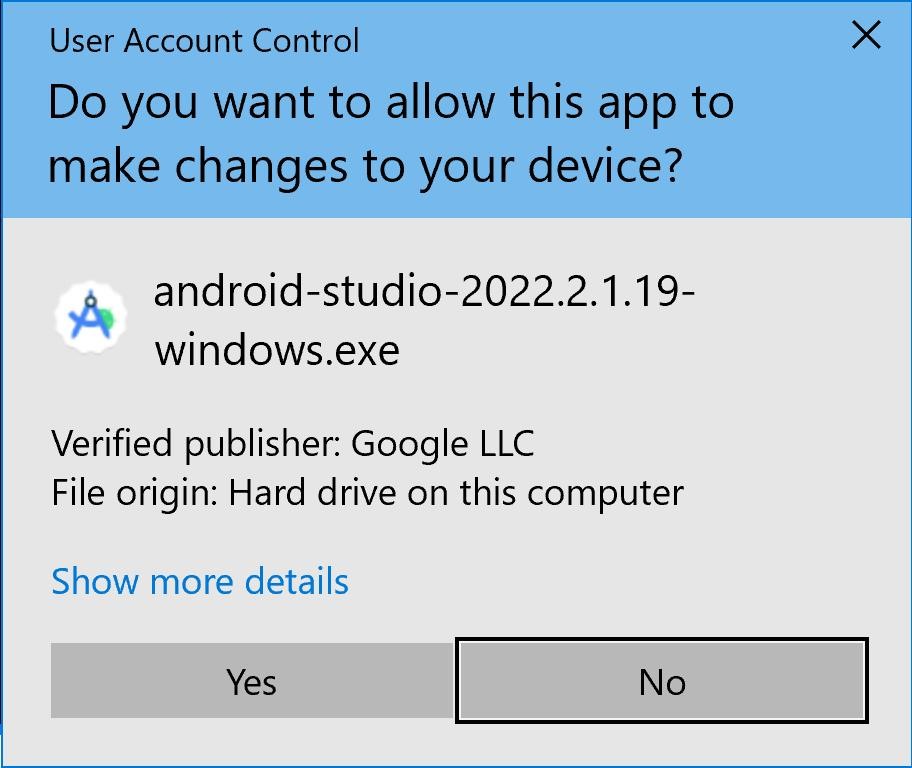


Figure 1: security pop up box Figure 2: installation steps

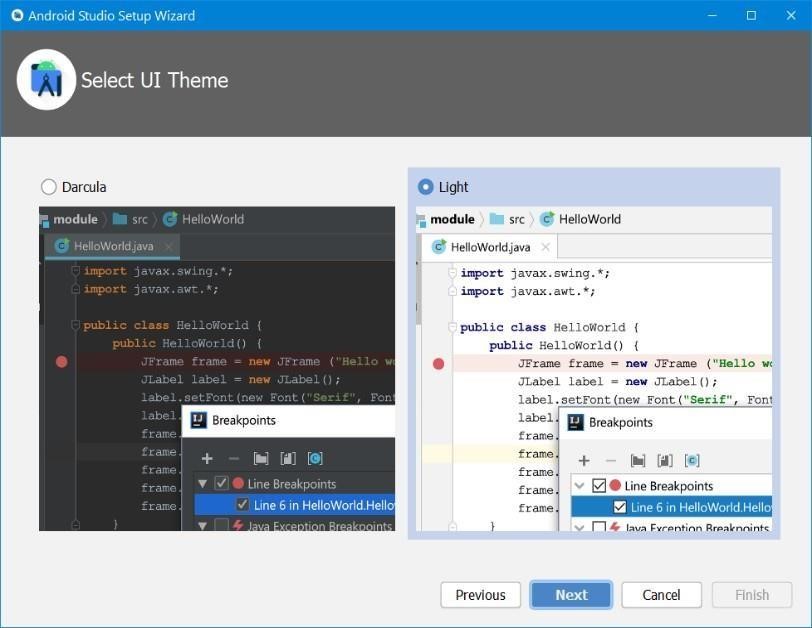
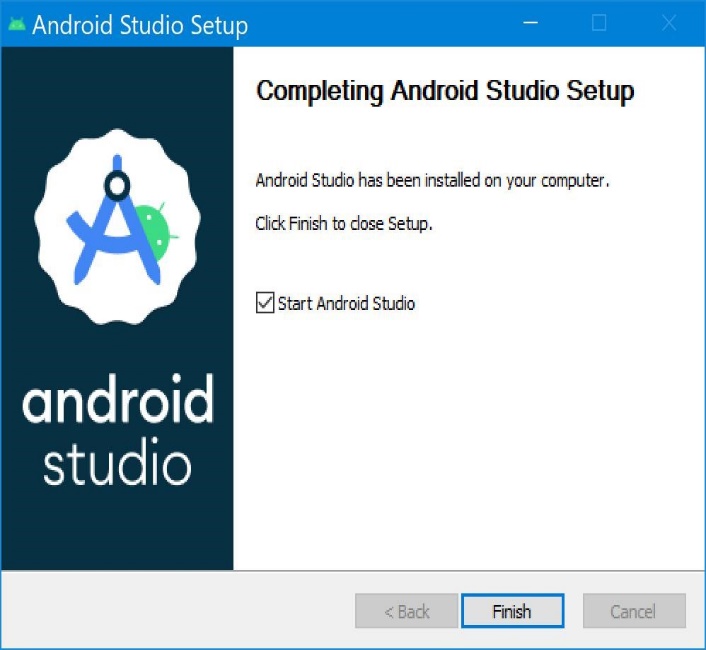


Figure 3: completion of installation Figure 4: select ui of android studio

****

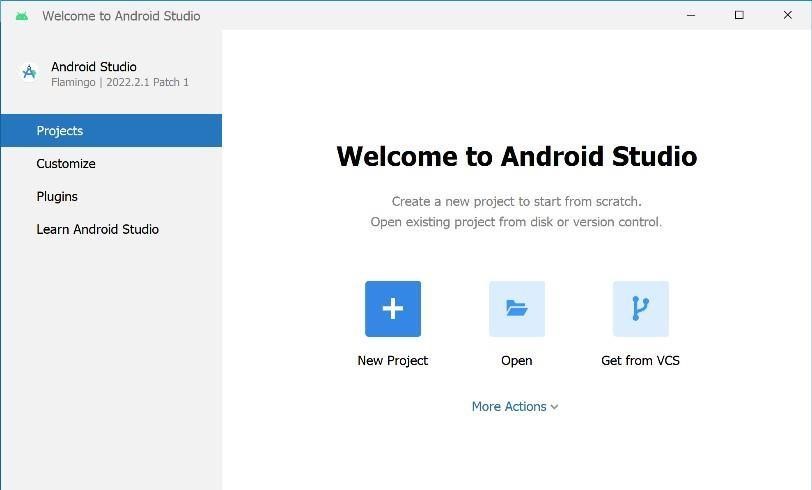
****Figure 5: give the access Figure 6: allow access

Figure 7: starting android studio

**2.Introduction to Android Studio:**

Android Studio is the official Integrated Development Environment (IDE) for Android App development. It is a powerful tool that allows developers to build high-quality applications for the Android platform. It has complete tools for the process of Android App development. From writing code to testing and deployment, Android studio has all the functionalities for developers to develop an Android App.

**3.Manifests File:**

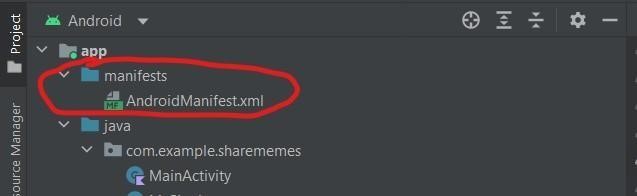
** AndroidManifest.xml**

Figure 8: Manifests file structure

**b) Java Folder:**

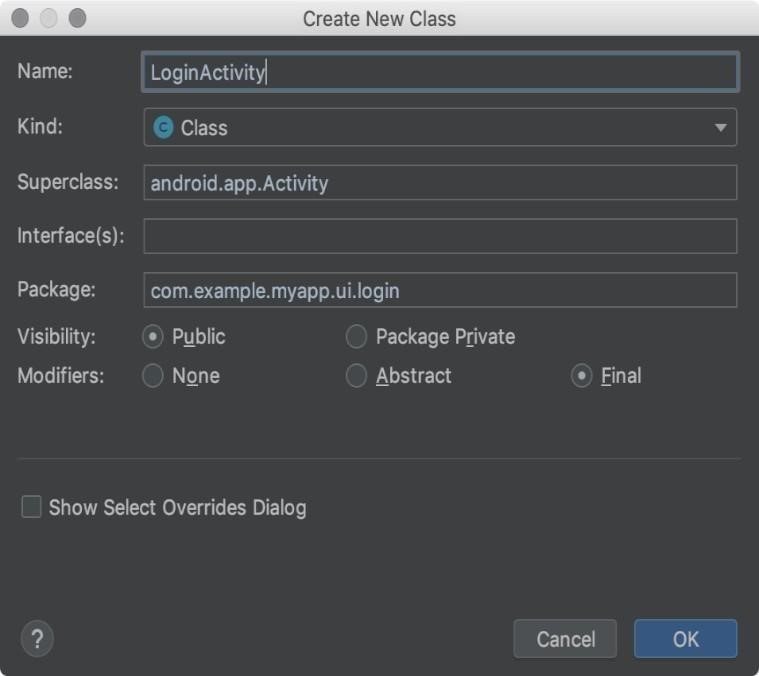
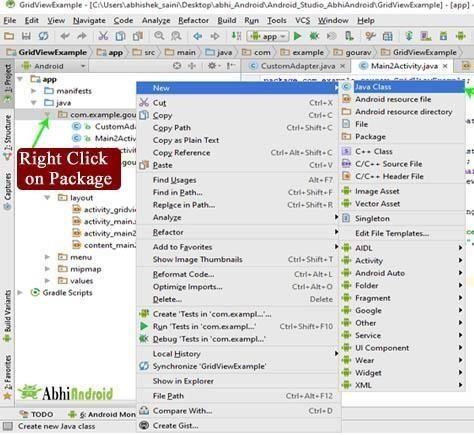


