1. Develop a Simple Android Application displaying “Hello World”:

Project Name: HelloWorld

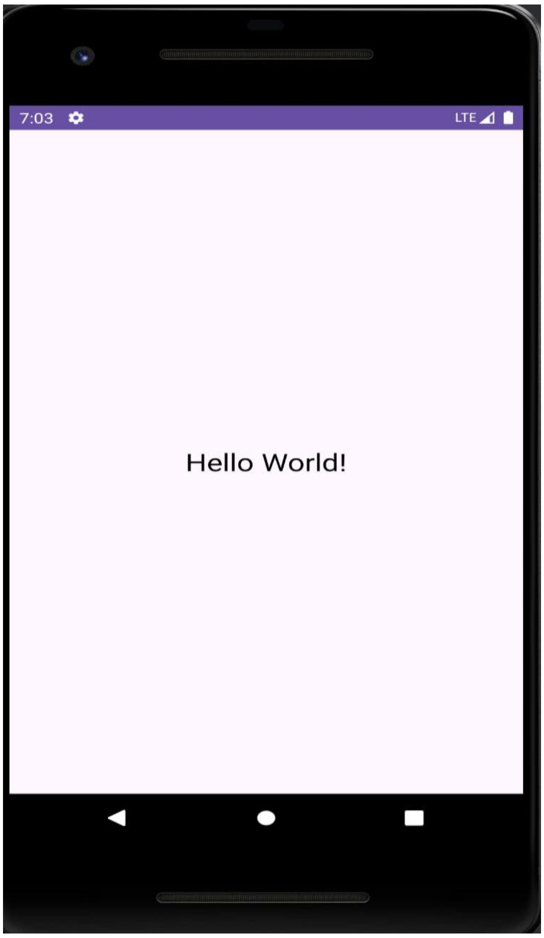
activity_main.xml:

```
<?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="match_parent"
        android:orientation="vertical"
        android:gravity="center">

        <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="24sp"
        android:textColor="#000000"/>

</LinearLayout>
```

MainActivity.java:

```
        package com.example.helloworld;

        import android.os.Bundle;

        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {

        @Override

        protected void onCreate(Bundle savedInstanceState)

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        }
        }
```

Output:



Hello World!

2. Develop an Android Application using “Button” Component:

Project Name: Buttons

activity_main.xml:

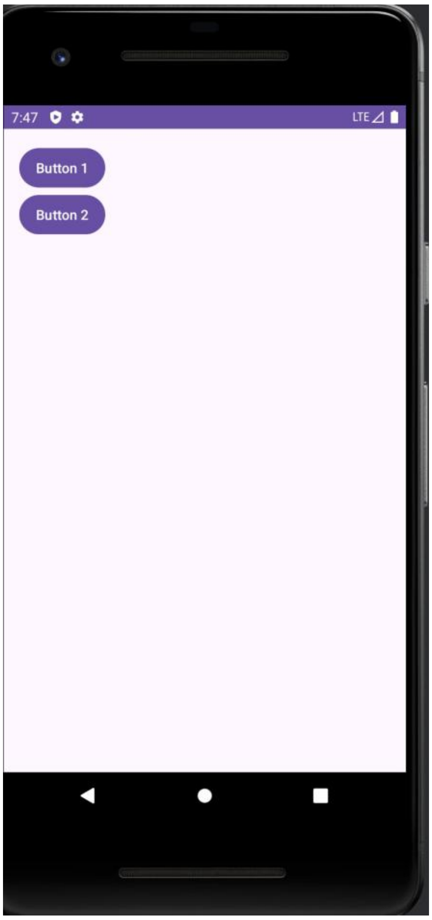
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

<!-- First Button -->

        <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:padding="8dp" />

<!-- Second Button -->

        <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2"
        android:padding="8dp" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.buttons;

        import android.os.Bundle;
        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);

