Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 1A**	
Title of Lab Assignment: Study about Android platform, the layers of android, Four kinds of android components, understanding the android Manifest.xml file, Android Activity Life cycle.		
DOP: 18-08-2024	**DOS: 24-08-2024**	
CO Mapped: **CO1**	**PO Mapped:** **PO3, PO5, PO7, PO12, PSO1, PSO2**	**Signature:**

Practical 1A

Aim: Study about Android platform, the layers of android, Four kinds of android components, understanding the android Manifest.xml file, Android Activity Life cycle.

Theory:

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android programming is based on Java programming language so if you have basic understanding of Java programming then it will be easier to learn Android application development.

Layers: Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

XMPP Service-Extensible Messaging and Presence Protocol is an open communication protocol designed for instant messaging, presence information, and contact list maintenance.

Surface manager responsible for managing access to the display subsystem. SGL(Scalable Graphics Library) and OpenGL ES both cross-language, cross-platform application program interface (API) are used for 2D and 3D computer graphics. SQLite provides database support. FreeType provides font support.

Web-Kit This open source web browser engine provides all the functionality to display web content and to simplify page loading.

SSL (Secure Sockets Layer) is security technology to establish an encrypted link between a web server and a web browser.

Like Java Virtual Machine (JVM), Dalvik Virtual Machine (DVM) is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently.

Four kinds of android components

Activities: An activity represents a single screen with a user interface,in-short Activity performs action on the screen. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

An activity is implemented as a subclass of Activity class as follows:

```
public class MainActivity extends Activity { }
```

Services: A service is a component that runs in the background to perform long-running operations. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity.

A service is implemented as a subclass of Service class as follows:

```
public class MyService extends Service { }
```

Broadcast: Receivers Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is the broadcast receiver who will intercept this communication and will initiate appropriate action. A broadcast receiver is implemented as a subclass of BroadcastReceiverclass and each message is broadcaster as an Intent object.

```
public class MyReceiver extends BroadcastReceiver {
        public void onReceive(context, intent){ }
}
```

Content Providers: A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolverclass. The data may be stored in the file system, the database or somewhere else entirely. A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyContentProvider extends ContentProvider {
        public void onCreate(){ }
}
```

Sr. No.	Components
1	**Activities**: They dictate the UI and handle the user interaction to the smartphone screen.
2	**Services**: They handle background processing associated with an application.
3	**Broadcast Receivers**: They handle communication between Android OS and applications.
4	**Content Providers**: They handle data and database management issues.

Understanding the android Manifest.xml file

Whatever component you develop as a part of your application, you must declare all its components in a manifest.xml which resides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS. For example, a default manifest file will look like as following file −

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.androidapp.myapplication">
<application android:allowBackup="true"
android:icon="@mipmap/ic_launcher" android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter> <action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter> </activity> </application> </manifest>
```

Here <application>...</application> tags enclosed the components related to the application. Attribute android:icon will point to the application icon available under res/drawable-hdpi. The application uses the image named ic_launcher.png located in the drawable folders.