Figure 9:create steps of java file Figure 10: file name

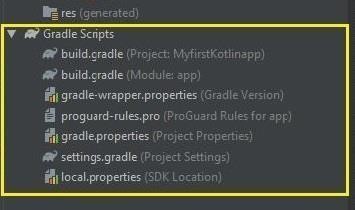
** c) Gradel File:**

Figure 11: gradle file structure

The Gradle file is a configuration script used in Android projects to manage dependencies, build settings, and tasks. It defines project structure, dependencies, and plugins, allowing automated build processes and version control. The build.gradle file is essential for compiling the project, packaging the APK, and integrating third-party libraries or tools.

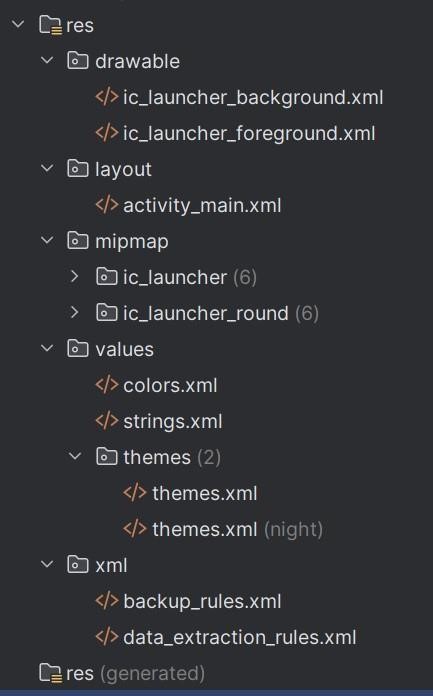
 **d) Res Folder:**

Figure 12: resource file structure

1. **Steps to enable developer mode**
2. Open Settings on your Android device and go to the About phone section. (Figure 13)
3. Locate the phone's build number and tap on it multiple times until you see a message regarding developer options and how many times you have to tap to enable it. On Samsung devices, you need to tap on Software information. (Figure 14, 15, 16)
4. After tapping the required amount of times, you will see a message saying that the developer options are on. You may also have to enter your phone's security code.
5. Now go back to your phone Settings and go to System. Scroll down to the bottom, and you will find the Developer options. (Figure 17)
6. Turn on USB debugging and wireless debugging, make sure your laptop and mobile with same network connection.

**Screenshots:**

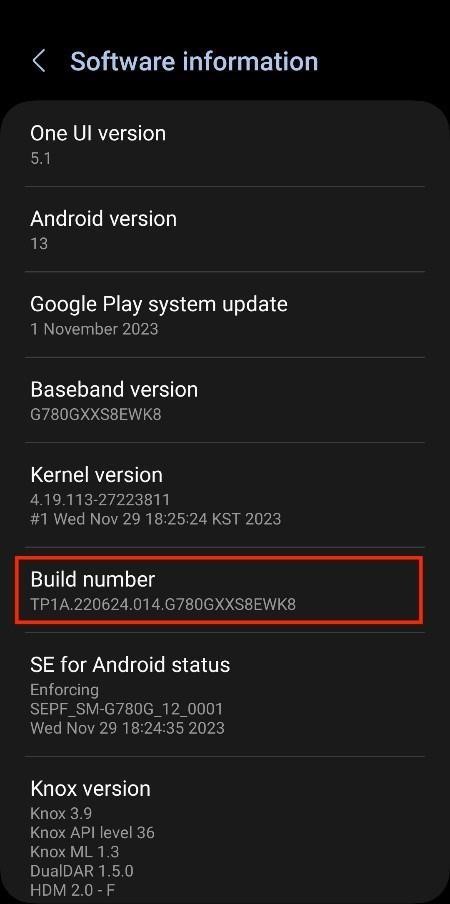
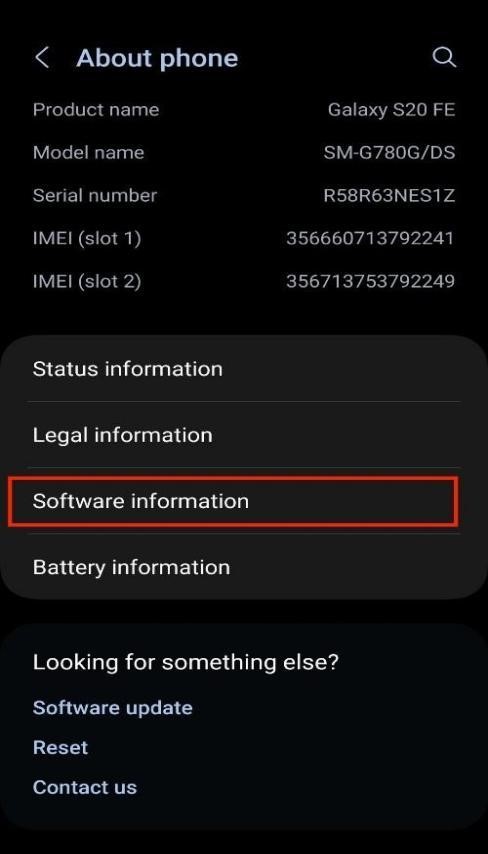


Figure 13 Figure 14 Figure 15

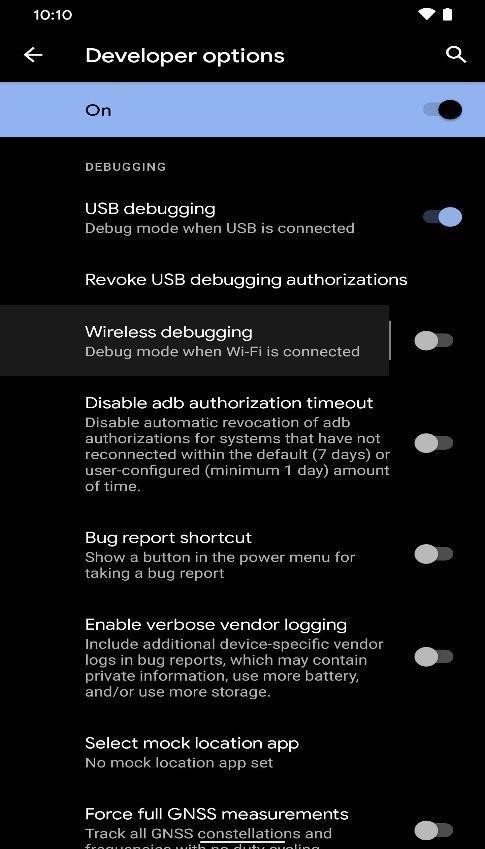
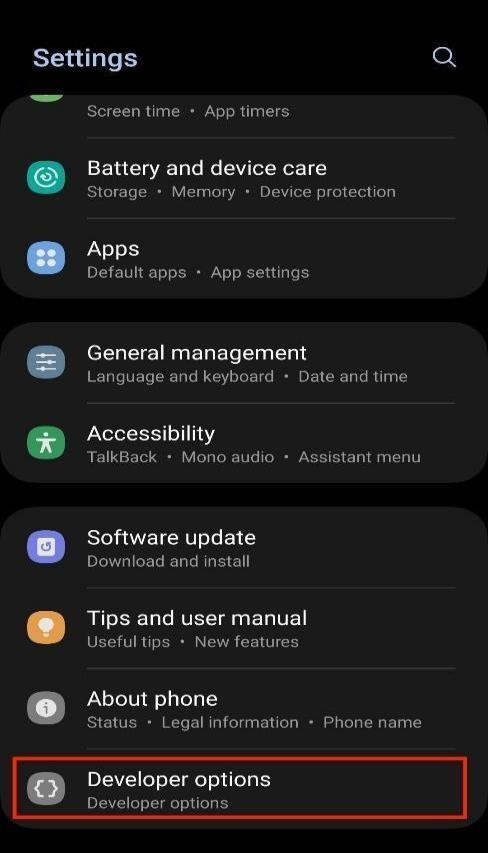


Figure 16 Figure 17 Figure 18

**4.Application Installation Methods**

There are 3 methods given below :

**a) APK Installation**

The APK (Android Package Kit) installation method involves downloading the app's APK file directly onto the user's device and manually installing it. This method is useful when the app is not available on official app stores like Google Play Store.