}
}
```

Output:



3. Develop an Android Application by implementing “Vertical Linear Layout”:

Project Name: VerticalLinearLayout

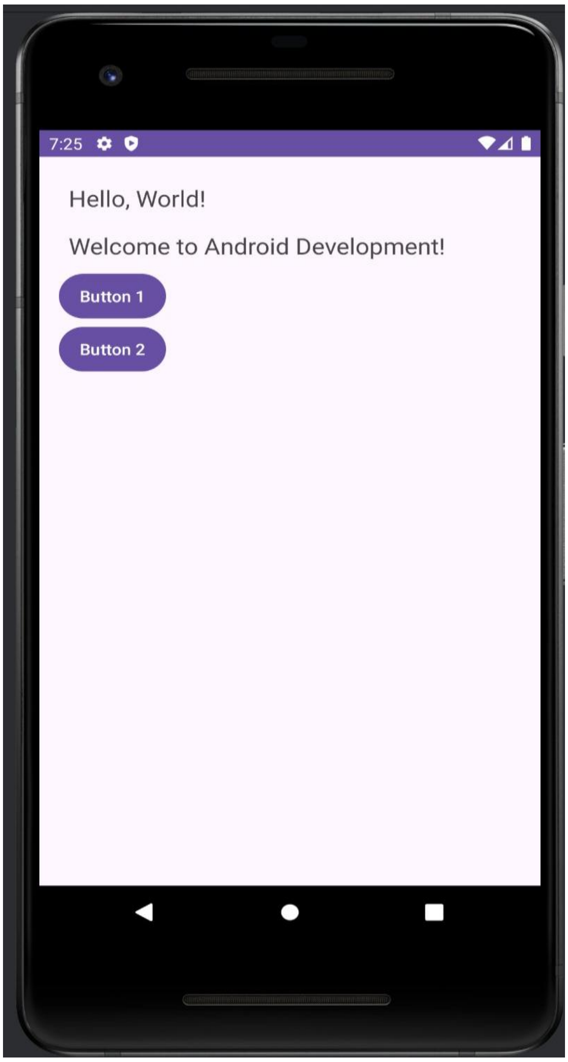
activity_main.xml:

```
<?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="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, World!"
        android:textSize="20sp"
        android:padding="8dp" />

        <TextView
        android:id="@+id/textview2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Android Development!"
        android:textSize="20sp"
        android:padding="8dp" />

<!-- First Button -->

        <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:padding="8dp" />

<!-- Second Button -->

        <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" a
        ndroid:text="Button 2"
        android:padding="8dp" />

</LinearLayout>
```

MainActivity.java:

```
package com.example.verticallinearlayout;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main); // Set the content view to the XML layout  with the vertical LinearLayout

}
}
```

Output:



Hello, World!

Welcome to Android Development!

4. Develop an Android Application by implementing “Horizontal Linear Layout”:

Project Name: HorizontalLinearLayout

activity_main.xml:

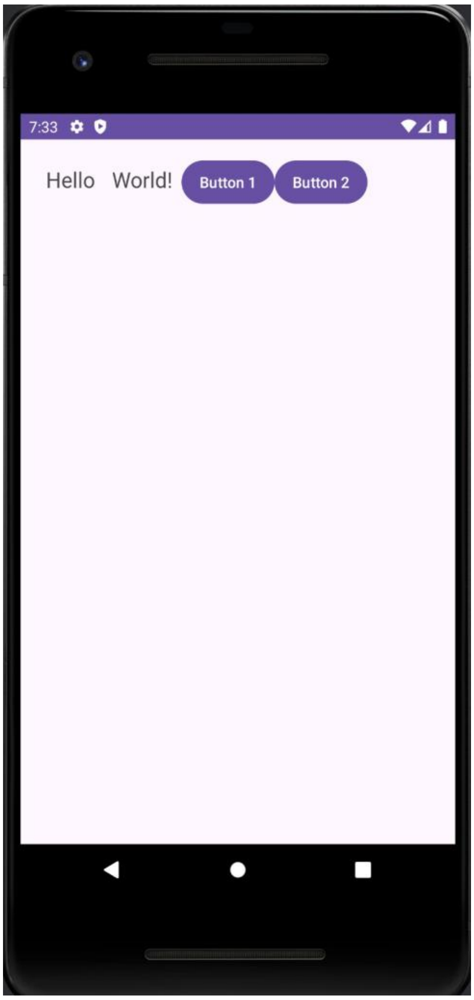
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:padding="16dp">

        <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello"
        android:textSize="20sp"
        android:padding="8dp" />

        <TextView
        android:id="@+id/textview2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="World!"
        android:textSize="20sp"
        android:padding="8dp" />

<!-- First Button -->

        <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:padding="8dp" />

<!-- Second Button -->

        <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="Button 2"
    android:padding="8dp" />

</LinearLayout>
```

