**Mobile Application and Development**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lab 3(a): Developing mobile app for login page**

**Task:**

In any app input validation is a very important thing to do. Input Validation eliminates the errors that can be done by user while giving inputs to our app. For example, if we want to get the user’s email we can check the entered email is a valid email or not before storing it inside the database.

**Experiment**:

Develop a Mobile App for login page, to collect User Name, Password, Re-enter Password with options to Display or Hide the Password and Submit button. Show the data stored on the screen by fetching it.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

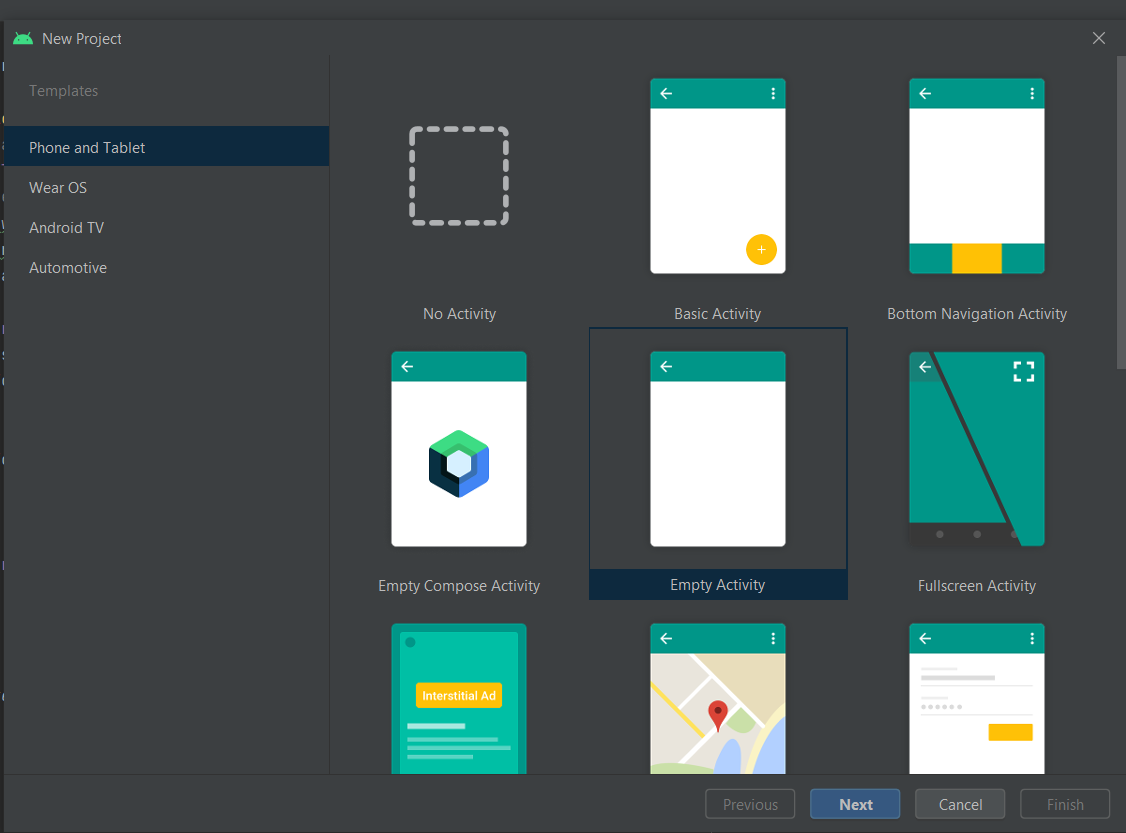
**Solution**:

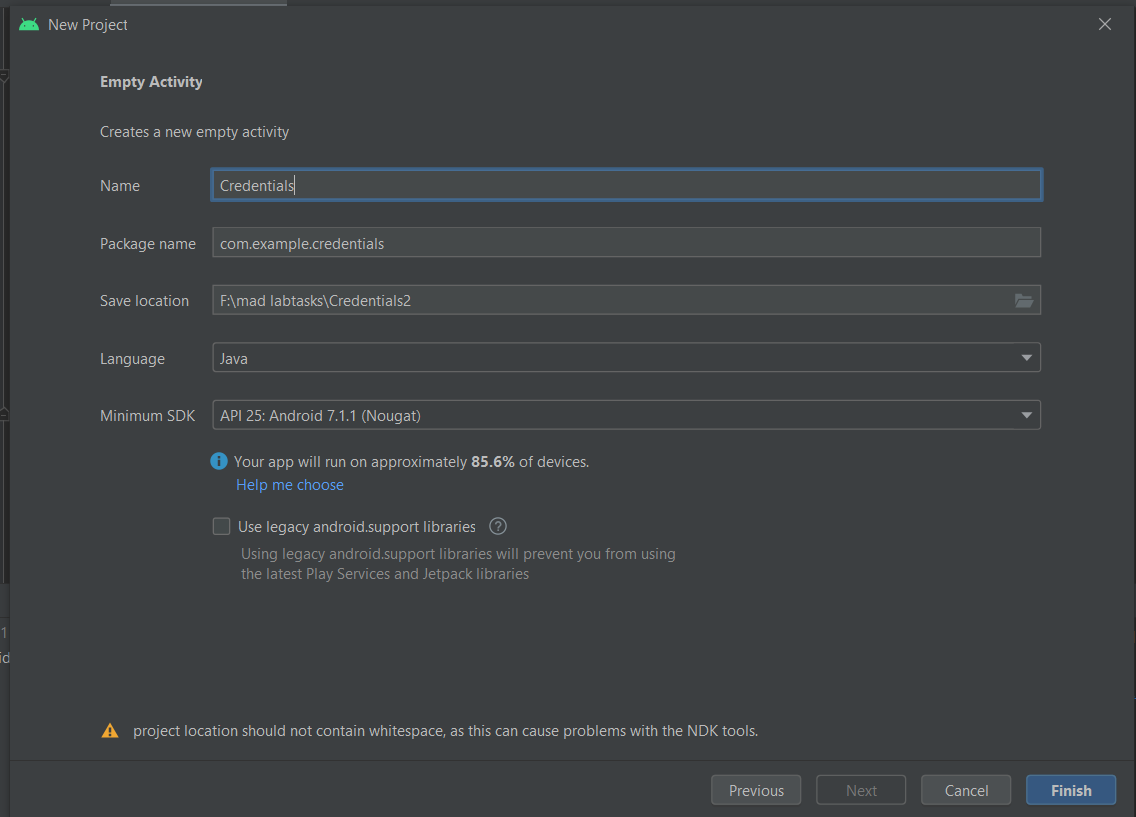
**Four steps:**

1. Project creation
2. Creating a Form (.XML file)
3. Android Form Validation
4. Show and Hide the password
5. Validate the input & following the requirements and Fetching data on screen (MainActivity.java).
6. Testing the App.

**Step 1: Project Creation**:

Open Android Studio and create a new project with an empty activity.

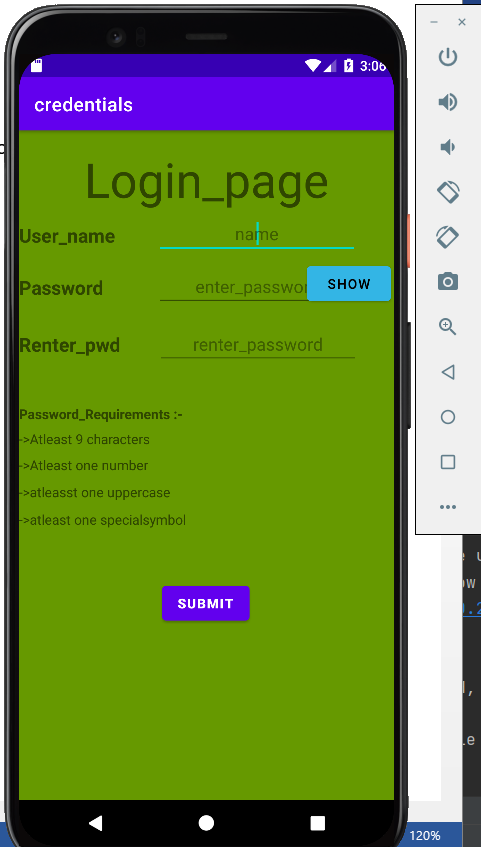


Wait for Android Studio to finish creating your project.