**b) Wired Installation**

Wired installation involves connecting the user's Android device to a computer via a USB cable and transferring the app directly from the computer to the device. This method is commonly used by developers or for sideloading apps.

**c) Wireless Installation**

Wireless installation allows users to install the Internship EviApp without needing a physical connection to a computer. This can be done through app distribution platforms, cloud services, or direct download links.

**Steps for installing :**

**APK Installation steps**

1. **Build the APK:**

* In Android Studio, go to Build > Build Bundle(s)/APK(s) > Build APK(s).
* Once the build is complete, you'll find the APK file in the app/build/outputs/apk/ directory.

1. **Transfer the APK to your device:**

* Connect your Android device to your computer via USB or use a cloud service/email totransfer the APK.

1. **Install the APK:**

* On your Android device, navigate to the location where you stored the APK file.
* Tap on the APK file to begin the installation process.
* You may need to enable "Unknown Sources" or "Install unknown apps" you’re your device settings to allow installation.

1. **Wired Installation steps**

**i). Enable Developer Options and USB Debugging:**

On your Android device, go to Settings > About phone and tap Build number seven times toenable Developer Options.

Go to Settings > Developer options and enable USB debugging.

**ii). Connect the device**:

Connect your Android device to your computer using a USB cable. o Ensure your device is recognized by Android Studio.

**iii). Run the app:**

In Android Studio, click the Run button or go to Run > Run 'app'.

Select your connected device from the list and click OK. o Android Studio will build the app and install it on your device.

1. **Wireless Installation steps**

**i). Enable Developer Options and Wireless Debugging:**

On your Android device, go to Settings > About phone and tap Build number seven timesto enable Developer Options. o Go to Settings > Developer options and enable Wireless debugging (available on Android11 and above).

**ii). Connect to the same network:**

Ensure your Android device and your computer are connected to the same Wi-Fi network.

**iii). Pair the device:**

In Android Studio, go to View > Tool Windows > Device Manager.

**iv). Steps to enable developer mode**

1. Open Settings on your Android device and go to the About phone section. (Figure 13)
2. Locate the phone's build number and tap on it multiple times until you see a message regarding developer options and how many times you have to tap to enable it. On Samsung devices, you need to tap on Software information. (Figure 14, 15, 16)
3. After tapping the required amount of times, you will see a message saying that the developer options are on. You may also have to enter your phone's security code.
4. Now go back to your phone Settings and go to System. Scroll down to the bottom, and you will find the Developer options. (Figure 17)
5. Turn on USB debugging and wireless debugging, make sure your laptop and mobile with same network connection.

**Experiment 2**

**Title: Study of layouts and UI components used for UI Design.**

**Step1: List Out UI components used for designing Layout.**

1. **TextView:** Displays text to the user. It can be customized with different fonts, sizes, colors, and text styles.
2. **Button:** A clickable element used to perform an action when tapped. It can display text, icons, or both.
3. **EditText:** Allows users to input and edit text. It supports various input types like passwords, numbers, and email.
4. **ImageView:** Displays images and can be used to load images from resources, assets, or URLs with different scaling options.
5. **RecyclerView:** A flexible view for displaying large data sets in a scrollable list with customizable layouts and animations.
6. **CheckBox:** A two-state button that can be either checked or unchecked, often used in forms and selection lists.
7. **RadioButton:** Allows users to select one option from a set of mutually exclusive choices, typically used within a RadioGroup.
8. **Switch:** A two-state toggle switch that allows users to turn on/off settings or functionality with a simple gesture.
9. **ProgressBar:** Shows the progress of a task, either determinate with a specific value or indeterminate with a looping animation.
10. **Spinner:** A drop-down menu that allows users to select a single item from a list of options, similar to a combo box.
11. **SeekBar**: A slider that allows users to choose a value from a range by moving the thumb along the bar.
12. **CardView**: A container that provides a flexible, rounded-corner layout with shadow effects, often used to group content with a clean design.
13. **Toolbar**: A customizable action bar at the top of the screen that provides navigation, branding, and actions like search or settings.

**Step 2 : List out all various Layouts use for designing Android Application.**

**1. LinearLayout**

**Description:** Aligns its child views in a single direction, either vertically or horizontally.

**Details:**

* + You can use android:orientation to specify the direction (vertical or horizontal).
  + Views are stacked one after another in the specified orientation.
  + Supports weight distribution through the android:layout\_weight attribute, allowing views to take up space proportionally.

**2. RelativeLayout**

**Description:** Positions its child views relative to each other or the parent layout.

**Details:**

* + Child views can be aligned to the left, right, top, or bottom of other views.
  + You can use attributes like android:layout\_alignParentTop, android:layout\_toRightOf, etc., for precise positioning.
  + Provides flexible layout designs, especially for complex UI where views need to overlap or position relative to one another.

**3. ConstraintLayout**

**Description:** A powerful layout that allows complex layouts with a flat hierarchy.

**Details:**

* Views are positioned relative to each other using constraints.
* Offers more flexibility and efficiency than RelativeLayout, with features like bias, chains, and guidelines.
* Helps reduce nested view hierarchies, leading to better performance.

**4. FrameLayout**

**Description:** A simple layout that allows child views to be stacked on top of each other.

**Details:**

* + Typically used for displaying a single view or overlapping views.
  + Child views are positioned at the top left corner by default, but can be positioned using android:layout\_gravity.
  + Commonly used for displaying fragments, custom drawing, or loading indicators.

**5. TableLayout**

**Description:** Organizes child views into rows and columns, similar to an HTML table.

**Details:**

* Each row is defined using a TableRow element.
* Columns are created automatically based on the number of views in each row.
* You can control the span of views across multiple columns using android:layout\_span.

**6. GridLayout**

**Description:** Organizes child views into a grid with rows and columns.

**Details:**

* Offers a more flexible grid system compared to TableLayout.
* Allows precise control over row and column spans.
* Supports dynamic layouts, with views that can be positioned based on their index in the grid.

**7. AbsoluteLayout *(Deprecated)***

**Description:** Positions child views at absolute coordinates (x, y).

**Details:**

* Requires setting android:layout\_x and android:layout\_y for each view.
* Not recommended due to difficulties in handling different screen sizes and resolutions.

**8. CoordinatorLayout**

**Description:** A super-powered FrameLayout that provides advanced control for animations and interactions between child views.

**Details:**

* + Commonly used as the root layout for implementing Material Design components like the collapsing toolbar.
  + Supports behaviors that allow child views to react to scroll events, touch events, and window insets.

**9. ScrollView**

**Description:** A layout that allows vertical scrolling of its child views.

**Details:**

* Can contain only one direct child view, but that child can be a layout containing multiple views.
* Often used to handle content that is too large to fit on the screen.
* There’s also a HorizontalScrollView for horizontal scrolling.

**10. DrawerLayout**

**Description:** A layout used to implement a navigation drawer, a panel that slides in from the edge of the screen.

**Details:**

* Typically used as the root layout for activities that include a navigation drawer.
* Contains two children: the main content view and the drawer view.
* Allows users to navigate between different sections of the app.

**11. RelativeLayout**

**Description:** Allows child views to be positioned relative to each other or the parent layout.

**Details:**

* Enables complex layouts without nesting, leading to better performance.
* Child views can be positioned using attributes like layout\_alignParentTop, layout\_centerHorizontal, etc.

**12. ViewGroup**

**Description:** A special view that can contain other views, serving as a base class for layouts.

**Details:**

* It’s an invisible container that helps structure the UI by grouping views together.
* Handles layout measurement, drawing, and interaction with its child views.
* Custom layouts are usually built by extending ViewGroup.

**13. FlexboxLayout**

**Description:** A layout that aligns its child views similarly to the CSS Flexbox model.

**Details:**

* Supports flexible sizing and aligning of child views in both horizontal and vertical directions.
* Provides control over wrapping, spacing, and alignment, making it suitable for responsive designs.

