School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning
(Learning by Doing and Discovery)

Name of the Experiement :

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Open Android Studio
- **Launch Android Studio and create a new project or open an existing one.**

Create or Edit the XML Layout File
- **In your project's res/layout folder, create or modify an XML file for the login page.**
- **Add the following UI elements:**
- **EditText for Username**
- **EditText for Password**
- **Button for Login**

Go to the Corresponding Activity/Fragment
- **Navigate to the .java or .kt file associated with your layout.**

Initialize UI Components in Code
- **Use findViewById() to reference the username field, password field, and login button.**

Set an OnClickListener for the Login Button
- **Inside the listener, validate the entered credentials.**
- **If credentials are incorrect or empty, proceed to create an Alert Dialog.**

Create the AlertDialog Builder
- **Instantiate AlertDialog.Builder within the onClick method.**
- **Set the title, message, and positive/negative buttons.**

Handle Dialog Button Clicks
- **In .setPositiveButton(), define the action for confirmation (e.g., retry or clear fields).**
- **In .setNegativeButton(), define the action for cancellation (e.g., dismiss the dialog).**

Show the Dialog
- **Call .show() on the AlertDialog.Builder instance to display the dialog.**
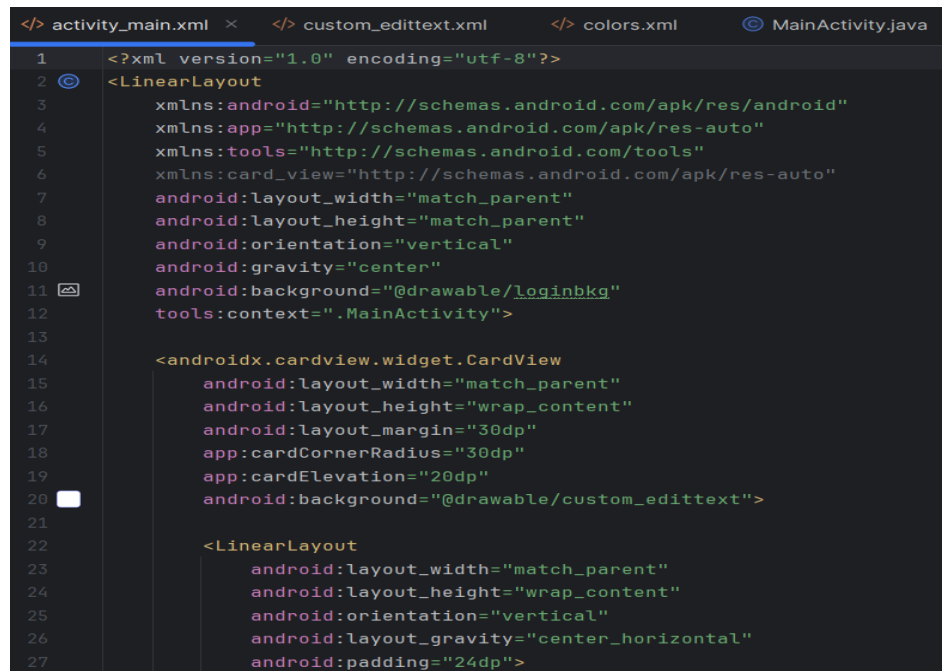
Test the Application

Testing Phase: Compilation of Code (error detection)

There is no error.

Implementation Phase: Final Output(no error)