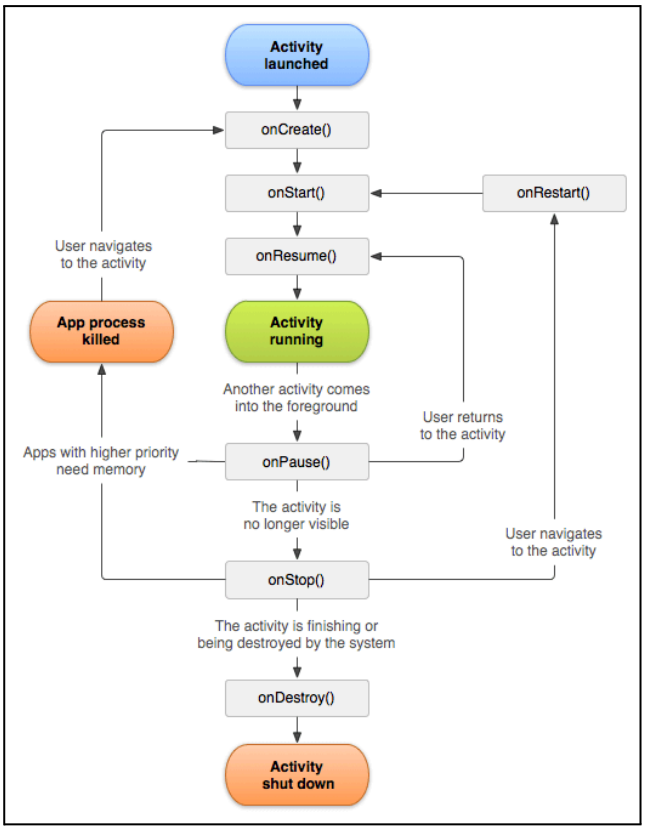
The <activity> tag is used to specify an activity and android:name attribute specifies the fully qualified class name of the Activity subclass and the android:label attributes specifies a string to use as the label for the activity. You can specify multiple activities using <activity> tags.

The action for the intent filter is named android.intent.action.MAIN to indicate that this activity serves as the entry point for the application. The category for the intent-filter is named android.intent.category.LAUNCHER to indicate that the application can be launched from the device's launcher icon.

The @string refers to the strings.xml file explained below. Hence, @string/app_name refers to the app_name string defined in the strings.xml file, which is "HelloWorld". Similarly, other strings get populated in the application.e that the application can be launched from the device's launcher icon.

Android Activity Life cycle.

An activity represents a single screen with a user interface just like a window or frame of Java. Android activity is the subclass of ContextThemeWrapper class.



The Activity class defines the following call backs i.e. events. You don't need to implement all the callbacks methods.

Sr. No.	Components and Description
1	**onCreate()**: This is the first callback and called when the activity is first created.
2	**onStart()**: This callback is called when the activity becomes visible to the user.
3	**onResume()**: This is called when the user starts interacting with the application.
4	**onPause()**: The paused activity does not receive user input and cannot execute any code and is called when the current activity is being paused and the previous activity is being resumed.
5	**onStop()**: This callback is called when the activity is no longer visible.
6	**onDestroy()**: This callback is called before the activity is destroyed by the system.
7	**onRestart()**: This callback is called when the activity restarts after stopping it.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 1B**	
Title of Lab Assignment: Android program using various UI Components.(EditText, RatingBar, CheckBox, RadioButton, Button etc). Get the values from the components and display them on the next activity.		
DOP: 18-08-2024	**DOS: 24-08-2024**	
CO Mapped: **CO1**	**PO Mapped:** **PO3, PO5, PO7, PO12, PSO1, PSO2**	**Signature:**

Practical 1B

Aim: Android program using various UI components and different layouts and views. Android program using various UI Components. (EditText, RatingBar, CheckBox, RadioButton, Button etc). Get the values from the components and display them on the next activity.

Code:

MainActivity.java

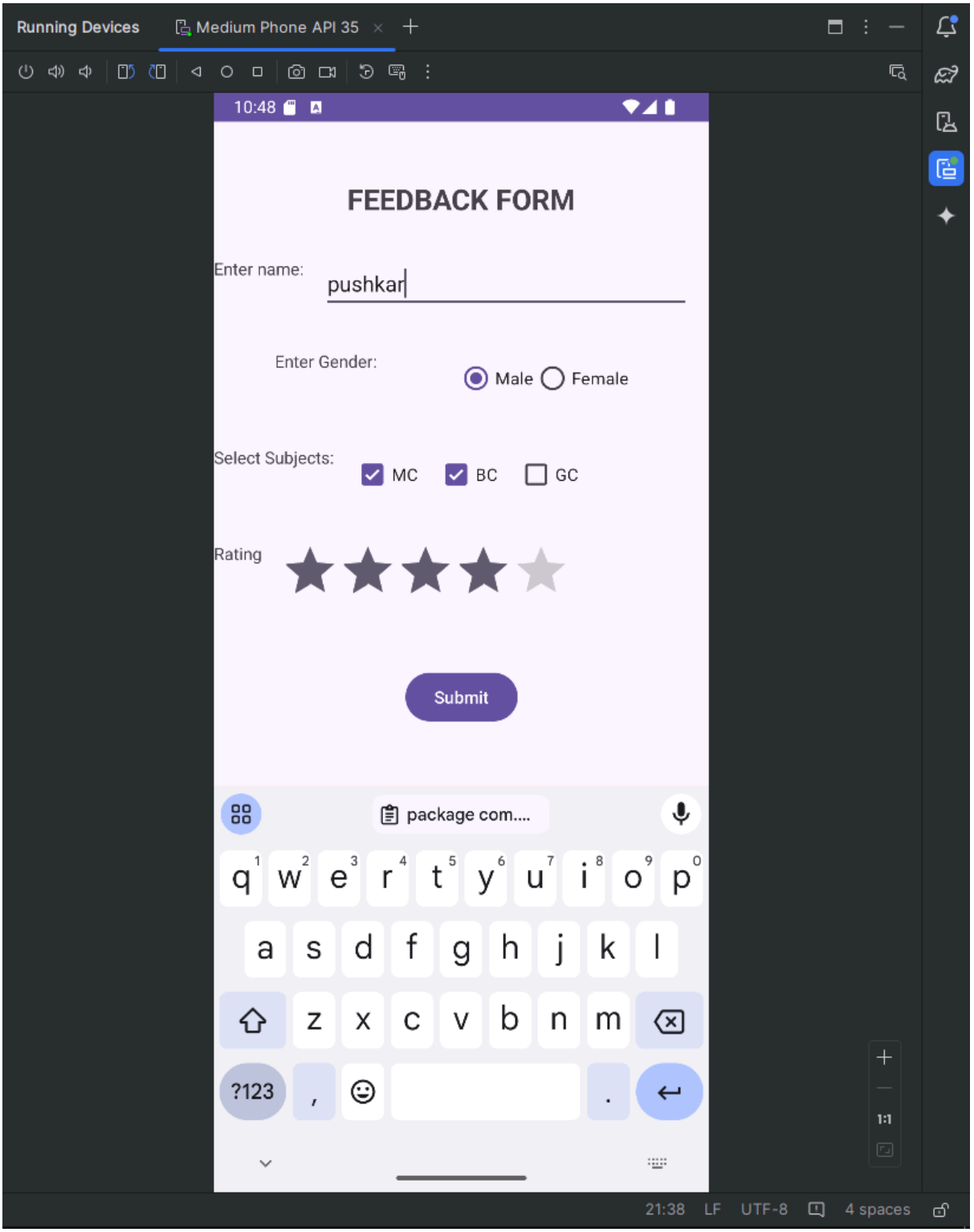
```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RatingBar;

public class MainActivity extends AppCompatActivity {
    EditText editTextName;
    RadioButton radioMale, radioFemale;
    CheckBox checkBoxMC, checkBoxBC, checkBoxGC;
    RatingBar ratingBar;
    Button submitButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize the components
        editTextName = findViewById(R.id.editTextText3);
        radioMale = findViewById(R.id.radioButton3);
```