**Step3: Demonstrate the different Ui components.**

**Layout/Screen XML code:**

1. **activity\_main.xml**

**Code:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity2">

<TextView

android:id="@+id/tv1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/Label"

android:fontFamily="sans-serif"

android:textStyle="bold"

android:textColor="@color/red"

android:textAllCaps="true"

tools:layout\_editor\_absoluteX="120dp"

tools:layout\_editor\_absoluteY="20dp"

tools:ignore="MissingConstraints" />

<Button

android:id="@+id/btn1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingRight="60dp"

android:paddingLeft="60dp"

tools:layout\_editor\_absoluteX="100dp"

tools:layout\_editor\_absoluteY="80dp"

android:text="@string/B"

android:drawableTop="@mipmap/ic\_launcher\_round"

tools:ignore="MissingConstraints" />

<ImageView

android:id="@+id/imgV1"

android:contentDescription="@string/todo"

android:src="@mipmap/ic\_launcher\_round"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:layout\_editor\_absoluteX="160dp"

tools:layout\_editor\_absoluteY="230dp"

tools:ignore="MissingConstraints" />

<EditText

android:id="@+id/edt1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:autofillHints="Enter your name"

android:inputType="text"

android:minHeight="48dp"

android:paddingHorizontal="30dp"

android:text="@string/enterName"

tools:ignore="LabelFor,MissingConstraints"

tools:layout\_editor\_absoluteX="100dp"

tools:layout\_editor\_absoluteY="320dp" />

<CheckBox

android:id="@+id/ckBox1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:checked="true"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="100dp"

tools:layout\_editor\_absoluteY="400dp" />

<RadioButton

android:id="@+id/rdBtn1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/yes"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="222dp"

tools:layout\_editor\_absoluteY="400dp" />

<RadioGroup

android:id="@+id/rg1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="100dp"

tools:layout\_editor\_absoluteY="486dp">

<RadioButton

android:id="@+id/male"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/male" />

<RadioButton

android:id="@+id/female"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/female" />

</RadioGroup>

<ImageButton

android:id="@+id/imgBtn1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@string/Label"

android:minWidth="50dp"

android:minHeight="50dp"

tools:ignore="MissingConstraints,TouchTargetSizeCheck"

tools:layout\_editor\_absoluteX="150dp"

tools:layout\_editor\_absoluteY="627dp"

android:src="@mipmap/ic\_launcher\_round"

/>

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:onClick="Intent"

android:text="Second"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="272dp"

tools:layout\_editor\_absoluteY="627dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Class File/Java code:**

**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;  
  
public class MainActivity extends AppCompatActivity {  
  
 @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;  
 });  
 }

}

**Screenshots:**

**A screenshot of a phone

Description automatically generated**

First Screen

**Step3: Demonstrate the different Layout.**

**Layout/Screen XML code:**

**Second.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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"  
 tools:context=".SecondActivity">  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginStart="40dp"  
 android:padding="30dp"  
 android:layout\_alignParentTop="true"  
 >  
 <EditText  
 android:id="@+id/edt1"  
 android:layout\_width="283dp"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/first"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
 <EditText  
 android:id="@+id/etd2"  
 android:layout\_width="283dp"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/second"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
  
 <TextView  
 android:id="@+id/tv1"  
 android:layout\_width="280dp"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="@string/res"  
 android:textSize="25sp"  
 android:textStyle="bold"  
 />  
 </LinearLayout>  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/linearLayout"  
 android:layout\_marginTop="100dp">  
 <Button  
 android:id="@+id/btn1"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="@string/add"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn2"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintWidth\_percent="0.4"/>  
 <Button  
 android:id="@+id/btn2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="@string/sub"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/btn1"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintWidth\_percent="0.4"/>  
 <Button  
 android:id="@+id/btn3"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="@string/mul"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn4"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn1"  
 app:layout\_constraintWidth\_percent="0.4"/>  
 <Button  
 android:id="@+id/btn4"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="@string/div"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/btn3"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn2"  
 app:layout\_constraintWidth\_percent="0.4"/>  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</RelativeLayout>

**Class File/Java code:**

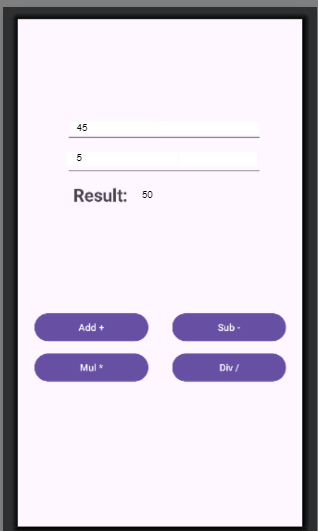
1. **SecondActivity.java**

package com.example.myapplication;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SecondActivity extends AppCompatActivity {  
  
 private EditText editText1, editText2;  
 private TextView textViewResult;  
 private Button buttonAdd, buttonSub, buttonMul, buttonDiv;  
  
 @Override

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*calculator*);  
  
 // Initialize UI elements  
 editText1 = findViewById(R.id.*editText1*);  
 editText2 = findViewById(R.id.*editText2*);  
 textViewResult = findViewById(R.id.*textView3*);  
 buttonAdd = findViewById(R.id.*button1*);  
 buttonSub = findViewById(R.id.*button2*);  
 buttonMul = findViewById(R.id.*button3*);  
 buttonDiv = findViewById(R.id.*button4*);  
  
 // Add button click listeners  
 buttonAdd.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 calculate('+');  
 }  
 });  
  
 buttonSub.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 calculate('-');  
 }  
 });  
  
 buttonMul.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 calculate('\*');  
 }  
 });  
  
 buttonDiv.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 calculate('/');  
 }  
 });  
 }  
  
 private void calculate(char operator) {String value1 = editText1.getText().toString();  
 String value2 = editText2.getText().toString();  
  
 if (value1.isEmpty() || value2.isEmpty()) {  
 textViewResult.setText("Please enter numbers");  
 return;  
 }  
  
 double num1 = Double.*parseDouble*(value1);  
 double num2 = Double.*parseDouble*(value2);  
 double result = 0;  
  
 switch (operator) {  
 case '+':  
 result = num1 + num2;  
 break;  
 case '-':  
 result = num1 - num2;  
 break;  
 case '\*':  
 result = num1 \* num2;  
 break;  
 case '/':  
 if (num2 != 0) {  
 result = num1 / num2;  
 } else {  
 textViewResult.setText("Cannot divide by zero");  
 return;  
 }  
 break;  
 }  
  
 textViewResult.setText(String.*valueOf*(result));  
 }  
}

**A screenshot of a cell phone

Description automatically generatedScreenshots:**

****

Default

Default Add

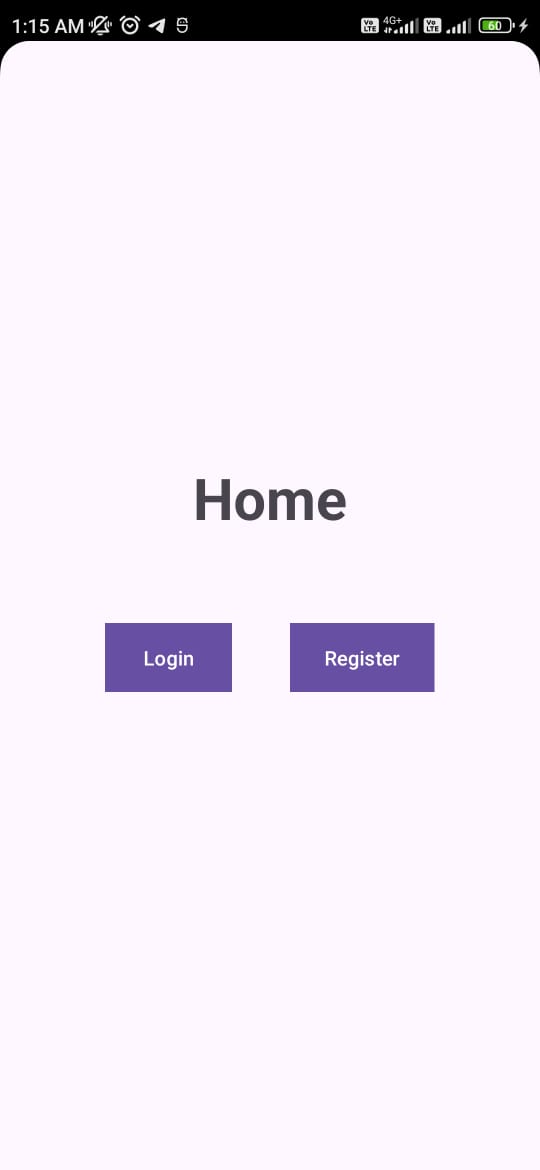
**Experiment 3**

**Title: Designing UI for Sign in and Sign up screen.**

Aim: Android UI design involves the use of prebuilt Android UI components, such as structured layout objects and UI controls, to build the graphical user interface for our applications.

