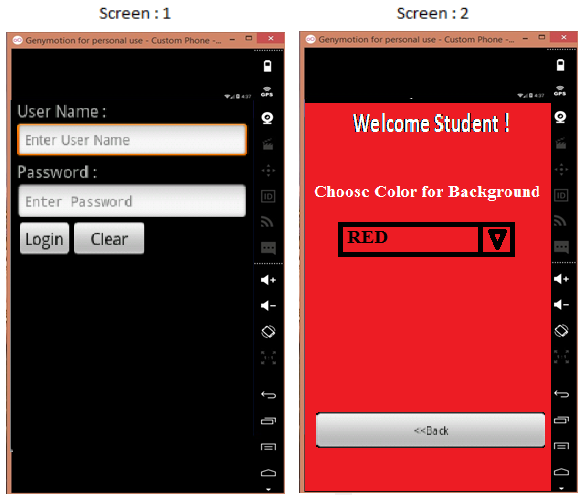
PRACTICAL 5

**AIM:**Activity and Intent, Dynamically design a UI screen with layout, properties and view components through java code.

**Scenario:**Create an application depicting a login module. Ask the user for username and password in the first screen and if successful login (correct username and password combination) go to the next screen else on failing login, alert user using Toast. Pass the successful username and password to the next screen to be displayed with a Welcome message and a back button. The second screen should be created dynamically, don’t design a layout xml file for the second screen.

****

**Part B (to be completed by students)**

**(Students must submit the soft copy as per the following segments. The soft copy must be uploaded on the Blackboard. The filename should be Batch\_RollNo\_Exp\_No)**

|  |  |
| --- | --- |
| **Roll No.: K005** | **Name: Jal Bafana** |
| **Prog/Yr/Sem:Btech. Cyber Sec (Sem-4)** | **Batch: K1** |
| **Date of Experiment: 05.03.2025** | **Date of Submission:05.03.2025** |

1. **Program Scenario and Program code:** (Write Scenario and Paste your program code (Java, xml resource and layout)).

**activity\_mail.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:orientation="vertical"  
android:padding="50dp"  
android:gravity="center">  
  
<!-- Username EditText -->  
<EditText  
android:id="@+id/usernameEditText"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:hint="Enter Username"  
android:inputType="text" />  
  
<!-- Password EditText -->  
<EditText  
android:id="@+id/passwordEditText"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:hint="Enter Password"  
android:inputType="textPassword" />  
  
<!-- Login Button -->  
<Button  
android:id="@+id/loginButton"  
android:layout\_width="match\_parent"  
android:layout\_height="wrap\_content"  
android:text="Login" />  
  
</LinearLayout>

**MainActivity.java**

package com.example.k005\_lab6;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivityextends AppCompatActivity {  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
setContentView(R.layout.*activity\_main*);   
  
EditTextusernameEditText = findViewById(R.id.*usernameEditText*);  
EditTextpasswordEditText = findViewById(R.id.*passwordEditText*);  
Button loginButton = findViewById(R.id.*loginButton*);  
  
loginButton.setOnClickListener(v -> {  
 String username = usernameEditText.getText().toString();  
String password = passwordEditText.getText().toString();  
  
 if ("user".equals(username) &&"password".equals(password)) {  
Intent intent = new Intent(MainActivity.this, SecondActivity.class);  
intent.putExtra("username", username);  
startActivity(intent);  
} else {  
Toast.*makeText*(MainActivity.this, "Invalid login credentials", Toast.*LENGTH\_SHORT*).show();  
}  
 });  
}  
}

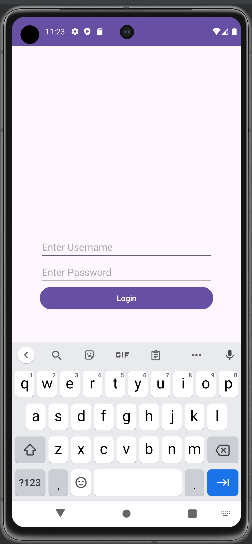
**SecondActivity.java**

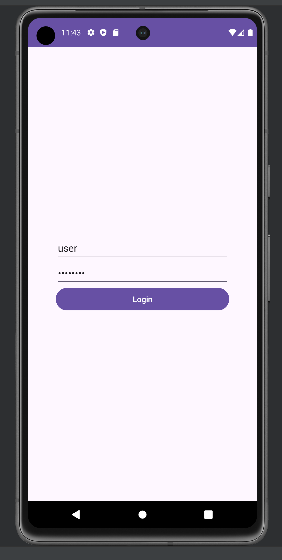
package com.example.k005\_lab6;  
  
import android.os.Bundle;  
import android.view.Gravity;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.LinearLayout;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SecondActivityextends AppCompatActivity {  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
super.onCreate(savedInstanceState);  
  
