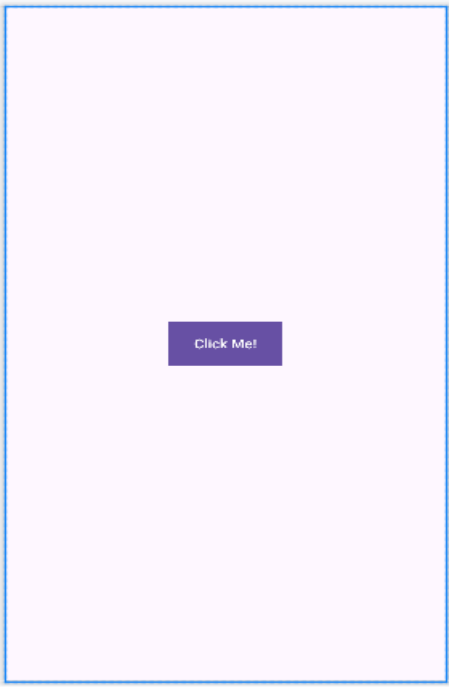
1. Creating “Hello world” Application.
    1. Click **Start →Android Studio**, a **Welcome to Android Studio** dialog box will appear. Click **New Project**, the **New Project Dialog box** appears.
    2. Choose **Empty Views Activity** then click **Next**.
    3. Specify the **Name** of your project, Select the **Language** as **Java**, and Select the **Minimum SDK** as **API 16 (“Jelly Bean”, Android 4.1)**. Click **Finish** Button.
    4. Create a **Button** resource in **activity_main.xml** and update the following code

```
<?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">
    <Button
        android:id="@+id/hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#535538"
        android:text="Click Me!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

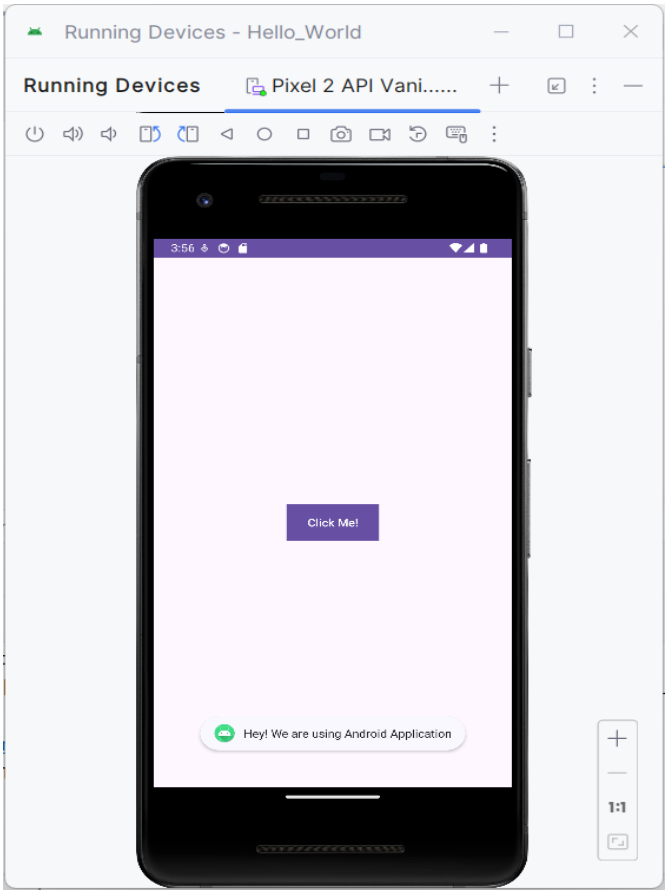
The following figure illustrates the design view of the application.



5. Create a **Button** object, create **clickListener**, **onClick** event and update the following code in **MainActivity.java**

```
package com.example.hello_world;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button b;
        b=findViewById(R.id.hello);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(MainActivity.this, "Hey! We are using Android Application", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