MainActivity.java:

```
    package com.example.horizontallinearlayout;

    import android.os.Bundle;
    import androidx.appcompat.app.AppCompatActivity;

    public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main); // Set the content view to the
    XML layout with the horizontal LinearLayout

    }
    }
```

Output:



ImageView: It is used to display images in an app. It allows you to show pictures, icons, or other graphics in your application's interface.

5. Develop an Android Application using “ImageView” Component:

Project Name: ImageViews

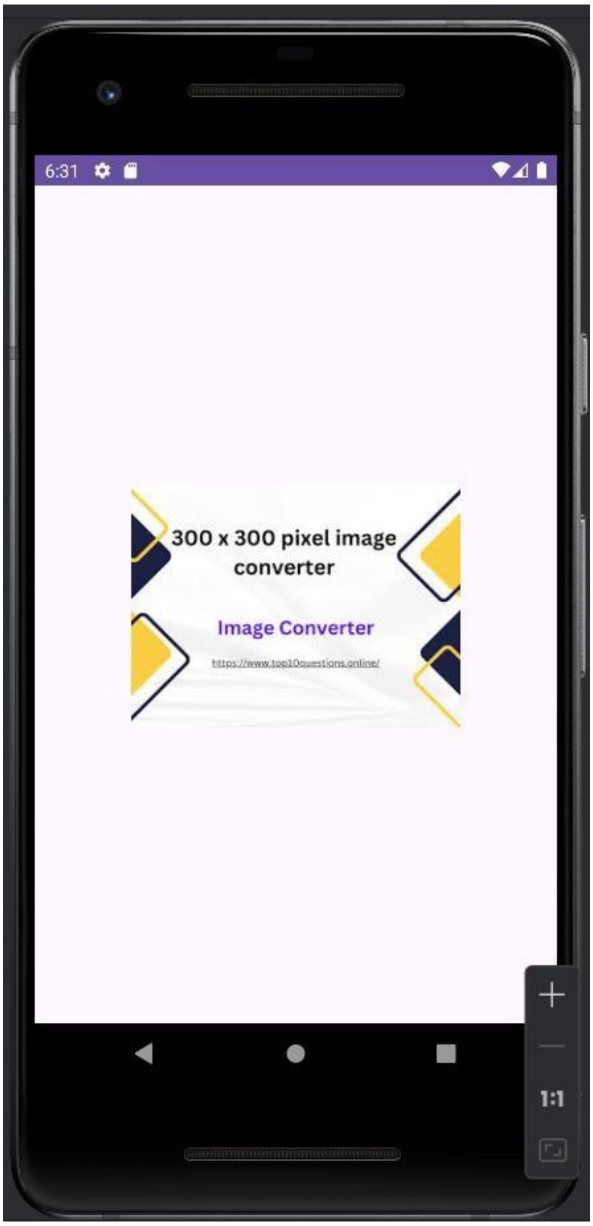
activity_main.xml:

```
<?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="match_parent"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="16dp">

<!-- ImageView -->

        <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/sample_image"
        android:contentDescription="Sample Image" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.imageviews;

        import android.os.Bundle;
        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        }
        }
```

Output:


6:31
300 x 300 pixel image
converter
Image Converter
https://www.top10questions.online/
1:1

Edit Text: It is a user interface component that allows users to input and edit text. It's similar to a text box in other programming environments.