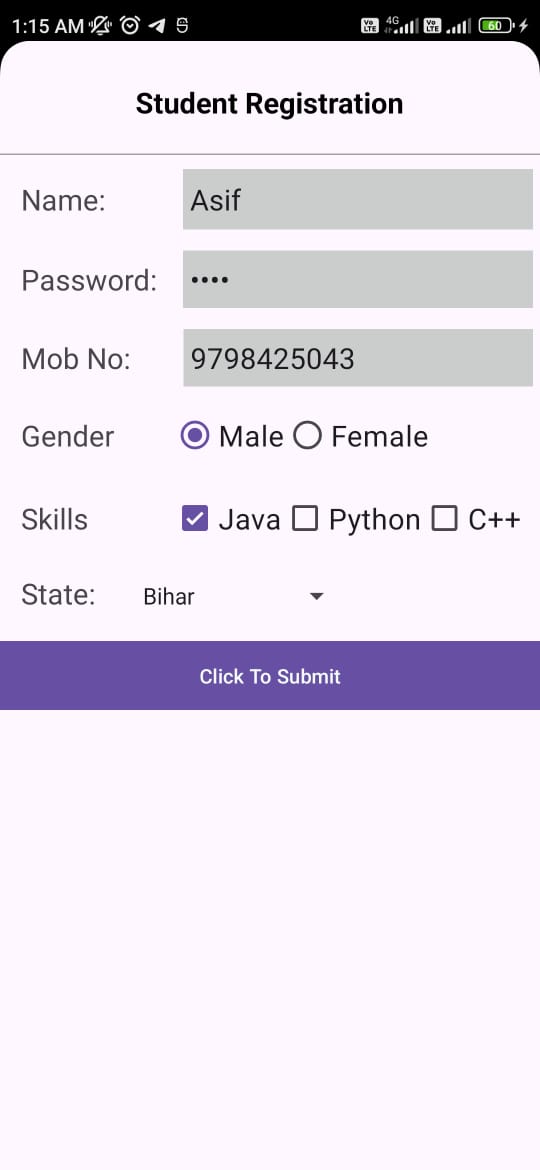
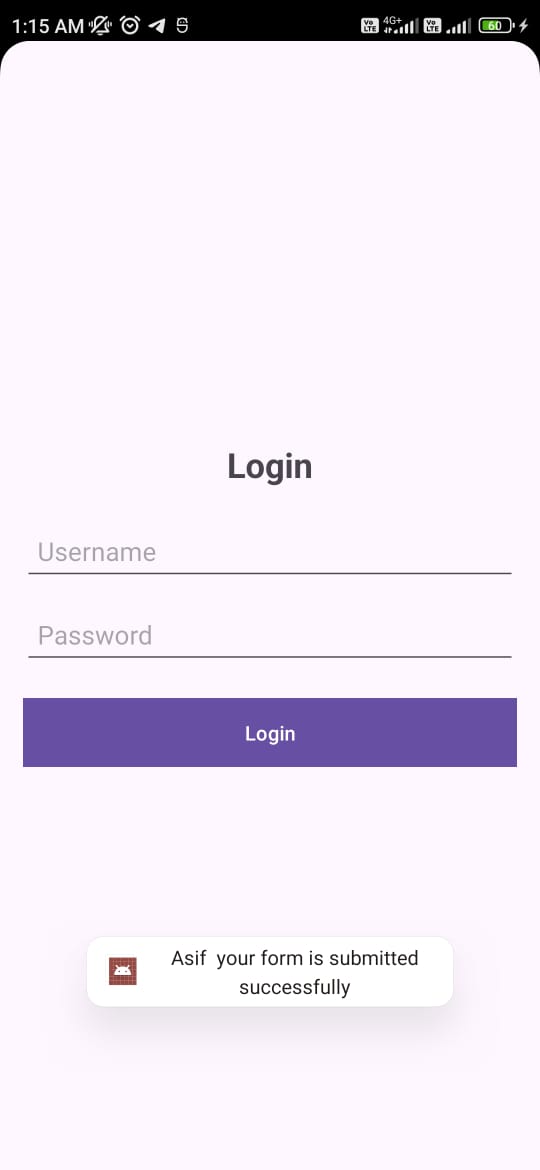
**Activity\_Main.xml**

<?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:layout\_width="match\_parent"  
android:id="@+id/main"  
android:layout\_height="match\_parent"  
android:orientation="vertical"  
android:padding="16dp"  
android:gravity="center"  
tools:context=".MainActivity">  
<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Home"  
 android:textSize="40sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="20dp" />  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp">  
 <Button  
 android:id="@+id/toLogin"  
 android:layout\_margin="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:background="#B71C1C"  
 android:textColor="#FFFFFF" />  
 <Button  
 android:id="@+id/toRegister"  
 android:layout\_margin="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Register"  
 android:background="#B71C1C"  
 android:textColor="#FFFFFF" />  
</LinearLayout> <!-- This line was missing -->  
</LinearLayout>



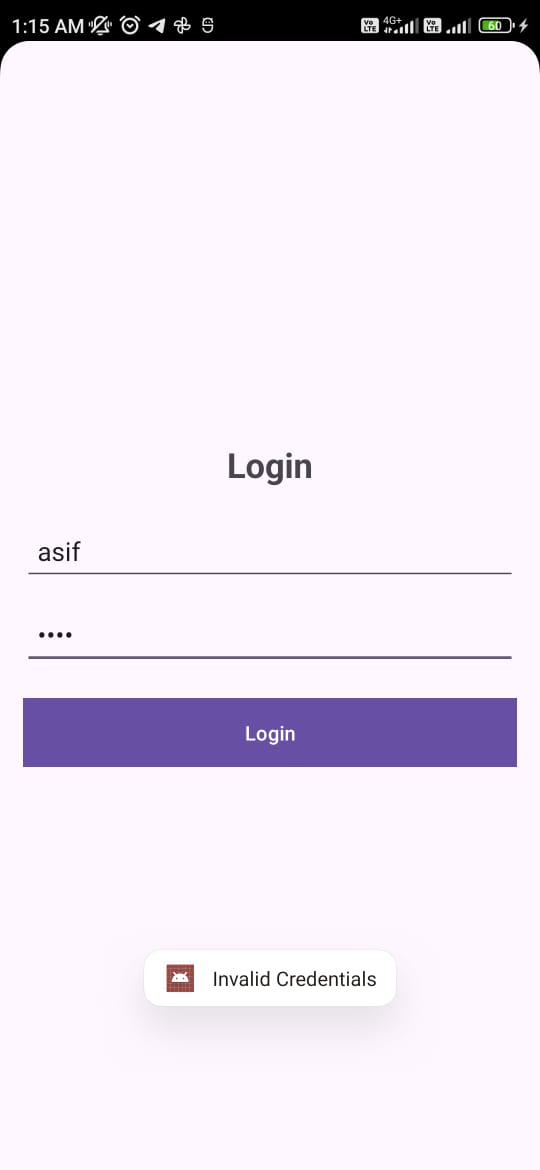
**registration.xml**

<?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"  
tools:context=".RegistrationActivity"  
android:orientation="vertical">  
<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:text="Student Registration"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:padding="20dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
<View  
 android:layout\_width="400dp"  
 android:layout\_height="1dp"  
 android:background="#ACA1A1"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="2dp"/>  
<LinearLayout  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="5dp">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Name:"  
 android:textSize="20dp"  
 android:layout\_marginLeft="5dp"/>  
 <EditText  
 android:padding="5dp"  
 android:id="@+id/userName"  
 android:layout\_width="300dp"  
 android:layout\_height="42dp"  
 android:layout\_marginLeft="49dp"  
 android:background="#5676"  
 android:hint="Enter Username.."  
 android:textSize="20dp"  
 tools:ignore="TouchTargetSizeCheck,TouchTargetSizeCheck"/>  
</LinearLayout>  
<LinearLayout  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="5dp">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password:"  
 android:textSize="20dp"  
 android:layout\_marginLeft="5dp" />  
 <EditText  
 android:padding="5dp"  
 android:id="@+id/userPassword"  
 android:layout\_width="300dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="13dp"  
 android:autofillHints=""  
 android:background="#5676"  
 android:hint="Enter Password.."  
 android:inputType="textPassword"  
 android:textSize="20sp"  
 tools:ignore="TouchTargetSizeCheck" />  
</LinearLayout>  
<LinearLayout  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="5dp">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Mob No:"  
 android:textSize="20dp"  
 android:layout\_marginLeft="5dp" />  
 <EditText  
 android:padding="5dp"  
 android:id="@+id/userContact"  
 android:layout\_width="300dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="32dp"  
 android:background="#5676"  
 android:hint="Enter Contact No.."  
 android:inputType="number"  
 android:textSize="20dp"  
 tools:ignore="TouchTargetSizeCheck" />  
</LinearLayout>  
<RadioGroup  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Gender"  
 android:textSize="20dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_marginLeft="5dp"/>  
 <RadioButton  
 android:id="@+id/Male"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Male"  
 android:gravity="center"  
 android:textSize="20dp"/>  
 <RadioButton  
 android:id="@+id/Female"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Female"  
 android:gravity="center"  
 android:textSize="20dp"/>  
</RadioGroup>  
<RadioGroup  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:text="Skills"  
 android:layout\_marginRight="53dp"  
 android:layout\_marginLeft="5dp"/>  
 <CheckBox  
 android:id="@+id/Java"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Java"  
 android:textSize="20dp"/>  
 <CheckBox  
 android:id="@+id/Python"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Python"  
 android:textSize="20dp"/>  
 <CheckBox  
 android:id="@+id/cpp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="C++"  
 android:textSize="20dp"/>  
</RadioGroup>  
 <LinearLayout  
 android:padding="5dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:padding="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="State:"  
 android:textSize="20dp"  
 android:layout\_marginLeft="5dp" />  
 <Spinner  
 android:id="@+id/stateSpinner"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:textSize="20dp"  
 tools:ignore="TouchTargetSizeCheck" />  
 </LinearLayout>  
<Button  
 android:id="@+id/btnSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Click To Submit"  
 android:textColor="#F5FBF6"  
 android:background="#B71C1C"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>



**Login.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="match\_parent"  
android:id="@+id/main"  
android:layout\_height="match\_parent"  
android:orientation="vertical"  
android:padding="16dp"  
android:gravity="center">  
<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="20dp" />  
 <EditText  
 android:id="@+id/etUsername"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:hint="Username"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
 <EditText  
 android:id="@+id/etPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:minHeight="48dp"  
 android:padding="10dp" />  
<Button  
 android:id="@+id/btnLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:background="#B71C1C"  
 android:textColor="#FFFFFF" />  
</LinearLayout>



**Experiment 4**

**Title: Building “Hello World” application with customized theme and material design.**

Aim: Simple application using Material design with customized theme.