**Step 2: Creating a Form**:

Open app > res > layout > activity\_main.xml. This file defines the layout for the user interface (UI). A UI in Android is defined in XML files.

In this project, we will use the ConstraintLayout. A ConstraintLayout is a layout that provides a flexible way for creating views. For our project, I have designed the screen given below.



In the above screen we have 9 TextViews, 3 EditTexts and 2 Buttons. Now we will validate these fields for the connect input when the submit button is pressed.

1. **Add a Title to the Layout**

In the Palette panel, click Text and add it to the design. You might need to move it around so that it stays at the top of the layout. This TextView control will display the form's description and purpose to the user.

**The XML for the TextView** should look like this:

<TextView  
 android:id="@+id/Login\_page"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login\_page"  
 android:gravity="center"  
 android:textSize="50dp"  
 android:padding="13dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.007"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="16dp" />  
Notice the **id** attribute on the text view; it is recommended that each control element is assigned an **id** so that it's easy to reference from the Java files.

**ii) Add a Name field**

Use ViewText element to label the text field.

The **XML for the TextView** should look like this:

<TextView  
 android:id="@+id/User\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User\_name"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.00"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/Login\_page"  
 app:layout\_constraintVertical\_bias="0.137" />  
For the “name“field, an EditText element is used to enter and modify text. Add an EditText control for the name just below the TextView. You must also specify the input type attribute.

In this case, we will use android:inputType="text" for plain text. Here is the full XML.

1. <EditText  
    android:id="@+id/name"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:hint="name"  
    android:gravity="center"  
    android:ems="10"  
    android:inputType="textPersonName"  
    app:layout\_constraintBottom\_toBottomOf="parent"  
    app:layout\_constraintEnd\_toEndOf="parent"  
    app:layout\_constraintHorizontal\_bias="0.790"  
    app:layout\_constraintStart\_toStartOf="parent"  
    app:layout\_constraintTop\_toTopOf="parent"  
    app:layout\_constraintVertical\_bias="0.131" />

**iii) Add an Password Field to the Layout**

Add another EditText below the phone field. This EditText will be used for the address of the user. You might also want to increase the space provided for the address field. The attribute for this element will be android : inputType =" textPsaaword ".

1. <EditText  
    android:id="@+id/enter\_password"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:ems="10"  
    android:gravity="center"  
    android:hint="enter\_password"  
    android:inputType="textPassword"  
    app:layout\_constraintBottom\_toBottomOf="parent"  
    app:layout\_constraintEnd\_toEndOf="parent"  
    app:layout\_constraintHorizontal\_bias="0.791"  
    app:layout\_constraintStart\_toStartOf="parent"  
    app:layout\_constraintTop\_toTopOf="parent"  
    app:layout\_constraintVertical\_bias="0.215" />

**iv) Add a Button to the layout**

Lastly, add the submit button and also add the Show/hide button for password hiding and showing.You can change the background color of the button to fit your brand's identity. The most important attribute of the button is the onClick attribute.

<Button  
 android:id="@+id/showHideBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Show"  
 tools:layout\_editor\_absoluteX="322dp"  
 tools:layout\_editor\_absoluteY="147dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.989"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.209"/>

<Button  
 android:id="@+id/submit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="submit"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.721"  
 />  
**v) Add a Textview to the Layout**

In the Palette panel, click Text and add it to the design. You might need to move it around so that it stays at the top of the layout. To add the requirements to the password.