6. Develop an Android Application using "EditText" Component:

Project Name: EditTexts

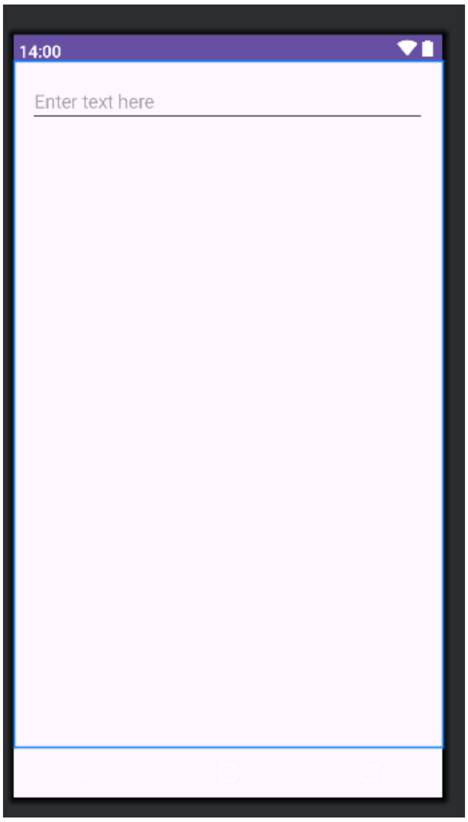
activity_main.xml:

```
<?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="match_parent"
            android:orientation="vertical"
            android:padding="16dp">

  <!-- EditText -->

            <EditText
              android:id="@+id/editText"
              android:layout_width="match_parent"
              android:layout_height="wrap_content"
              android:hint="Enter text here"
              android:inputType="text" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.edittexts;

        import android.os.Bundle;
        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);



        }
        }
```

Output:



Radio Button: It is commonly used UI component in Android that allow users to select one option from a set of mutually exclusive choices.

7. Develop an Android Application using "Radio Button" Component:

Project Name: RadioButtonExample

activity_main.xml:

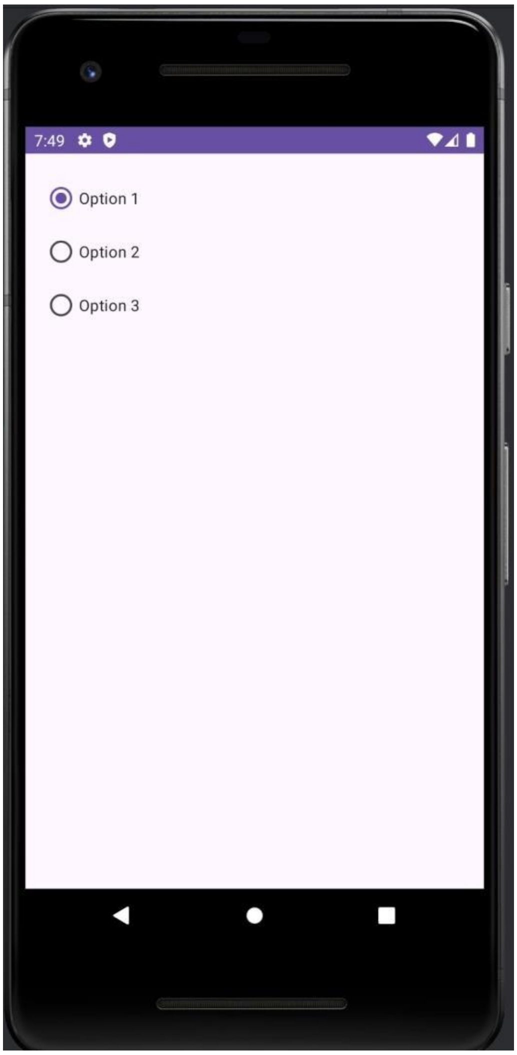
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

<!-- RadioGroup to group RadioButtons -->

        <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

<!-- RadioButton options -->

        <RadioButton
        android:id="@+id/radioButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 1" />

