

Srikanth Gude – sgcnm@umsystem.edu

Git Hub Link - <https://github.com/gudesrikanth/webcourse/tree/main/Webpart/ICP8>

Gopi Nelluri – gng75@umsystem.edu

Git Hub Link -

https://github.com/gopinelluri9/demo_remote_repository/tree/main/WebPart/ICP8

ICP – 8(Login and Logout in Android App)

Introduction:

We used following technologies to complete the task - Java, Android Studio, XML

Android Studio: Android Studio is an open-source development platform that allows us to create applications for Android-based mobile devices, tablets, and televisions. In our Task, we will be designing and developing a mobile application with the help of Android Studio.

Java: We have chosen Java as the programming language for the development of the mobile application. Whenever a user performs an action, the functionality of the action is written in Java code.

XML: For the layout of the app, we used the XML format, which is generally the default format used by Android studios for the design layout of apps.

Task

In this Task, we need to create a Login and Logout page in Android APP.

- The Login page should contain two input text areas for the user's credentials: Username and Password. The user should enter their credentials into these fields.
- When a user successfully submits their username and password, they should be redirected to a welcome page, where they should have access to the logout option.
- When the user enters an invalid username or password, app should not redirect the user to the welcome page and display's respected error message.
- When the user clicks on the logout button, the app should allow the user to logout and display the login page.

Procedure:

Create an empty Empty activity for the app and name it with the project name. After creating an empty blank activity and naming our project LoginApp, we selected Java as the programming language as previously mentioned and clicked on the finish button, which results in the creation of a Login app.

Login:

activity main.xml has been modified to include two new Text fields and a button for the Login process.

First Text Field: Username

We've included a Text Field with the username, input type @+id/username is the id of the text field, and the hint is username, which is shown with faded text to let people know that they should fill in the text box with their username. Also created vector asset using drawable resource for username field to display user icon beside the username field.

```
24 <EditText
25     android:id="@+id/username"
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:layout_below="@id/signin"
29     android:layout_marginStart="10dp"
30     android:layout_marginTop="6dp"
31     android:layout_marginEnd="10dp"
32     android:layout_marginBottom="10dp"
33     android:background="#30ffffff"
34     android:drawableLeft="@drawable/ic_baseline_person_outline_24"
35     android:drawablePadding="20dp"
36
37     android:hint="Username"
38     android:padding="20dp"
39     android:textColor="@color/black"
40     android:textColorHint="@color/black" />
```

Second Text Field: Password.

In the second text field we've added, the password is the second one. It's password, and the hint is password, which is shown with faded text to let people know that they should fill in the text box with their password. The id for the text box is @+id/password. Also created vector asset using drawable resource for password field to display info icon beside the password field.

```
42 <EditText
43     android:id="@+id/password"
44     android:layout_width="match_parent"
45     android:layout_height="wrap_content"
46     android:layout_below="@id/username"
47     android:layout_marginStart="10dp"
48     android:layout_marginTop="-8dp"
49     android:layout_marginEnd="10dp"
50     android:layout_marginBottom="10dp"
51     android:background="#30ffffff"
52     android:drawableLeft="@drawable/ic_baseline_info_24"
53     android:drawablePadding="20dp"
54
55     android:hint="Password"
56     android:padding="20dp"
57     android:textColor="@color/black"
58     android:textColorHint="@color/black"
59 />
```

Login Button:

Similar to this, we have added another element for login button, with the text Login and the onClick event named onClick. We have implemented this function in the MainActivity.java file.

onClick Activity:

Functionality: When a user clicks on the login button, this function validates the login credentials and will check for correct username & password; if the check is successful, the user is directed to the welcome page; otherwise, it will display respected error message.