<TextView  
 android:id="@+id/Requirements"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password\_Requirements :-"  
 android:textStyle="bold"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="274dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.421"/>

<?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"  
 android:background="@android:color/holo\_green\_dark"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/Login\_page"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login\_page"  
 android:gravity="center"  
 android:textSize="50dp"  
 android:padding="13dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.007"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="16dp" />  
  
 <TextView  
 android:id="@+id/User\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User\_name"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.00"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/Login\_page"  
 app:layout\_constraintVertical\_bias="0.137" />  
  
 <EditText  
 android:id="@+id/name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:hint="name"  
 android:gravity="center"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.790"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.131" />  
  
 <TextView  
 android:id="@+id/Password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@id/Login\_page"  
 app:layout\_constraintVertical\_bias="0.218" />  
  
 <EditText  
 android:id="@+id/enter\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:gravity="center"  
 android:hint="enter\_password"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.791"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.215" />  
  
 <TextView  
 android:id="@+id/Renter\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Renter\_pwd"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.312" />  
  
 <EditText  
 android:id="@+id/confirm\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:gravity="center"  
 android:hint="renter\_password"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.796"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.307" />  
  
 <Button  
 android:id="@+id/submit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="submit"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.721"  
 />  
  
 <TextView  
 android:id="@+id/Requirements"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password\_Requirements :-"  
 android:textStyle="bold"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="274dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.421"/>  
  
 <TextView  
 android:id="@+id/character"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="->Atleast 9 characters"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="305dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.459"/>  
  
 <TextView  
 android:id="@+id/number"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="->Atleast one number"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="336dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.499"/>  
  
 <TextView  
 android:id="@+id/uppercase"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="->atleasst one uppercase"  
 tools:layout\_editor\_absoluteX="13dp"  
 tools:layout\_editor\_absoluteY="367dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.541"/>  
  
 <TextView  
 android:id="@+id/specialsymbol"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="->atleast one specialsymbol"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="395dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.583"/>  
  
 <Button  
 android:id="@+id/showHideBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Show"  
 tools:layout\_editor\_absoluteX="322dp"  
 tools:layout\_editor\_absoluteY="147dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.989"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.209"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Step 3: Android Form Validation**

* Adding **AwesomeValidation** Library

So we are going to use AwesomeValidation. So first come inside app level **build.gradle** file and add the following line inside dependencies block.

implementation 'com.basgeekball:awesome-validation:4.2'  
implementation group: 'com.google.guava', name: 'guava', version: '27.0.1-android'

* Now after adding the above line **sync** your project.

**Step 4: MainActivity.java.**

What we need to do?

* First we define an AwesomeValidation object.
* We initialize the object with BASIC style, we have two more different style COLORATION and UNDERLABEL.
* Then we add the validation to required EditTexts using addValidation() method with AwesomeValidation object.
* addValidation(): This method takes 4 arguments.

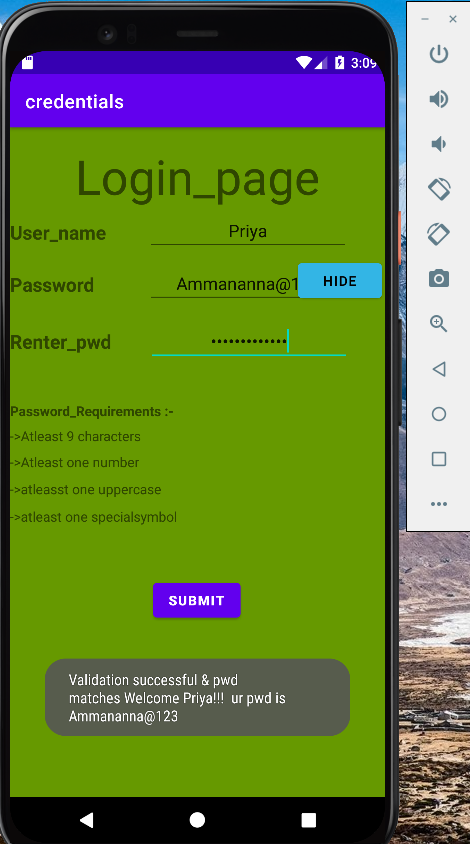
1. Activity Context, we used this to pass the current activity context.
2. Resource id of EditText in which we want to add validation, we used R.id.viewid
3. Regex String for adding validation, you can use Regex for more kind of validations
4. String resource id for error message to show the error if validation fails