6. Click **Run app** or **shift+F10** to execute the application.
   Output:

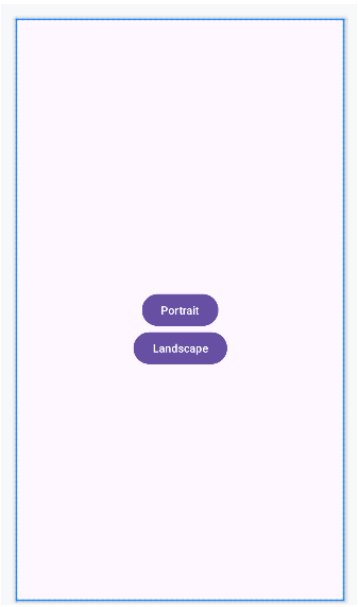


2. Creating an application that displays message based on the screen orientation.

1. Click **Start →Android Studio**, a **Welcome to Android Studio** dialog box will appear. Click **New Project**, the **New Project Dialog box** appears.
2. Choose **Empty Views Activity** then click **Next**.
3. Specify the **Name** of your project, Select the **Language** as **Java**, and Select the **Minimum SDK** as **API 16 ("Jelly Bean", Android 4.1)**. Click **Finish** Button.
4. Create two **Button** resources in **activity_main.xml** and update the following code.

```
<?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=".MainActivity">

    <Button
        android:id="@+id/por"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Portrait"
        android:layout_centerInParent="true"/>
    <Button
        android:id="@+id/lan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Landscape"
        android:layout_below="@id/por"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

The following figure illustrates the design view of the application.

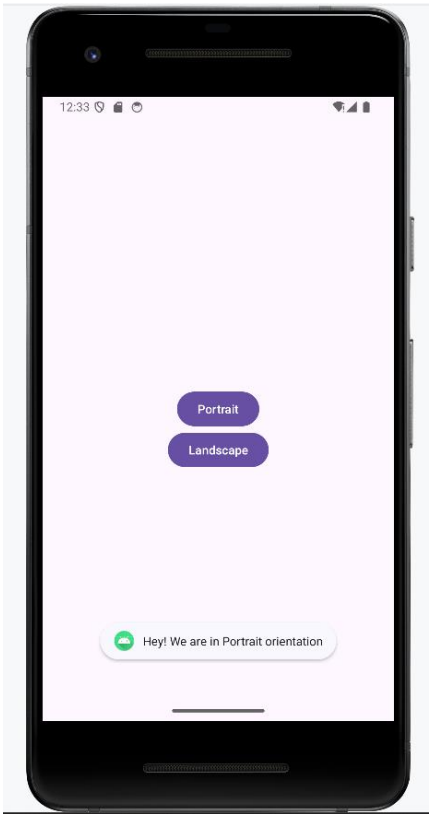


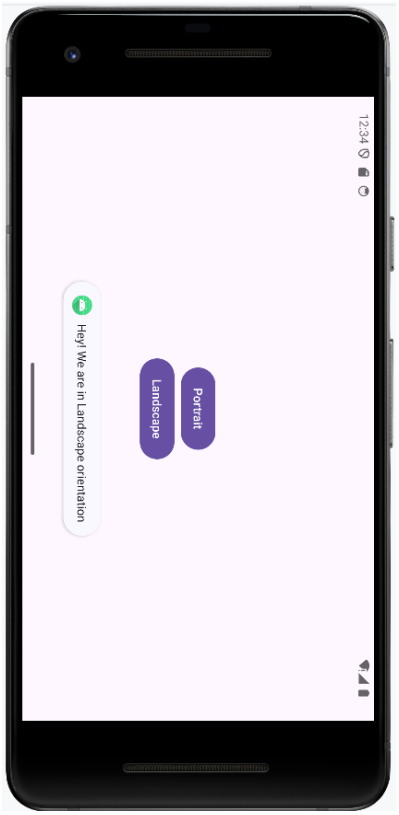
5. Create two **Button** object, create **clickListener**, **onClick** event and update the following code in **MainActivity.java**

```
package com.example.screen;

import android.content.pm.ActivityInfo;
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 {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        Button l,p;
        l=findViewById(R.id.lan);
        p=findViewById(R.id.por);
        l.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
                Toast.makeText(MainActivity.this, "Hey! We are in Landscape
orientation", Toast.LENGTH_SHORT).show();
            }
        });
        p.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
                Toast.makeText(MainActivity.this, "Hey! We are in Portrait orientation",
Toast.LENGTH_SHORT).show();
            }
        });