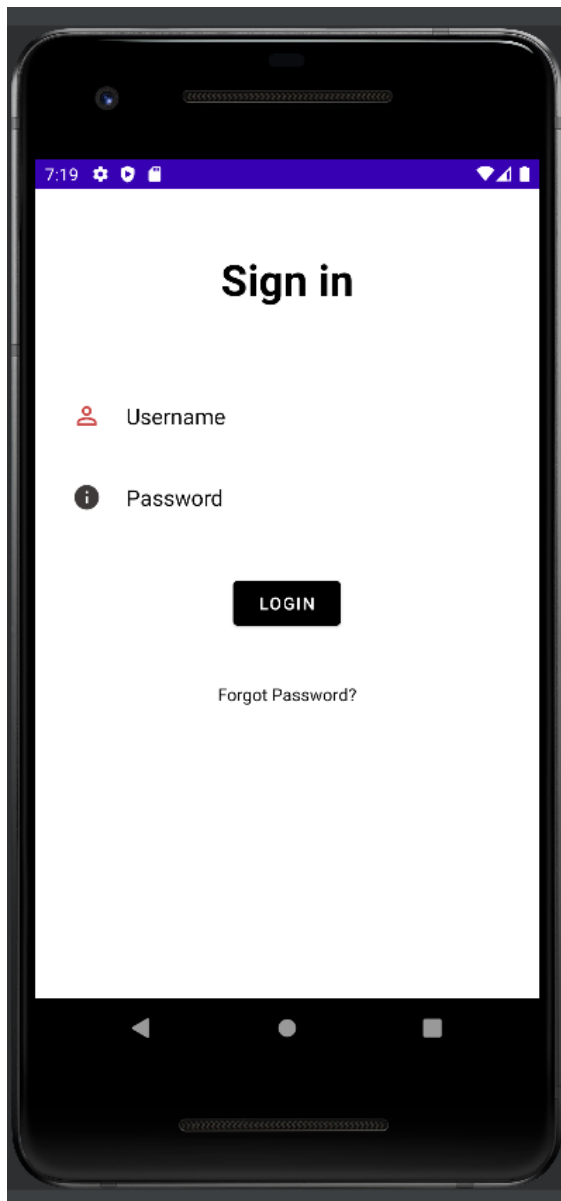
Parameters: This function takes a View parameter, which contains the username and password, as well as other parameters.

We'll use `getText()` to get the username and password. And validates the inputs, if the validation is successful, proceed to `HomeActivity.class`. A new `Intent` is created and passed to the method as an argument. Created multiple `Toast` messages using `if else` function to display appropriate message.

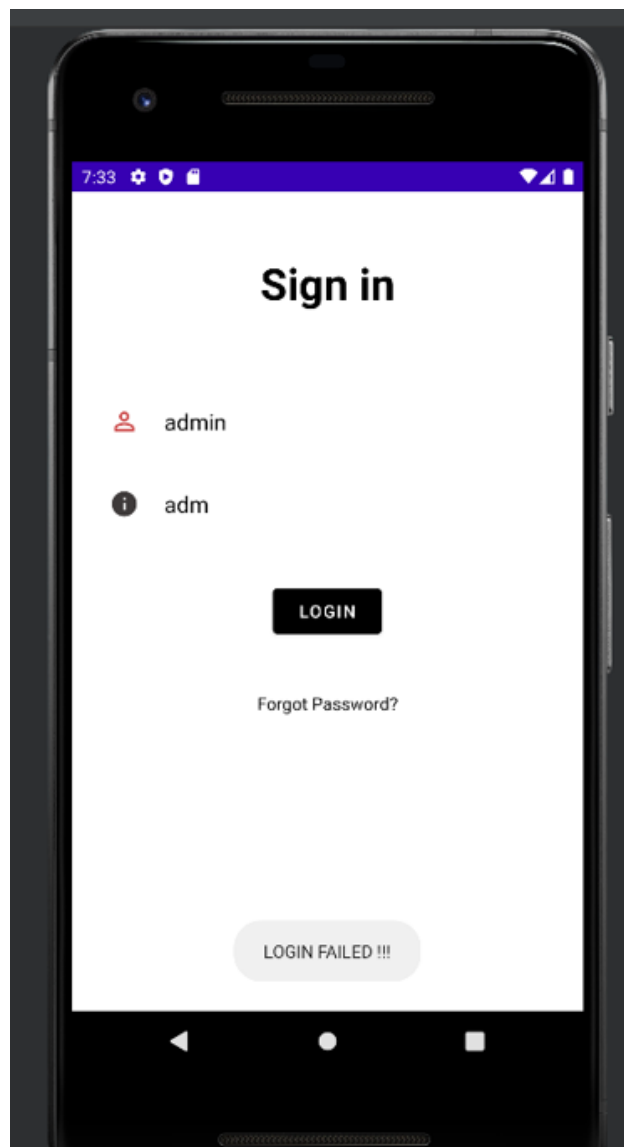
```
34      String user = username.getText().toString();
35      String pass = password.getText().toString();
36
37      if(user.equals("admin") && pass.equals("admin")){
38          //correct
39          Toast.makeText(context: MainActivity.this, text: "LOGIN SUCCESSFUL",Toast.LENGTH_SHORT).show();
40          Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
41          startActivity(intent);
42      }else
43          //incorrect
44          Toast.makeText(context: MainActivity.this, text: "LOGIN FAILED !!!",Toast.LENGTH_SHORT).show();
45      if(TextUtils.isEmpty(user) || TextUtils.isEmpty(pass)){
46          Toast.makeText(context: MainActivity.this, text: "All Fields Required",Toast.LENGTH_SHORT).show();
47      }
48  }
49  });
50
51
52  }
53 }
```