        <RadioButton
        android:id="@+id/radioButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 2" />

        <RadioButton
        android:id="@+id/radioButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 3" />

        </RadioGroup>
```

```
</LinearLayout>
```

MainActivity.java:

```
package com.example.radiobuttonexample;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

// No implementation action needed

}
}
```

Output:



Task:

Develop an android Application to display your college name in the Centre of the screen.

TextView: It is used to display text on the screen. It's often used to show labels, instructions, or any other text content in an app.

8. Develop an Android Application using "TextView" Component:

Project Name: TextViews

activity_main.xml:

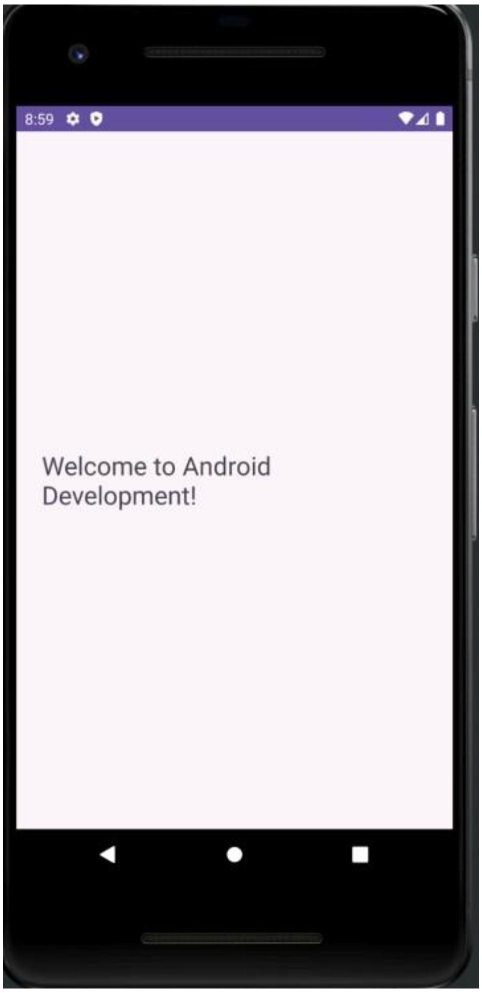
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="16dp">

<!-- TextView -->

        <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, World!"
        android:textSize="24sp"
        android:padding="8dp" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.textviews;

        import android.os.Bundle;
        import android.widget.TextView;
        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
// Find the TextView by its ID
TextView textView = findViewById(R.id.textView);

textView.setText("Welcome to Android Development!"); // Change the
text programmatically

}
}
```

Output:



Welcome to Android
Development!

Checkbox: It is an UI Element which allows users to select or deselect an option. It is typically used when you want to let users select multiple options from a set. When checked, it returns true; when unchecked, it returns false.

9. Develop an Android Application using “CheckBox” Component:

Project Name: CheckBoxExample

activity_main.xml:

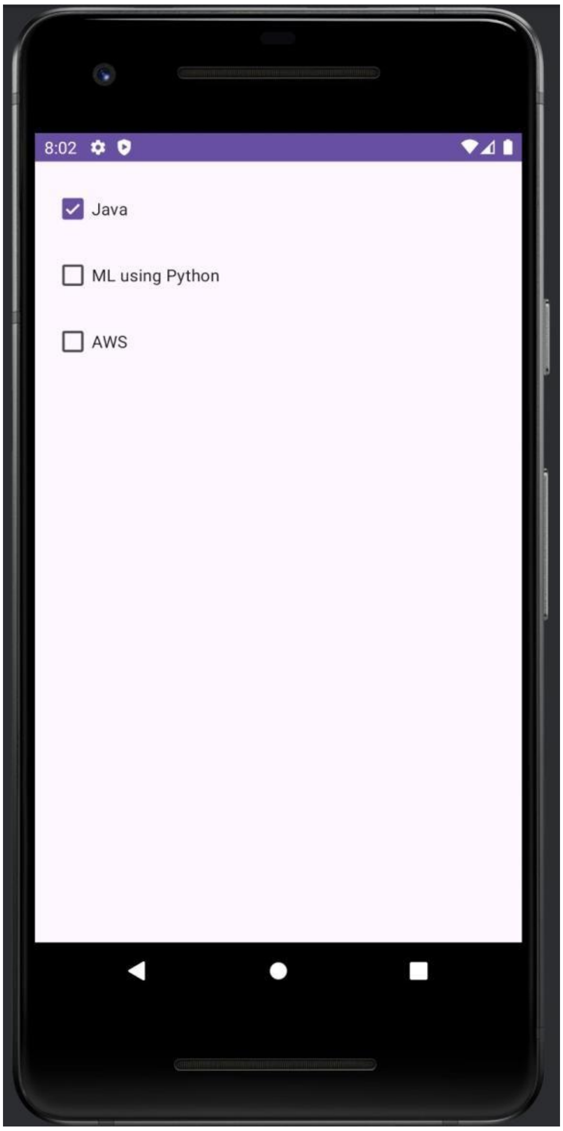
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