    }
}
```

6. Click **Run app** or **shift+F10** to execute the application.

Output





3. Create an application to develop Login window using UI controls.

1. Click **New Project**, the **New Project Dialog box** appears.
2. Choose **Empty Views Activity** then click **Next**.
3. Specify the **Name** of your project, Select the **Language** as **Java**, and Select the **Minimum SDK** as **API 16 ("Jelly Bean", Android 4.1)**. Click **Finish** Button.
4. Create **background** resources(**bg_outer.xml**, **bg_inner.xml**)
    a. To create resource file click **app→res→drawable.** Right click **drawable→New→ Drawable Resource File.** The **New Resource File** dialog box appears.
    b. Set **filename** as **bg_outer.xml**, **root element** as **shape** and then click **ok.** Modify the bg_outer.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
   <corners android:radius="12dp" />
   <gradient
      android:startColor="#B388FF"
      android:endColor="#397C9A"
      android:angle="100"/>
</shape>
```

c. Likewise, create another background resource for inner layout. Set **filename** as **bg_inner.xml**, **root element** as **shape** and then click **ok.** Modify the bg_outer.xml file

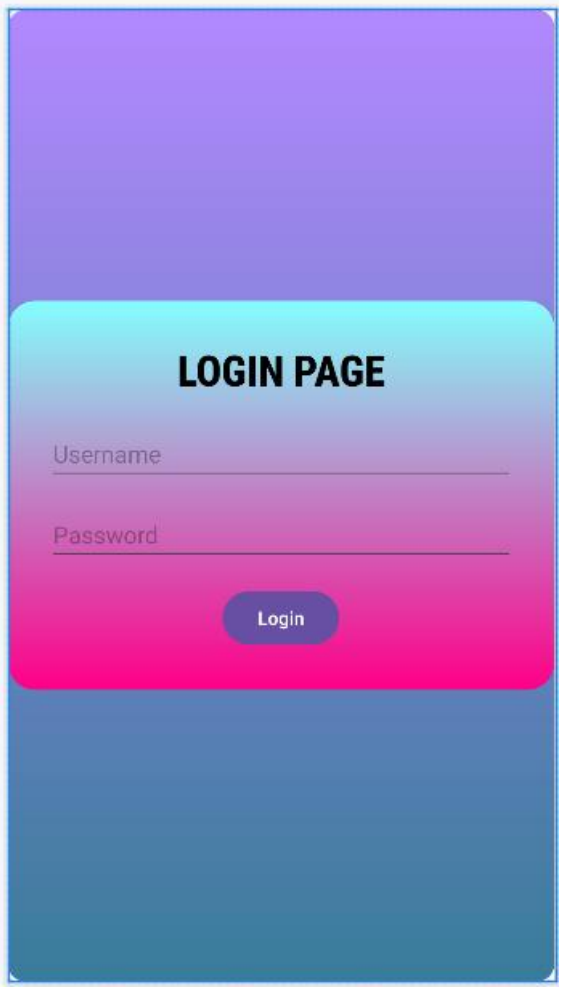
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
<gradient
   android:startColor="#84FFFF"
   android:endColor="#f08"
   android:angle="100"/>
   <corners android:radius="20dp"/>

</shape>
```

5. Create **two EditText box** and **a Button** resource in **activity_main.xml** and update the following code.