Step 1: Design the Layout

- Open `activity_main.xml` from the **res → layout** folder.
- Drag a **Button** from the **Palette** to the design view.

```xml
</> activity_main.xml  ×    </> custom_edittext.xml    </> colors.xml    © MainActivity.java
1      <?xml version="1.0" encoding="utf-8"?>
2 ©    <LinearLayout
3          xmlns:android="http://schemas.android.com/apk/res/android"
4          xmlns:app="http://schemas.android.com/apk/res-auto"
5          xmlns:tools="http://schemas.android.com/tools"
6          xmlns:card_view="http://schemas.android.com/apk/res-auto"
7          android:layout_width="match_parent"
8          android:layout_height="match_parent"
9          android:orientation="vertical"
10         android:gravity="center"
11         android:background="@drawable/loginbkg"
12         tools:context=".MainActivity">
13
14         <androidx.cardview.widget.CardView
15             android:layout_width="match_parent"
16             android:layout_height="wrap_content"
17             android:layout_margin="30dp"
18             app:cardCornerRadius="30dp"
19             app:cardElevation="20dp"
20             android:background="@drawable/custom_edittext">
21
22             <LinearLayout
23                 android:layout_width="match_parent"
24                 android:layout_height="wrap_content"
25                 android:orientation="vertical"
26                 android:layout_gravity="center_horizontal"
27                 android:padding="24dp">
```
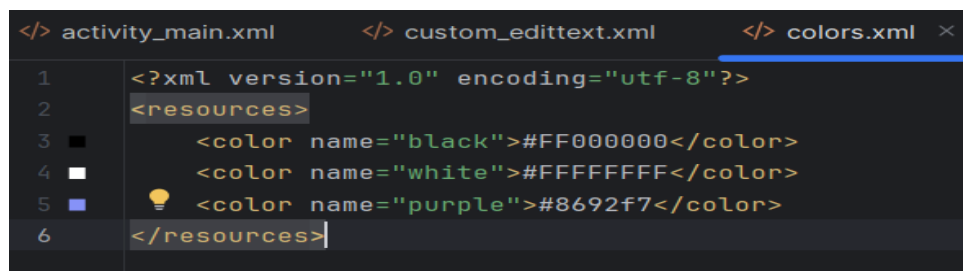
Configuring colors.xml (For Custom Theme/Style)
Location: res → values → colors.xml

```xml
</> activity_main.xml       </> custom_edittext.xml        </> colors.xml  ×
1      <?xml version="1.0" encoding="utf-8"?>
2      <resources>
3          <color name="black">#FF000000</color>