<!-- Checkbox options -->

        <CheckBox
        android:id="@+id/checkBoxJava"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Java" />

        <CheckBox
        android:id="@+id/checkBoxML"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ML Using Python" />

        <CheckBox
        android:id="@+id/checkBoxAWS"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="AWS" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.checkboxexample;

        import android.os.Bundle;
```

```
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

// No implementation action needed

}
}
```

Output:



☑ Java

□ ML using Python

□ AWS

Toggle Button: It means to switch between two states, like turning something on or off.

10. Develop an Android Application using “Toggle Button” Component:

Project Name: ToggleButtonExample

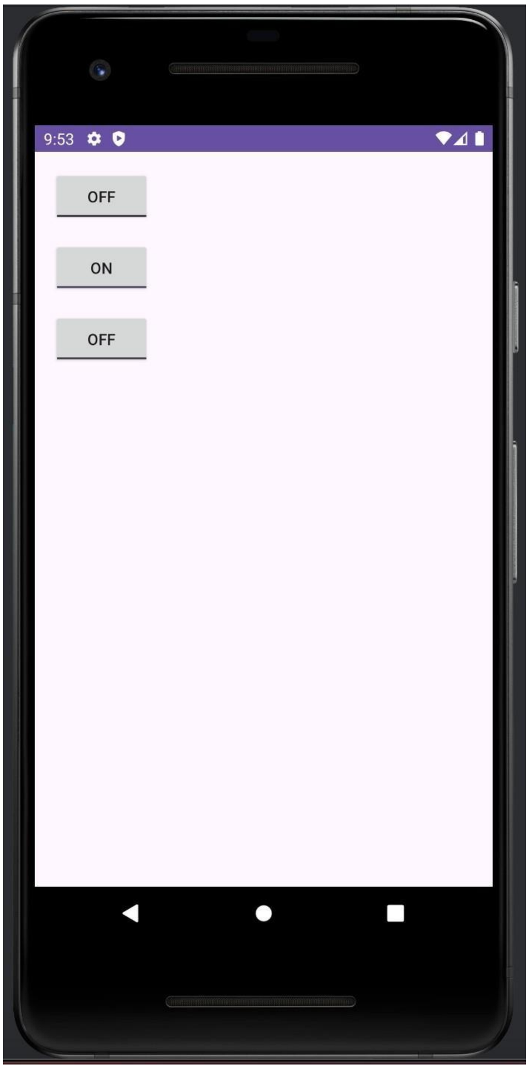
activity_main.xml:

```
<?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="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

    <!-- First ToggleButton -->

        <ToggleButton
            android:id="@+id/toggleButton1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textOn="On"
            android:textOff="Off"/>

    <!-- Second ToggleButton -->

        <ToggleButton
            android:id="@+id/toggleButton2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textOn="Off"
            android:textOff="on" />

    <!--Third ToggleButton -->

        <ToggleButton
            android:id="@+id/toggleButton3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textOn="On"
            android:textOff="Off" />