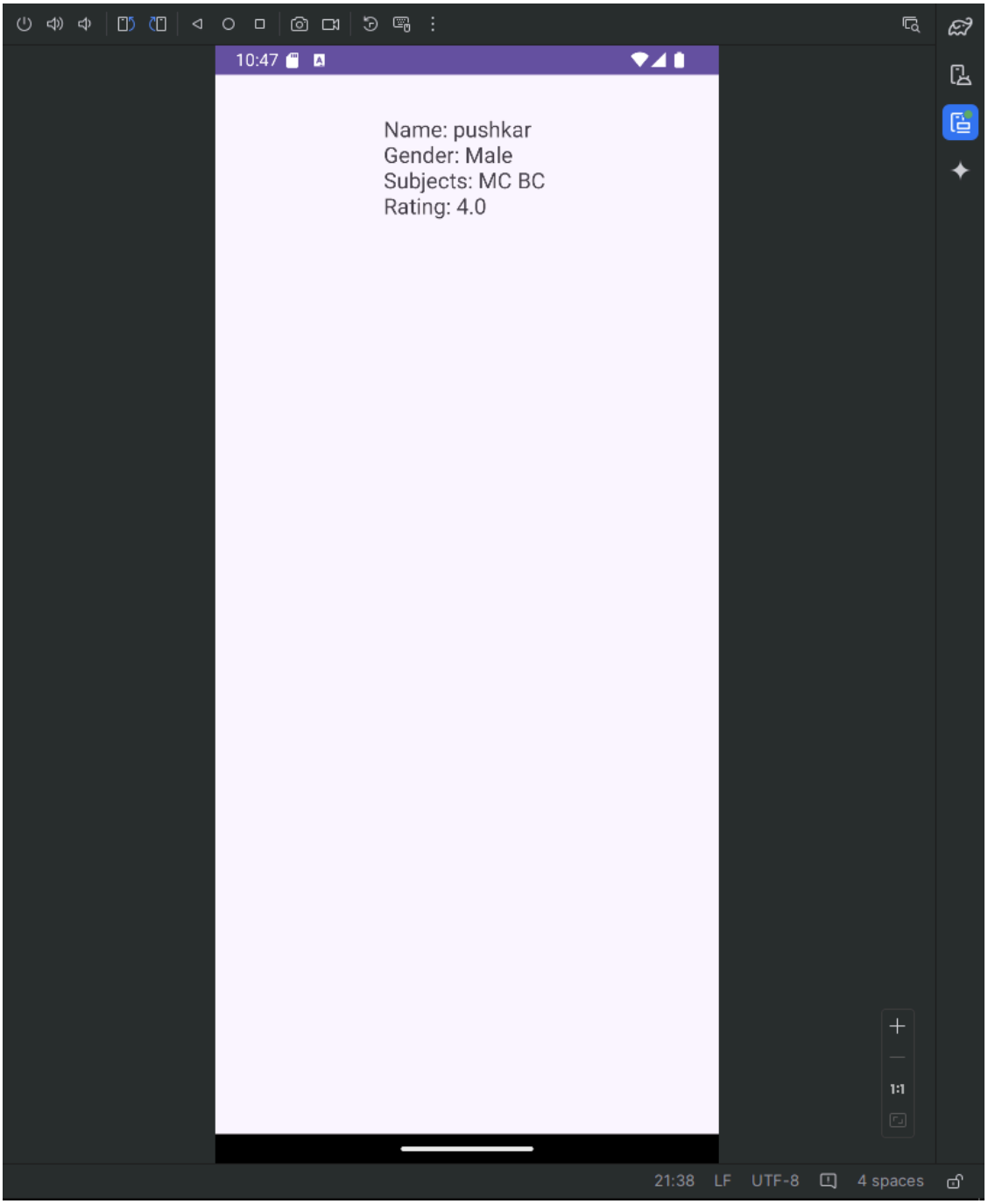
```
        radioFemale = findViewById(R.id.radioButton4);
        checkBoxMC = findViewById(R.id.checkBox);
        checkBoxBC = findViewById(R.id.checkBox2);
        checkBoxGC = findViewById(R.id.checkBox3);
        ratingBar = findViewById(R.id.ratingBar);
        submitButton = findViewById(R.id.button);

        // Set onClickListener for the button
        submitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get the values from the components
                String name = editTextName.getText().toString();
                String gender = radioMale.isChecked() ? "Male" : "Female";
                String subjects = "";
                if (checkBoxMC.isChecked()) subjects += "MC ";
                if (checkBoxBC.isChecked()) subjects += "BC ";
                if (checkBoxGC.isChecked()) subjects += "GC ";
                float rating = ratingBar.getRating();

                // Create an intent to go to the next activity
                Intent intent = new Intent(MainActivity.this, NextActivity.class);
                // Pass the collected data to the next activity
                intent.putExtra("name", name);
                intent.putExtra("gender", gender);
                intent.putExtra("subjects", subjects);
                intent.putExtra("rating", rating);
                startActivity(intent);
            }
        });
    }
}
```

NextActivity.java

```
package com.example.myapplication;

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

public class NextActivity extends AppCompatActivity {

    TextView textViewSummary;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_next);

        textViewSummary = findViewById(R.id.textViewSummary);

        // Retrieve the data passed from MainActivity
        String name = getIntent().getStringExtra("name");
        String gender = getIntent().getStringExtra("gender");
        String subjects = getIntent().getStringExtra("subjects");
        float rating = getIntent().getFloatExtra("rating", 0);

        // Display the retrieved data
        String summary = "Name: " + name + "\n" +
                "Gender: " + gender + "\n" +
                "Subjects: " + subjects + "\n" +
                "Rating: " + rating;

        textViewSummary.setText(summary);
    }
}
```

Output:


Running Devices
Medium Phone API 35
10:48
FEEDBACK FORM
Enter name:
pushkar
Enter Gender:
Male
Female
Select Subjects:
MC
BC
GC
Rating
Submit
package com....
21:38
LF
UTF-8
4 spaces



Conclusion:

Implementing UI in Android using Android Studio involves designing layouts with XML and coding interactions in Java. The process includes creating activities and defining their UI elements, such as buttons, text fields, and checkboxes. To enable navigation between activities, you use Intent objects to start new activities and pass data between them. This seamless transition allows for dynamic user experiences, where data collected from user inputs in one activity can be efficiently transferred and utilized in another, enhancing the overall functionality and interactivity of the app.

Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 1**	
Title of Lab Assignment: Create a login form. For a successful login, display the welcome page, and in case of failure display an alert box indicating an error message and attempts made. Disable the submit button after 3 wrong attempts and display the alert message indicating the same.		
DOP: 18-08-2024	**DOS: 24-08-2024**	
CO Mapped: **CO1**	**PO Mapped:** **PO3, PO5, PO7, PO12, PSO1, PSO2**	**Signature:**

Practical 1C

Aim: Create a login form. For a successful login, display the welcome page, and in case of failure display an alert box indicating an error message and attempts made. Disable the submit button after 3 wrong attempts and display the alert message indicating the same.

Code:

MainActivity.java

```
package com.example.practical1c;

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 MainActivity extends AppCompatActivity {
    private EditText usernameEditText;
    private EditText passwordEditText;
    private Button loginButton;
    private int attemptCount = 0;
    private static final int MAX_ATTEMPTS = 3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        usernameEditText = findViewById(R.id.username);
        passwordEditText = findViewById(R.id.password);
        loginButton = findViewById(R.id.login_button);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
```