LinearLayout layout = new LinearLayout(this);  
layout.setOrientation(LinearLayout.*VERTICAL*);  
layout.setPadding(50, 50, 50, 50);  
layout.setGravity(Gravity.*CENTER\_HORIZONTAL*);   
  
TextViewwelcomeMessageTextView = new TextView(this);  
welcomeMessageTextView.setText("Welcome Student !!!");  
welcomeMessageTextView.setTextSize(30);  
welcomeMessageTextView.setLayoutParams(new LinearLayout.LayoutParams(  
LinearLayout.LayoutParams.*MATCH\_PARENT*, LinearLayout.LayoutParams.*WRAP\_CONTENT*));  
  
Button backButton = new Button(this);  
backButton.setText("Back");  
  
LinearLayout.LayoutParamsbuttonLayoutParams = new LinearLayout.LayoutParams(  
LinearLayout.LayoutParams.*WRAP\_CONTENT*, LinearLayout.LayoutParams.*WRAP\_CONTENT*);  
buttonLayoutParams.gravity= Gravity.*CENTER*; // Center the button horizontally  
buttonLayoutParams.topMargin= 200;  
backButton.setLayoutParams(buttonLayoutParams);  
  
layout.addView(welcomeMessageTextView);  
layout.addView(backButton);  
  
setContentView(layout);  
   
backButton.setOnClickListener(v -> {  
 finish();  
});  
}  
}

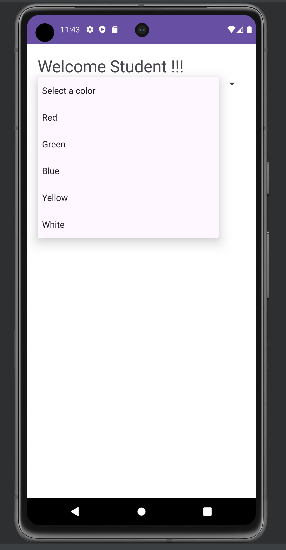
**Manifest.xml**

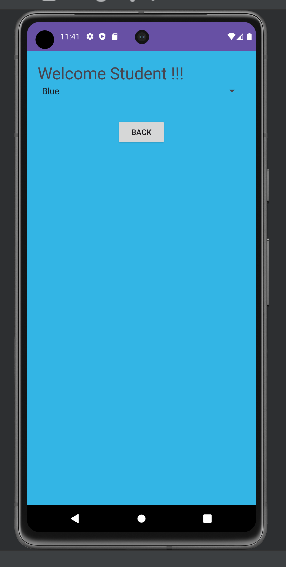
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools">  
  
<application  
android:allowBackup="true"  
android:dataExtractionRules="@xml/data\_extraction\_rules"  
android:fullBackupContent="@xml/backup\_rules"  
android:icon="@mipmap/ic\_launcher"  
android:label="@string/app\_name"  
android:roundIcon="@mipmap/ic\_launcher\_round"  
android:supportsRtl="true"  
android:theme="@style/Theme.K005\_Lab6"  
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>  
<activity android:name=".SecondActivity"></activity>  
  
</application>  
  
</manifest>

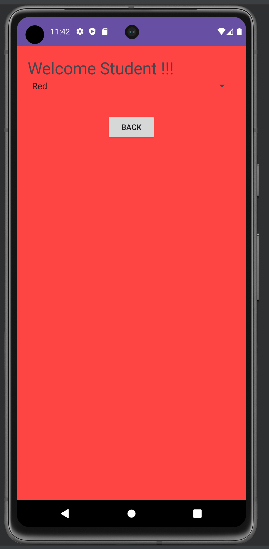
1. **Output:** (Paste your program input and output screen shots).

****

****

****

****

****

1. **Observations:** A brief description of the design aspects and working of the code in your own words.

The second screen (SecondActivity) is created dynamically using Java, without XML. A LinearLayout is used to arrange elements vertically, with a TextView displaying a welcome message, a Spinner for color selection, and a Button to navigate back to the main screen. The background color of the screen changes based on the spinner selection. The button is positioned at the bottom and centered horizontally.

1. **Questions:** Draw & Explain with respect to layouts the Scene Graph of the experiment.

**LinearLayout (Parent)**

**├──TextView (Welcome Message)**

**├── Spinner (Color Selector)**

**└── Button (Back Button)**

**Explanation:**

* **LinearLayout** is the parent container with vertical orientation.
* **TextView** displays a welcome message at the top.
* **Spinner** allows the user to select a color, changing the background color.
* **Button** is placed at the bottom, providing functionality to return to the main screen.

1. **Conclusion (Learning Outcomes):** How were the outcomes defined for the experiment in Part A fulfilled through the scenarios?

The experiment shows how to create dynamic UIs with Java, handle user input, and change UI elements like background color. It helps in understanding dynamic layouts and event handling in Android.