Output:

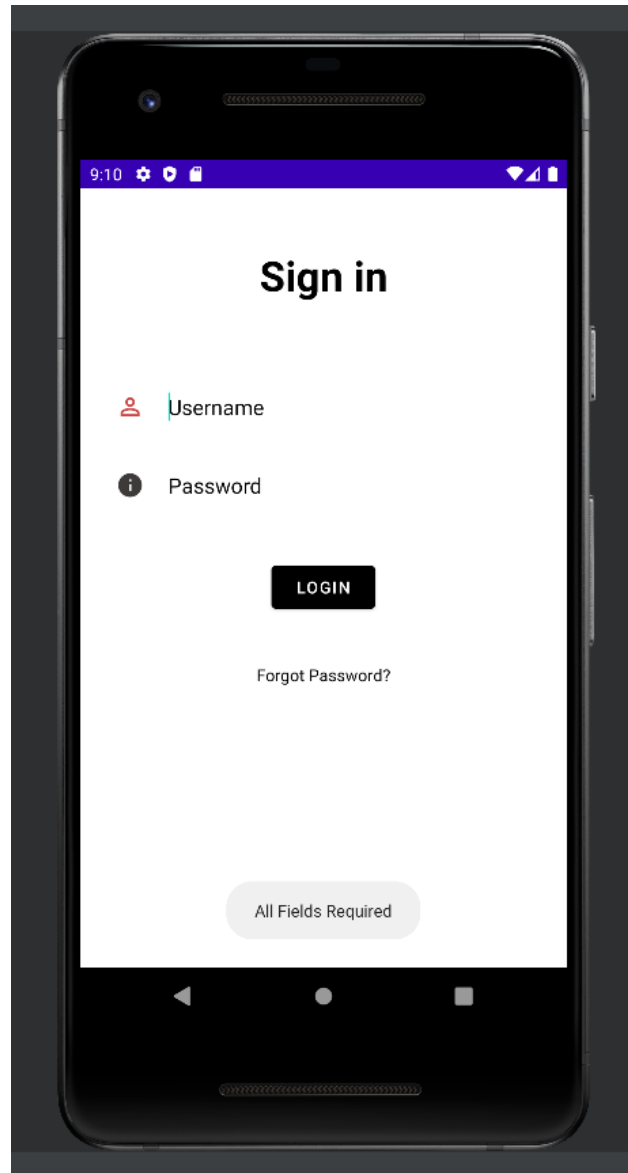
We have run the application on the virtual simulator where it has been deployed and is running. The UI Screen of the app looks like this:



UI LOGIN SCREEN



UNSUCCESSFUL LOGIN



UNSUCCESSFUL LOGIN

In above output screenshots when we enter invalid credentials to the input fields and by leaving the fields as blank it is displaying the error messages in unsuccessful login screens.

activity_home.xml:

After entering the successful login, **activity_home.xml** has a Text field and a Button.

Text Filed: This field has a text Logged In successfully Text message

Logout Button: This button enables the user to get logged out of the screen. The click functionality of the button is **onLogoutClick**.

```
<Button
    android:id="@+id/logoutButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="288dp"
    android:onClick="onLogoutClick"
    android:text="Logout"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    android:background="#673AB7"
    android:textColor="@android:color/white"
/>
```

HomeActivity.java:

When the method is executed then it should navigate the user to the login page which is present in MainActivity.java. This function has Viewed as a parameter that has button id as a value.