</LinearLayout>
```

MainActivity.java:

```
package com.example.togglebuttonexample;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

// No implementation action needed

}
}
```

Output:



Tasks:

11. Design an XML layout with a TextView at the top, three EditText fields below it (for Name, Email, and Phone Number), and a Button labeled "Register," all centered vertically on the screen.

12. Design an XML layout with an ImageView at the top displaying a logo, followed by two RadioButtons for selecting a "Beginner" or "Advanced" level, and a Button labeled "Start Challenge" at the bottom, all elements centered horizontally.

13. Design an XML layout with a TextView at the top displaying "User Feedback", an EditText field below it for user input, and two Buttons at the bottom labeled "Submit" and "Cancel".

14. Design an XML layout with a TextView displaying "Profile Setup" at the top, an ImageView for a profile picture placeholder below it, three EditText fields for Name, Email, and Bio, all arranged in a visually appealing manner.

Toast Message: It is a small, temporary pop-up message on the screen for shorter period. It's often used to give quick feedback or show a brief notification to the user, like "Message sent" or "Saved successfully."

15. Develop an Android Application using "Toast Message" Component:

Project Name: ToastDemo

activity_main.xml:

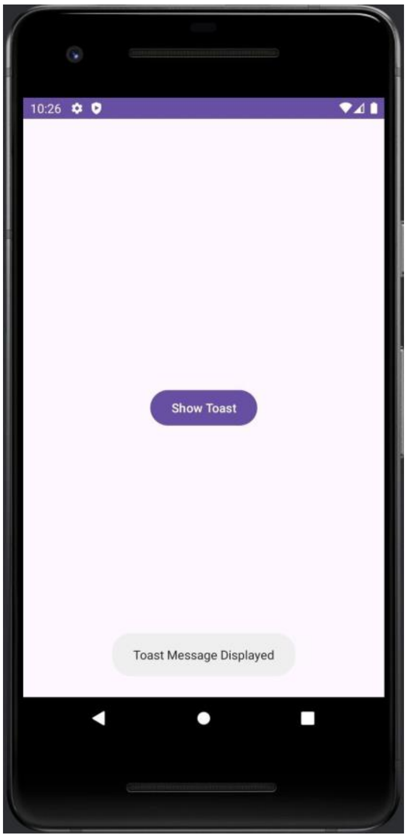
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

        <Button
        android:id="@+id/show_toast_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Toast"
        android:layout_gravity="center" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.toastdemo;

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

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the Button in the layout
```

```
Button showToastButton = findViewById(R.id.show_toast_button);

// Set an OnClickListener to the Button
showToastButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Show a Toast message when the Button is clicked
Toast.makeText(MainActivity.this, "Toast Message Displayed",
Toast.LENGTH_SHORT).show();
}});
}
}
```

Output:



Snack Bar Messages: A brief message that appears at the bottom of the screen to notify users about an event or action button (like "UNDO" or "RETRY"), prompts user to take an immediate action. It usually disappears automatically after a few seconds.

16. Develop an Android Application using "Snack Bar Message" Component:

Project Name: SnackBarDemo

activity_main.xml:

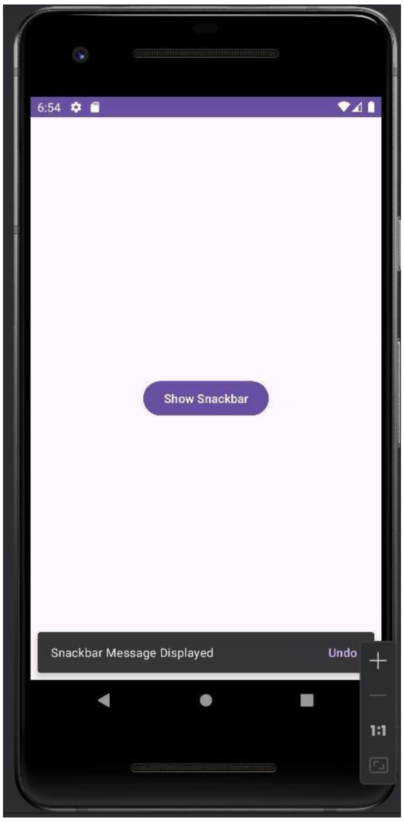
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:orientation="vertical"
        android:gravity="center">