**MainActivity.java**

package com.example.practical4;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
import com.google.android.material.button.MaterialButton;  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);

MaterialButton button = findViewById(R.id.*materialButton*);  
 button.setOnClickListener(v -> {  
 });  
 }  
}

**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:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
<com.google.android.material.textview.MaterialTextView  
 android:id="@+id/helloWorldText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello, World!"  
 android:textSize="24sp"  
 android:textColor="@color/purple\_500"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent" />  
  
<com.google.android.material.button.MaterialButton  
 android:id="@+id/materialButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Click Me"  
 android:textColor="@android:color/white"  
 android:backgroundTint="@color/teal\_700"  
 app:cornerRadius="12dp"  
 app:layout\_constraintTop\_toBottomOf="@id/helloWorldText"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 android:layout\_marginTop="16dp"/>  
</androidx.constraintlayout.widget.ConstraintLayout>

**AndroidManifest.xml**

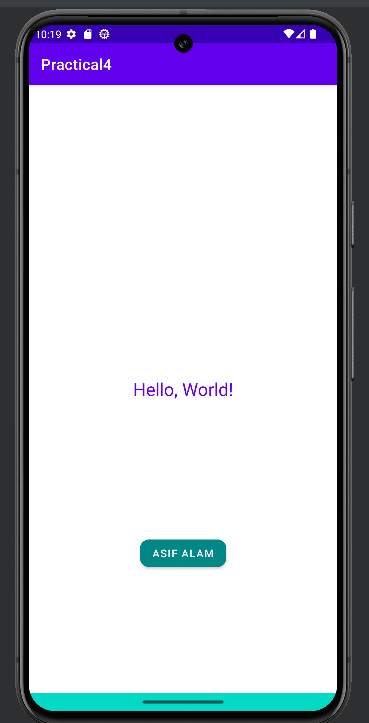
<manifest xmlns:android="http://schemas.android.com/apk/res/android">  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Practical4">  
 <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>  
 </application>  
</manifest>

**res/values/themes.xml**

<resources xmlns:tools="http://schemas.android.com/tools">  
 <style name="Theme.Practical4" parent="Theme.MaterialComponents.DayNight.DarkActionBar">  
 <item name="colorPrimary">@color/purple\_500</item>  
 <item name="colorPrimaryVariant">@color/purple\_700</item>  
 <item name="colorOnPrimary">@android:color/white</item>  
 <item name="colorSecondary">@color/teal\_200</item>  
 <item name="colorSecondaryVariant">@color/teal\_700</item>  
 <item name="colorOnSecondary">@android:color/white</item>  
 <item name="android:statusBarColor" tools:targetApi="l">@color/purple\_700</item>  
 <item name="android:navigationBarColor">@color/teal\_200</item>  
 </style>  
</resources>

**res/values/colors.xml**

<resources>  
 <color name="purple\_200">#BB86FC</color>  
 <color name="purple\_500">#6200EE</color>  
 <color name="purple\_700">#3700B3</color>  
 <color name="teal\_200">#03DAC5</color>  
 <color name="teal\_700">#018786</color>  
 <color name="black">#000000</color>  
 <color name="white">#FFFFFF</color>  
</resources>

**Output:**

**Experiment 5**

**Title: Demonstration of Android Activity and Fragment life cycle for “Hello World” application.**

**AndroidManifest.xml**

<?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.Practical5"  
 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>  
 </application>  
</manifest>

**MainActivity.java**

package com.example.practical5;  
import android.os.Bundle;  
import android.util.Log;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 if (savedInstanceState == null) {  
 getSupportFragmentManager().beginTransaction()  
 .replace(R.id.*main*, new BlankFragment())  
 .commit();  
 }  
 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;  
 });  
 }  
  
 protected void onStart() {  
 super.onStart();  
 Log.*d*("Lifecycle", "onStart");  
 }  
 protected void onResume() {  
 super.onResume();  
 Log.*d*("Lifecycle", "onResume");}  
  
 protected void onPause() {  
 super.onPause();  
 Log.*d*("Lifecycle", "onPause");  
 }  
 protected void onStop() {  
 super.onStop();  
 Log.*d*("Lifecycle", "onStop");  
 }  
 protected void onDestroy() {  
 super.onDestroy();  
 Log.*d*("Lifecycle", "onDestroy");  
 }  
 protected void onRestart() {  
 super.onRestart();  
 Log.*d*("Lifecycle", "onRestart");  
 }  
}

**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:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 <FrameLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/f1"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"/>  
</androidx.constraintlayout.widget.ConstraintLayout>

**BlankFragment.java**

package com.example.practical5;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
  
public class BlankFragment extends Fragment {  
 private static final String *TAG* = "HelloWorldFragment";  
 @Nullable  
 @Override  
 public View onCreateView(@NonNull LayoutInflater inflater, @Nullable  
 ViewGroup container, @Nullable Bundle savedInstanceState) {  
 Log.*d*("fragmentlifecycle", "onCreateView called");  
 return inflater.inflate(R.layout.*fragment\_blank*, container,  
 false);  
 }