```
<?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=".MainActivity"
   android:background="@drawable/bg_outer">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:layout_centerInParent="true"
      android:orientation="vertical"
      android:background="@drawable/bg_inner"
      android:padding="30dp"
      >
      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="LOGIN PAGE"
         android:textSize="32sp"
         android:textStyle="bold"
         android:fontFamily="sans-serif-condensed-medium"
         android:textColor="@color/black"
         android:paddingBottom="20dp"
         />

   <EditText
      android:id="@+id/editTextUsername"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:hint="Username"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:layout_below="@id/editTextUsername"
        android:layout_marginBottom="16dp"
        android:inputType="textPassword"/>

    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_below="@id/editTextPassword"/>
    </LinearLayout>
</RelativeLayout>
```

The following figure illustrates the design view of the application.

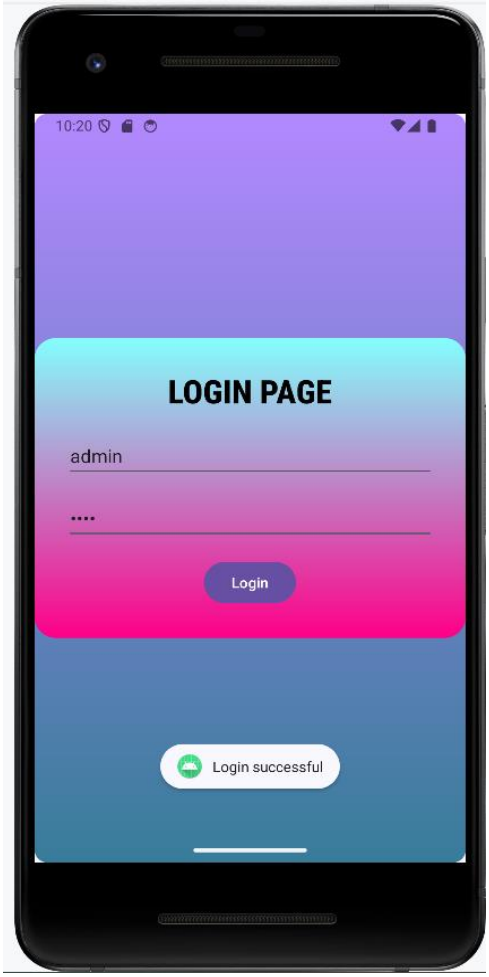


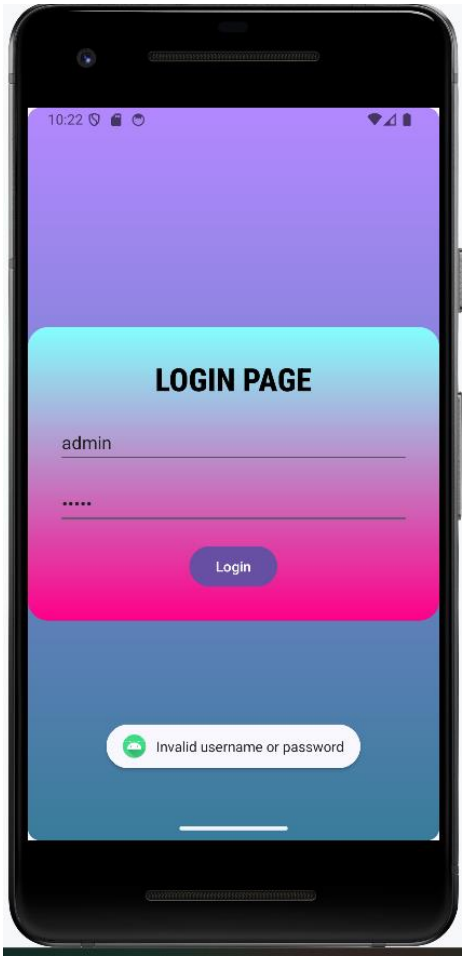
6. Create two **EditText** and a **Button** object, create **clickListener**, **onClick** event for button object and update the following code in **MainActivity.java**

```
package com.example.controls;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
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 {
    private EditText editTextUsername,editTextPassword;
    private Button buttonLogin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        editTextUsername = findViewById(R.id.editTextUsername);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonLogin = findViewById(R.id.buttonLogin);
        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = editTextUsername.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();

                if(username.equals("admin") && password.equals("pass")){
                    Toast.makeText(MainActivity.this, "Login successful",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Invalid username or password",
Toast.LENGTH_SHORT).show();
                }
            }
        });

    }

}
```

7. Click **Run app** or **shift+F10** to execute the application.

Output


10:20
LOGIN PAGE
admin
....
Login
Login successful


10:22
LOGIN PAGE
admin
.....
Login
Invalid username or password