        <Button
        android:id="@+id/show_snackbar_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Snackbar" />

</LinearLayout>
```

MainActivity.java:

```
        package com.example.snackbardemo;

        import android.os.Bundle;
        import android.view.View;
        import android.widget.Button;
        import com.google.android.material.snackbar.Snackbar;
        import androidx.appcompat.app.AppCompatActivity;

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the Button in the layout
```

```
Button showSnackbarButton = findViewById(R.id.show_snackbar_button);

// Set an OnClickListener to the Button
showSnackbarButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Show a Snackbar message when the Button is clicked
Snackbar.make(v, "Snackbar Message Displayed",
Snackbar.LENGTH_LONG).setAction("Undo", new View.OnClickListener() {
@Override
public void onClick(View v) {

// Handle the Undo action

}
}).show();
}
});
}
}
```

Output:



Snackbar Message Displayed

Dialogue Message: It is a small window that pops up on the screen to interact with the user. It can show a message, ask for input, or provide options like "OK" or "Cancel." Dialog boxes help in getting a quick response or confirmation from the user.

17. Develop an Android Application using "Dialogue Message" Component:

Project name: DialogDemo

activity_main.xml:

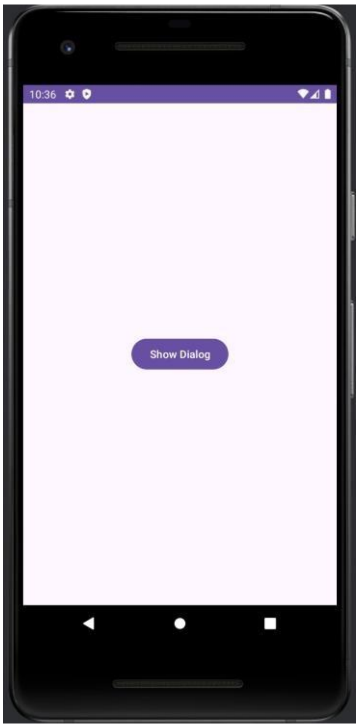
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center">

        <Button
        android:id="@+id/show_dialog_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Dialog" />

</LinearLayout>
```

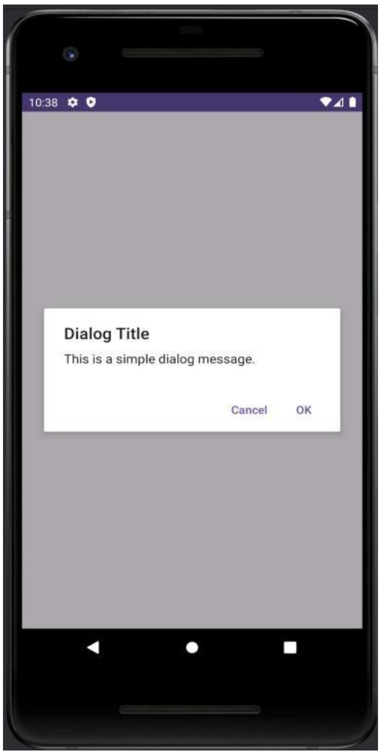
MainActivity.java:

```
        package com.example.dialogdemo;

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

        public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the Button in the layout
        Button showDialogButton = findViewById(R.id.show_dialog_button);
```

```
// Set an OnClickListener to the Button
showDialogButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Create and show an AlertDialog
new AlertDialog.Builder(MainActivity.this)
.setTitle("Dialog Title")
.setMessage("This is a simple dialog message.")
.setPositiveButton("OK", null)
.setNegativeButton("Cancel", null)
.show();
}
});
}
}
```

Output:




Dialog Title

This is a simple dialog message.