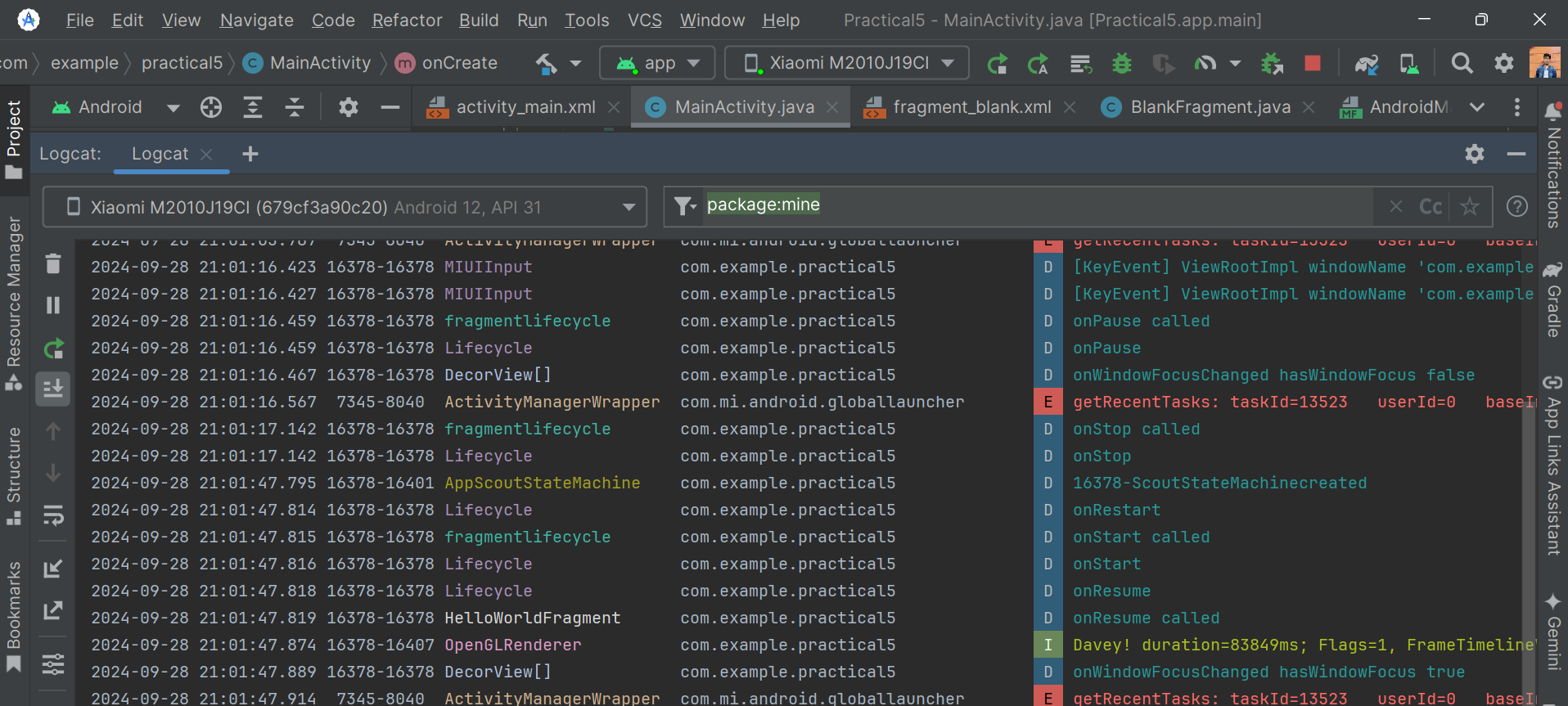
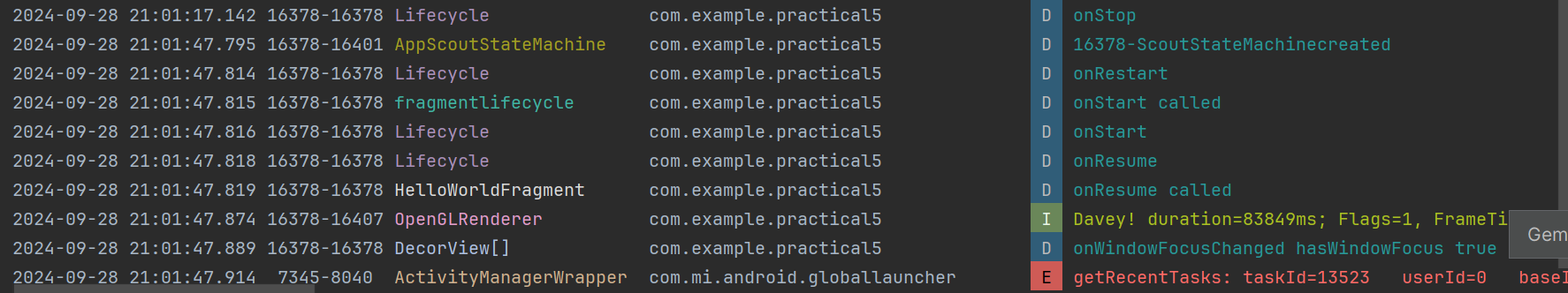
@Override  
 public void onViewCreated(@NonNull View view, @Nullable Bundle  
 savedInstanceState) {  
 super.onViewCreated(view, savedInstanceState);  
 Log.*d*("fragmentlifecycle", "onViewCreated called");  
 }  
 @Override  
 public void onStart() {  
 super.onStart();  
 Log.*d*("fragmentlifecycle", "onStart called");  
 }  
 @Override  
 public void onResume() {  
 super.onResume();  
 Log.*d*(*TAG*, "onResume called");  
 }  
 @Override  
 public void onPause() {  
 super.onPause();  
 Log.*d*("fragmentlifecycle", "onPause called");  
 }  
 @Override  
 public void onStop() {  
 super.onStop();  
 Log.*d*("fragmentlifecycle", "onStop called");  
 }  
 @Override  
 public void onDestroyView() {  
 super.onDestroyView();  
 Log.*d*("fragmentlifecycle", "onDestroyView called");  
 }  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 Log.*d*("fragmentlifecycle", "onDestroy called");  
 }  
}

**fragment\_blank.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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"  
 tools:context=".BlankFragment">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="@string/hello\_blank\_fragment" />  
</FrameLayout>

**Output:**

A screen shot of a computer

Description automatically generated

**Experiment 6**

**Title: Create Sign in and Sign up screen using Toast and Recycler view.**

**MainActivity.java**

package com.example.practical3;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
 Button btn1, btn2;  
 @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;  
 });  
 btn1 = findViewById(R.id.*toLogin*);  
 btn2 = findViewById(R.id.*toRegister*);  
  
 btn1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {  
 Intent intent = new Intent(MainActivity.this, LoginActivity.class);  
 startActivity(intent);  
 }  
 });  
 btn2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MainActivity.this, RegistrationActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 }  
}

**activity\_main.xml**

<?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:id="@+id/main"  
android:layout\_height="match\_parent"  
android:orientation="vertical"  
android:padding="16dp"  
android:gravity="center"  
tools:context=".MainActivity">  
<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Home"  
 android:textSize="40sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="20dp" />  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp">

<Button  
 android:id="@+id/toLogin"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 android:background="#f5602f"  
 android:text="Login"  
 android:textColor="#212121" />  
 <Button  
 android:id="@+id/toRegister"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 android:background="#f5602f"  
 android:text="Register"  
 android:textColor="#212121" />  
</LinearLayout>  
</LinearLayout>

**LoginActivity.java**

**Login.xml**

**RegistrationActivity.java**

**Registration.xml**

These four file is almost same as Practical 3

**Build.gradle - implementation 'androidx.recyclerview:recyclerview:1.2.1'**

**LandingPageActivity.java**

package com.example.practical3;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;  
import java.util.ArrayList;  
import java.util.List;

public class LandingPageActivity extends AppCompatActivity {  
 private RecyclerView recyclerView;  
 private List<String> dataList;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*landing\_page*);  
 recyclerView = findViewById(R.id.*recyclerView2*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
 initializeData();  
 recyclerView.setAdapter(new MyAdapter(dataList));  
 }  
  
 private void initializeData() {  
 dataList = new ArrayList<>();  
 dataList.add("Item 1");  
 dataList.add("Item 2");  
 dataList.add("Item 3");  
 dataList.add("Item 4");  
 dataList.add("Item 5");  
 dataList.add("Item 6");  
 dataList.add("Item 7");  
 dataList.add("Item 8");  
 dataList.add("Item 9");  
 dataList.add("Item 10");  
 dataList.add("Item 11");  
 dataList.add("Item 12");  
 dataList.add("Item 13");  
 dataList.add("Item 14");  
 dataList.add("Item 15");  
 dataList.add("Item 16");  
 dataList.add("Item 17");  
 dataList.add("Item 18");  
 dataList.add("Item 19");

dataList.add("Item 20");  
 }  
 public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {  
  
 private List<String> dataList;  
 public MyAdapter(List<String> dataList) {  
 this.dataList = dataList;  
 }  
 @Override  
 public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*item\_text*, parent, false);  
 return new MyViewHolder(view);  
 }  
 @Override  
 public void onBindViewHolder(MyViewHolder holder, int position) {  
 holder.textView.setText(dataList.get(position));  
 }  
 @Override  
 public int getItemCount() {  
 return dataList.size();  
 }  
 public class MyViewHolder extends RecyclerView.ViewHolder {  
 TextView textView;  
 public MyViewHolder(View itemView) {  
 super(itemView);  
 textView = itemView.findViewById(R.id.*textView*);  
 } }

}}

**Landing\_page.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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".LandingPageActivity">  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="10dp"  
 android:layout\_weight="1" />  
 <LinearLayout  
 android:id="@+id/item\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:text="My name is Asif Alam" />  
 </LinearLayout>  
</LinearLayout>

**item.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="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"

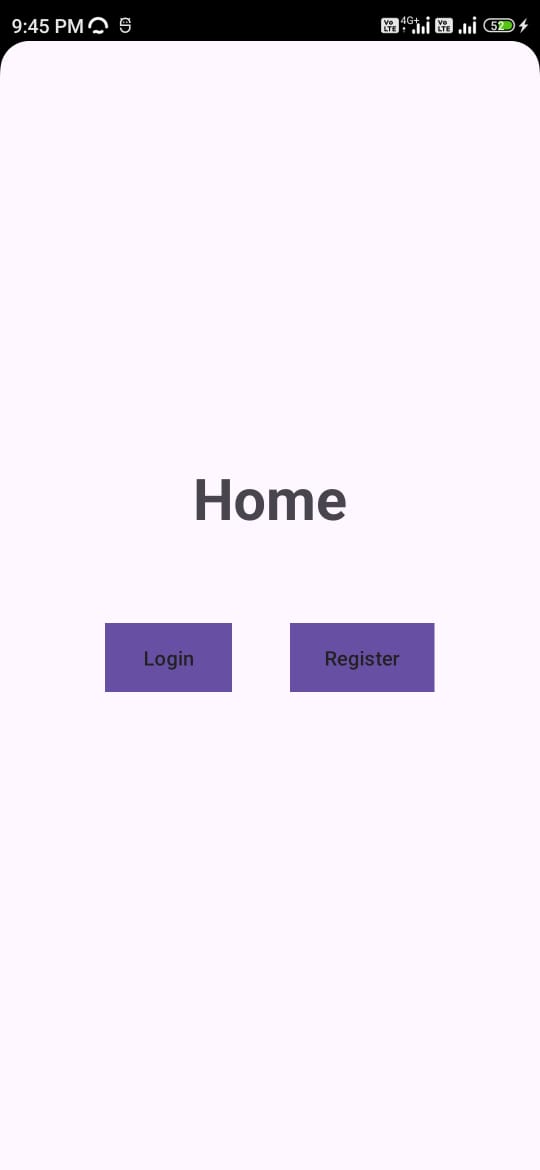
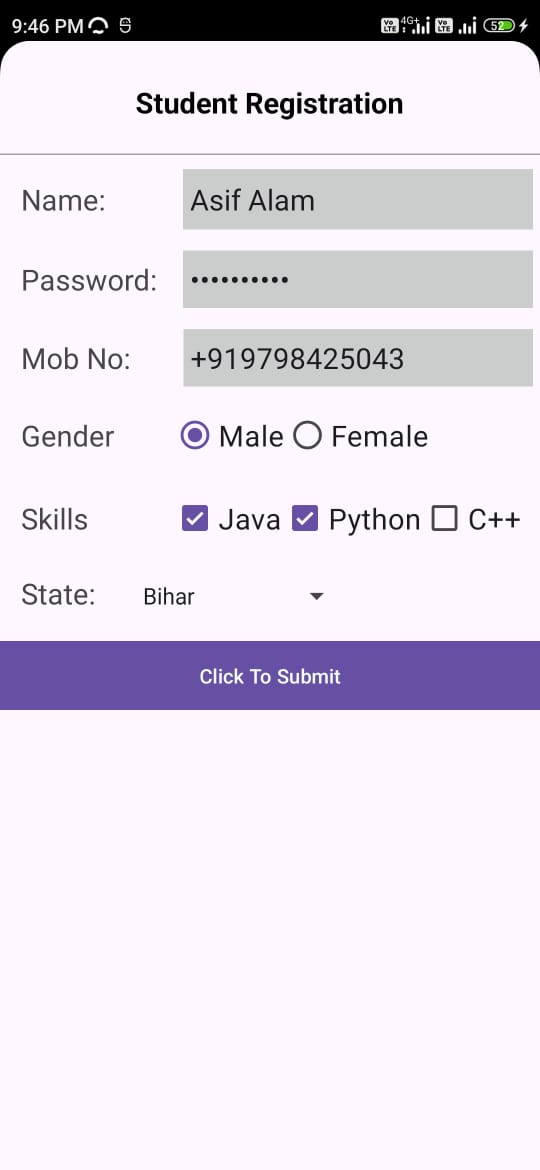
android:textSize="18sp" />  
</LinearLayout>

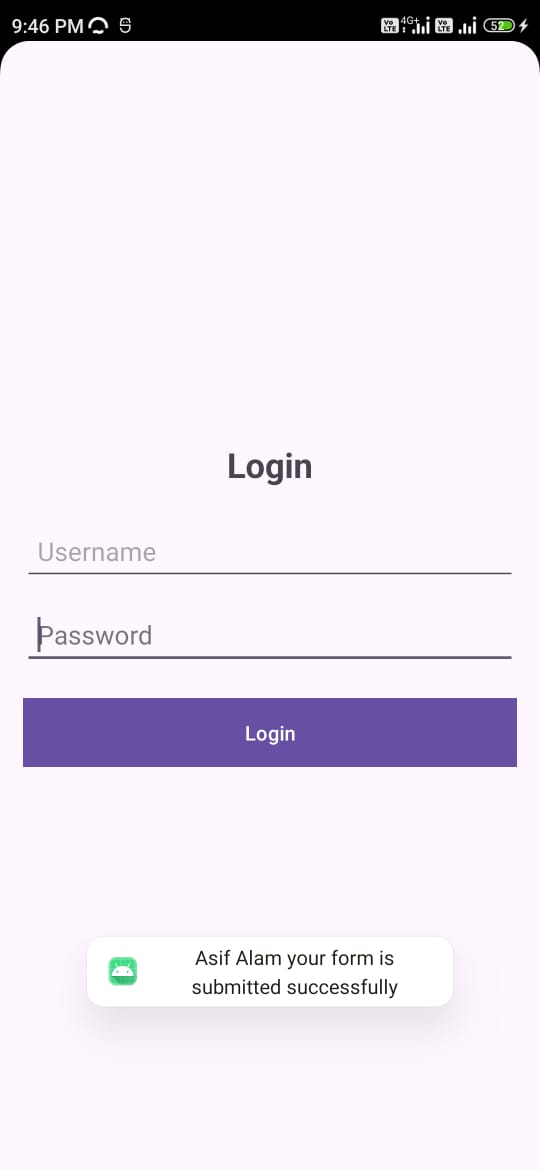
**AndroidManifest.xml**

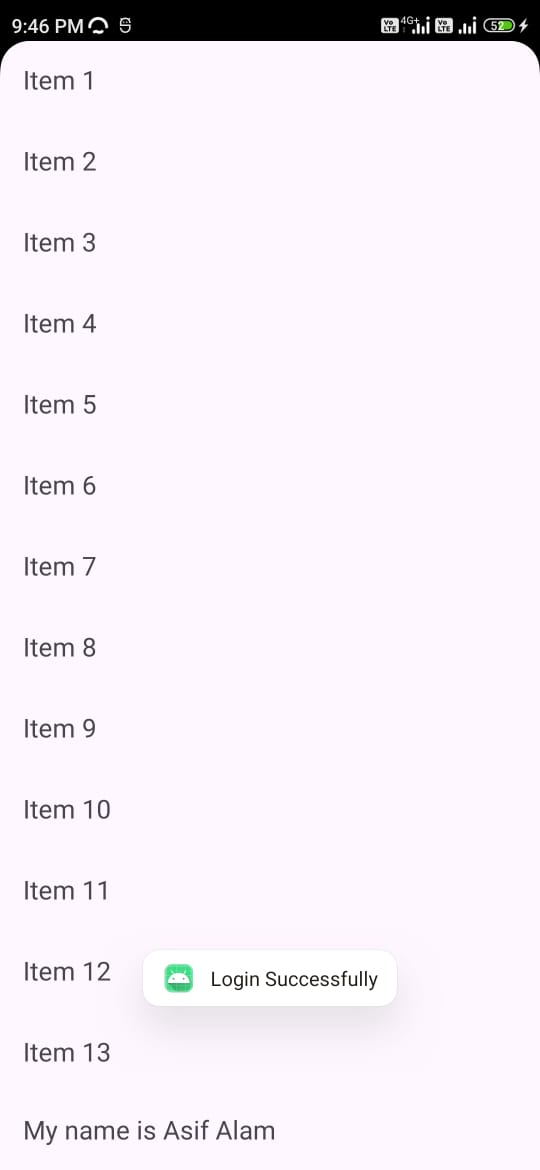
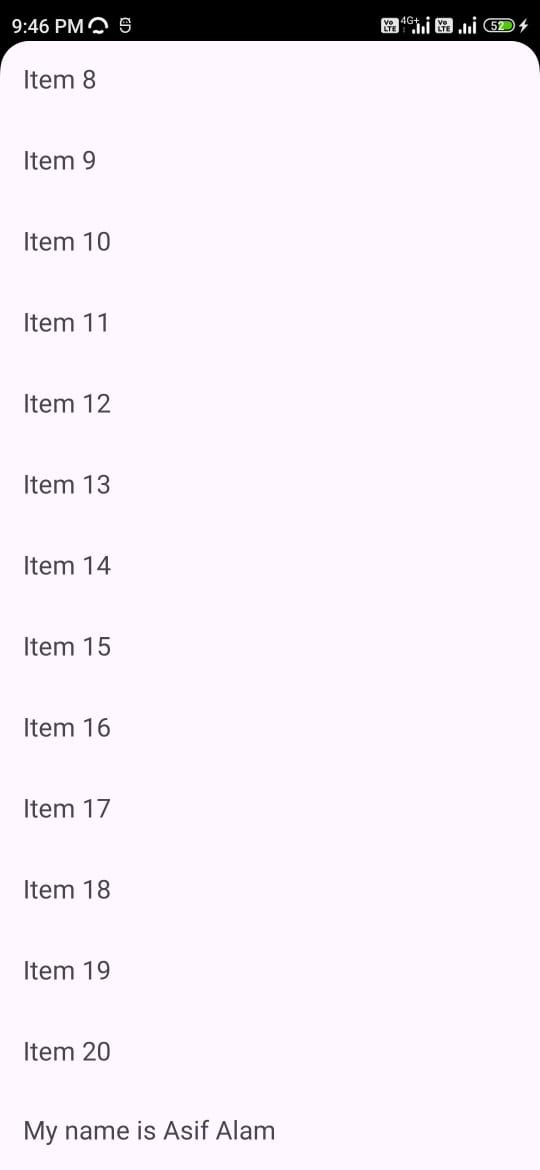
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Practical3"  
 tools:targetApi="31">  
 <activity android:name=".LandingPageActivity"  
 android:exported="false" />  
 <activity android:name=".LoginActivity"  
 android:exported="true" />  
 <activity android:name=".RegistrationActivity"  
 android:exported="true" />  
 <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>  
 </application>  
</manifest>

**Strings.xml**

<resources>  
 <string name="app\_name">Practical 3</string>  
 <string-array name="states\_array">  
 <item>Bihar</item>  
 <item>Gujarat</item>  
 <item>West Bengal</item>  
 </string-array>  
</resources>

**Output:**





**Experiment 7**

**Title: Demonstration of Android Activity and Fragment life cycle for “Hello World” application.**