* And also we follow the requirements of Password and see whether the Password matches the re-entered Password.
* Then in setOnClickListener () method we are validating the inputs with validate() method. The method returns true or false, true if validation succeeds and false if validation fails. To validate and follow the requirements and also fetch the data entered on screen , the modified code is as below:
* package com.example.credentials;  
    
  import androidx.appcompat.app.AppCompatActivity;  
    
  import android.content.Intent;  
  import android.os.Bundle;  
  import android.util.Patterns;  
  import android.view.View;  
  import android.widget.Button;  
  import android.widget.EditText;  
  import android.widget.Toast;  
  import android.text.method.PasswordTransformationMethod;  
    
  import com.basgeekball.awesomevalidation.AwesomeValidation;  
  import com.basgeekball.awesomevalidation.ValidationStyle;  
  import com.basgeekball.awesomevalidation.utility.RegexTemplate;  
    
    
    
  public class MainActivity extends AppCompatActivity {  
   //Initialize Variables  
   EditText name,pwd,rpwd;  
   Button sub,show;  
   Boolean ischar=false;  
   AwesomeValidation awesomeValidation;  
   @Override  
   protected void onCreate(Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
   setContentView(R.layout.*activity\_main*);  
   //Assign variables  
   name=findViewById(R.id.*name*);  
   pwd=findViewById(R.id.*enter\_password*);  
   rpwd=findViewById(R.id.*confirm\_password*);  
   sub=findViewById(R.id.*submit*);  
   show=findViewById(R.id.*showHideBtn*);  
   //Initialize validation Style  
   awesomeValidation =new AwesomeValidation(ValidationStyle.*BASIC*);  
   //Add Validation For Name  
   awesomeValidation.addValidation(this,R.id.*name*, RegexTemplate.*NOT\_EMPTY*,R.string.*invalid\_name*);  
   //Add Validation For Password  
   awesomeValidation.addValidation(this,R.id.*enter\_password*,".{9,}",R.string.*invalid\_pwd*);  
   //Add Validation For Renter\_password  
   awesomeValidation.addValidation(this,R.id.*confirm\_password*,".{9,}",R.string.*invalid\_rpwd*);  
   show.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View view) {  
   if(show.getText().toString().equals("Show")){  
   pwd.setTransformationMethod(null);  
   show.setText("Hide");  
   } else{  
   pwd.setTransformationMethod(new PasswordTransformationMethod());  
   show.setText("Show");  
   }  
   }  
   });  
    
   sub.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   //Check Validation  
   if (awesomeValidation.validate()){  
   //On Success  
   String pswd=pwd.getText().toString();  
   String confpswd=rpwd.getText().toString();  
   if(pswd.matches("(.\*[A**-**Z].\*)") && pswd.matches("^(?=.\*[\_.()$&@]).\*$") && pswd.matches("(.\*[0**-**9].\*)") )  
   {  
   ischar = true;  
   if(!pswd.equals(confpswd)){  
   Toast.*makeText*(getApplicationContext(),"Make sure that ur pwd matches",Toast.*LENGTH\_SHORT*).show();  
   }  
   else{  
   Toast.*makeText*(getApplicationContext(),"Validation successful & pwd matches" +" Welcome "+name.getText().toString()+ "!!! ur pwd is "+pwd.getText().toString(),Toast.*LENGTH\_SHORT*).show();  
    
   }  
    
    
   }else{  
   ischar = false;  
   Toast.*makeText*(getApplicationContext(),"make sure to follow pwd requirements",Toast.*LENGTH\_SHORT*).show();  
   }  
    
   } else{  
   Toast.*makeText*(getApplicationContext(),"Validation Failed",Toast.*LENGTH\_SHORT*).show(); } }  
   }); }}

**Step 5: Testing the App.**

Now you can run the app to test whether it is working or not.

* If validation success ,Follows the password requirements and password matches then we fetch the data entered as shown in the output.



* You can see in the above screenshot it is working absolutely fine.