4          <color name="white">#FFFFFFFF</color>
5          <color name="purple">#8692f7</color>
6      </resources>
```
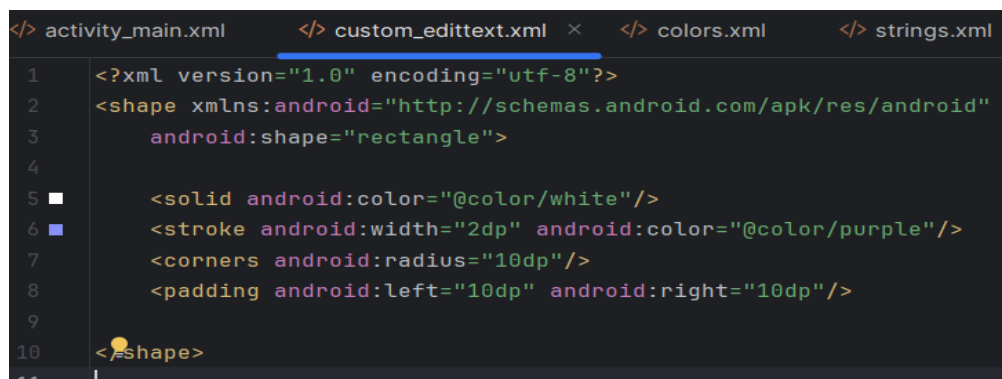
Open customedit.xml from res → values.

```xml
</> activity_main.xml       </> custom_edittext.xml  ×   </> colors.xml       </> strings.xml
1      <?xml version="1.0" encoding="utf-8"?>
2      <shape xmlns:android="http://schemas.android.com/apk/res/android"
3          android:shape="rectangle">
4
5          <solid android:color="@color/white"/>
6          <stroke android:width="2dp" android:color="@color/purple"/>
7          <corners android:radius="10dp"/>
8          <padding android:left="10dp" android:right="10dp"/>
9
10     </shape>
11
```
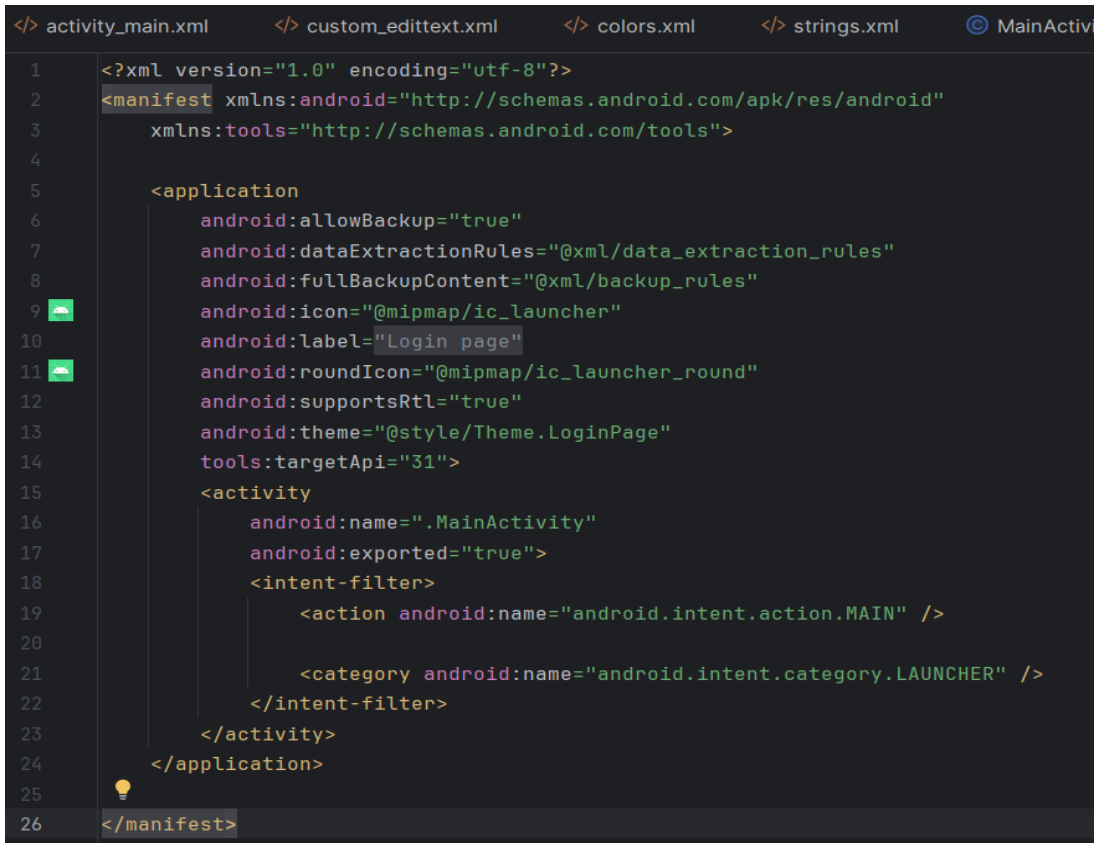
Implementation Phase: Final Output(no error)
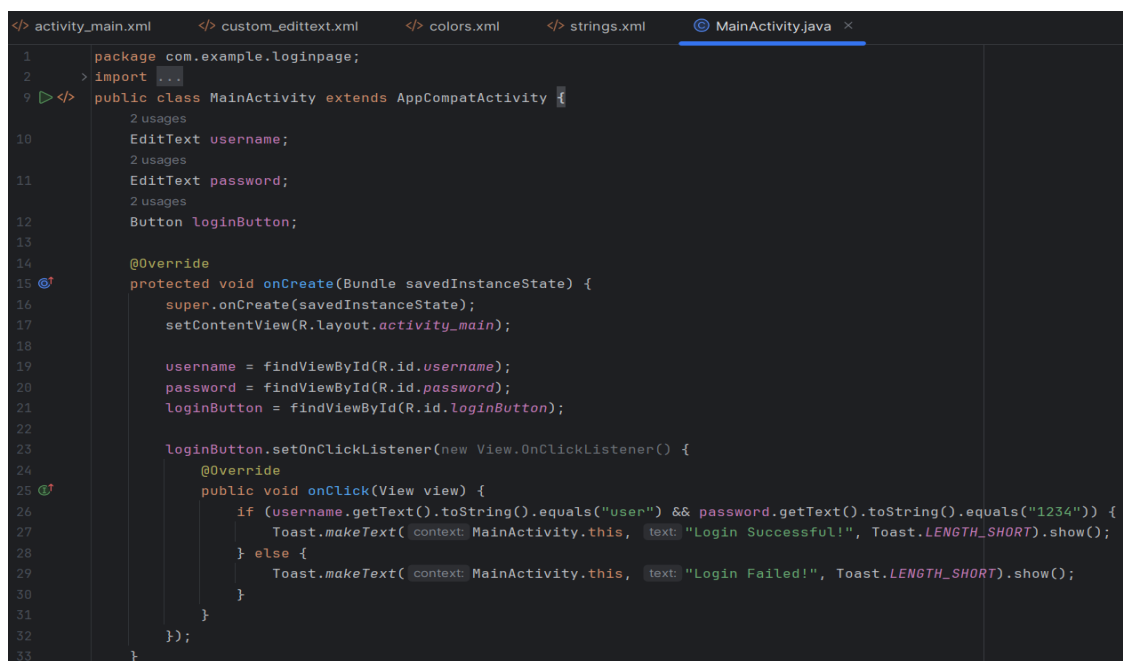
Open AndroidManifest.xml in the app → src → main folder.

```xml
</> activity_main.xml    </> custom_edittext.xml    </> colors.xml    </> strings.xml    © MainActiv

1    <?xml version="1.0" encoding="utf-8"?>
2    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3        xmlns:tools="http://schemas.android.com/tools">
4
5        <application
6            android:allowBackup="true"
7            android:dataExtractionRules="@xml/data_extraction_rules"
8            android:fullBackupContent="@xml/backup_rules"
9            android:icon="@mipmap/ic_launcher"
10           android:label="Login page"
11           android:roundIcon="@mipmap/ic_launcher_round"
12           android:supportsRtl="true"
13           android:theme="@style/Theme.LoginPage"
14           tools:targetApi="31">
15           <activity
16               android:name=".MainActivity"
17               android:exported="true">
18               <intent-filter>