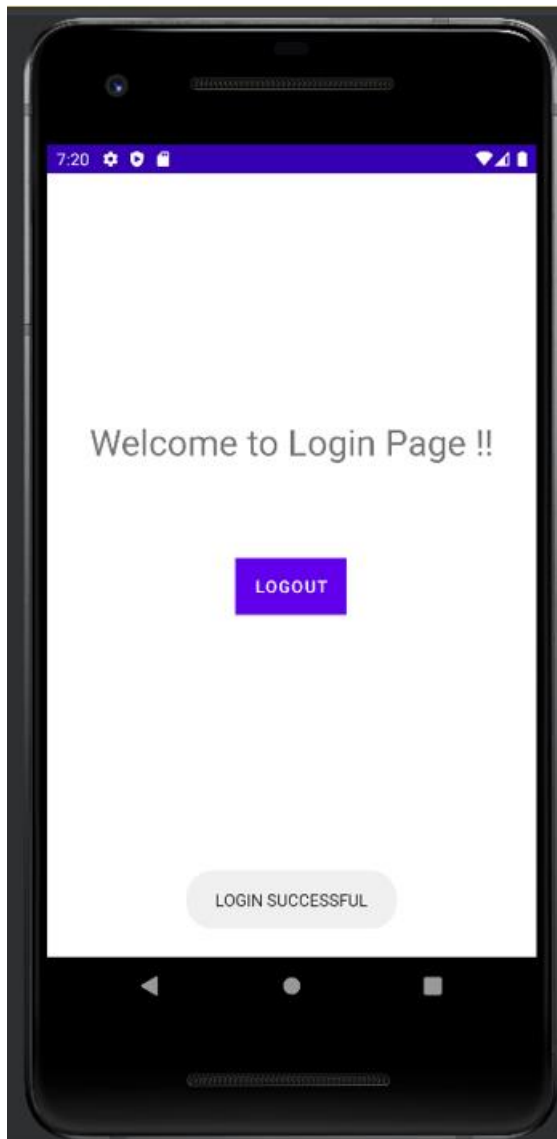
We've created an Intent and navigated to a login page in the MainActivity class.

```
public class HomeActivity extends MainActivity {

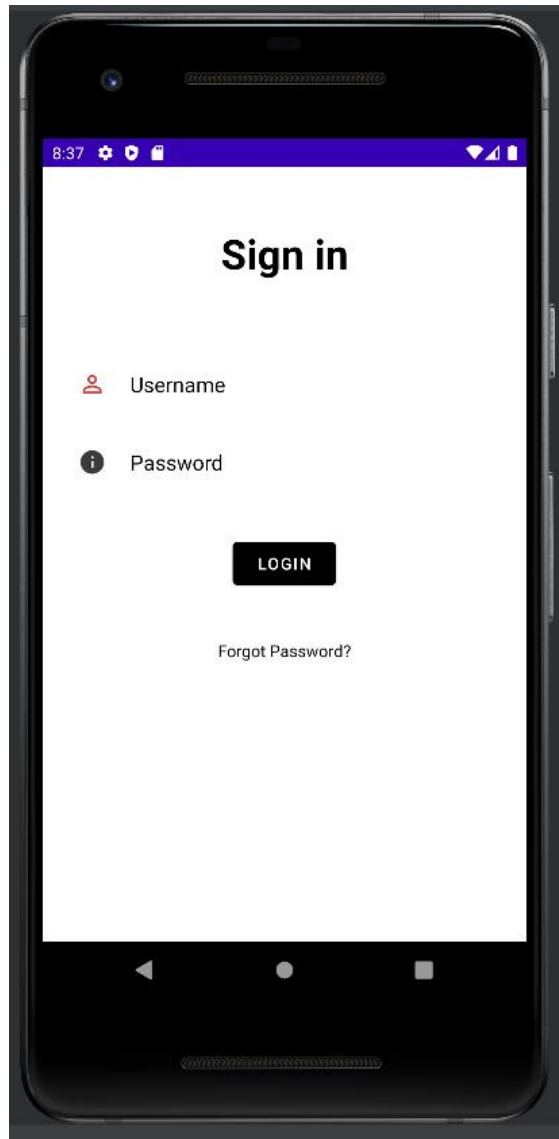
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
    }

    public void onLogoutClick(View v){
        if(v.getId() == R.id.logoutButton)
        {
            Intent i = new Intent( packageContext: HomeActivity.this,MainActivity.class);
            startActivity(i);
        }
    }
}
```


Output:



SUCCESSFUL LOGIN



SIGNIN SCREEN

In above output screenshot when we click on logout button it will redirect to signin screen.

Conclusion:

In this ICP we learned about how to develop a android application by using the Android studio, with java, xml. And at last, we have designed and developed Android Application successfully. We didn't face any major issues while doing this ICP.