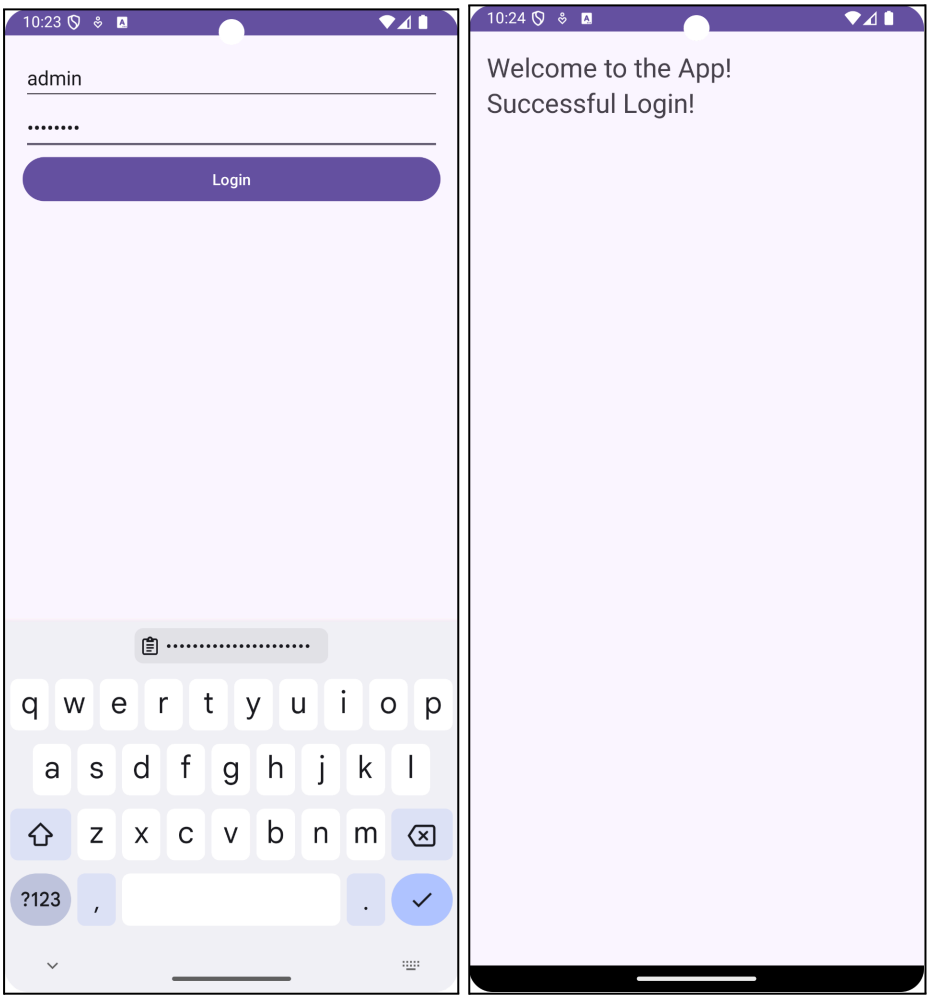
```
        public void onClick(View v) {
            handleLogin();
        }
    });
}

private void handleLogin() {
    String username = usernameEditText.getText().toString();
    String password = passwordEditText.getText().toString();

    if (username.equals("admin") && password.equals("password")) {
        // Successful login
        Intent intent = new Intent(MainActivity.this, WelcomeActivity.class);
        startActivity(intent);
        finish();
    } else {
        // Failed login
        attemptCount++;
        if (attemptCount >= MAX_ATTEMPTS) {
            loginButton.setEnabled(false);
                Toast.makeText(MainActivity.this, "Too many attempts. Please try again later.",
Toast.LENGTH_LONG).show();
        } else {
                        Toast.makeText(MainActivity.this, "Login failed. Attempts left: " +
(MAX_ATTEMPTS - attemptCount), Toast.LENGTH_SHORT).show();
        }
    }
}
```

WelcomeActivity.java

```
package com.example.practical1c;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class WelcomeActivity extends AppCompatActivity {
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
    }
}
```

Output:



Conclusion:

In conclusion, creating a login form in Android Studio involves designing an interface that handles user authentication by displaying a welcome page upon successful login. For failed login attempts, an alert box shows an error message and the number of attempts made. To enhance security, the submit button is disabled after three incorrect attempts, accompanied by an alert message informing the user of the same. This approach ensures a user-friendly and secure login process.