19                   <action android:name="android.intent.action.MAIN" />
20
21                   <category android:name="android.intent.category.LAUNCHER" />
22               </intent-filter>
23           </activity>
24       </application>
25
26   </manifest>
```
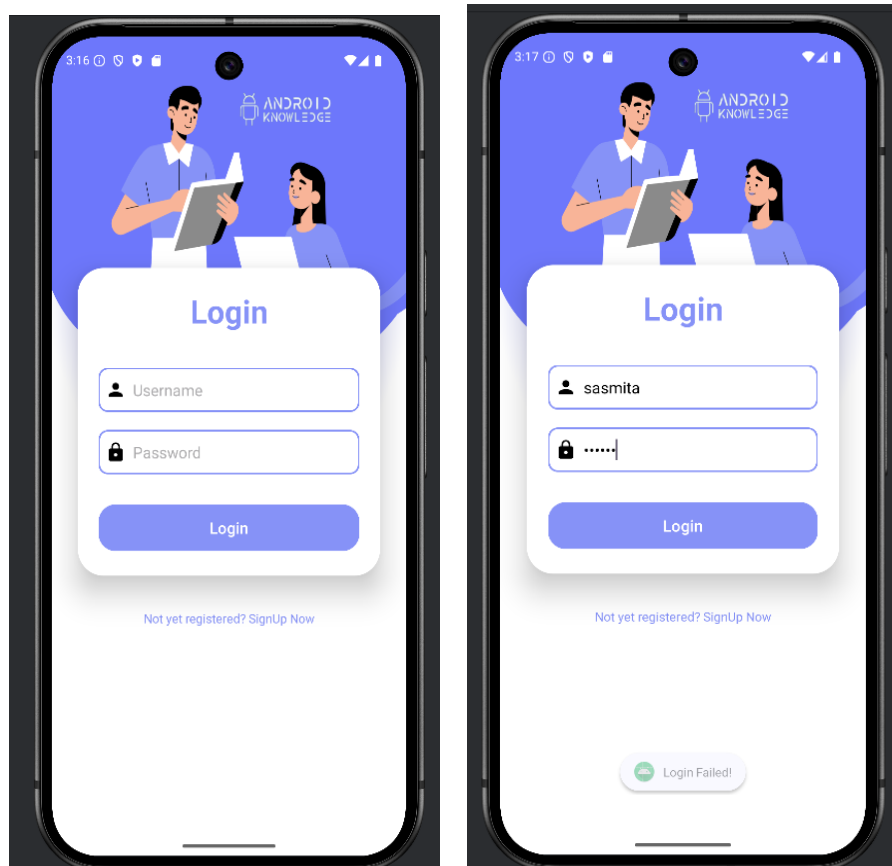
Configuring Mainactivity.java

```java
</> activity_main.xml    </> custom_edittext.xml    </> colors.xml    </> strings.xml    © MainActivity.java  ×

1     package com.example.loginpage;
2     import ...
9     public class MainActivity extends AppCompatActivity {
          2 usages
10        EditText username;
          2 usages
11        EditText password;
          2 usages
12        Button loginButton;
13
14        @Override
15        protected void onCreate(Bundle savedInstanceState) {
16            super.onCreate(savedInstanceState);
17            setContentView(R.layout.activity_main);
18
19            username = findViewById(R.id.username);
20            password = findViewById(R.id.password);
21            loginButton = findViewById(R.id.loginButton);
22
23            loginButton.setOnClickListener(new View.OnClickListener() {
24                @Override
25                public void onClick(View view) {
26                    if (username.getText().toString().equals("user") && password.getText().toString().equals("1234")) {
27                        Toast.makeText( context: MainActivity.this, text: "Login Successful!", Toast.LENGTH_SHORT).show();
28                    } else {
29                        Toast.makeText( context: MainActivity.this, text: "Login Failed!", Toast.LENGTH_SHORT).show();
30                    }
31                }
32            });
33        }
```

Click **Run**.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :