**Hybrid Lab Assignment**

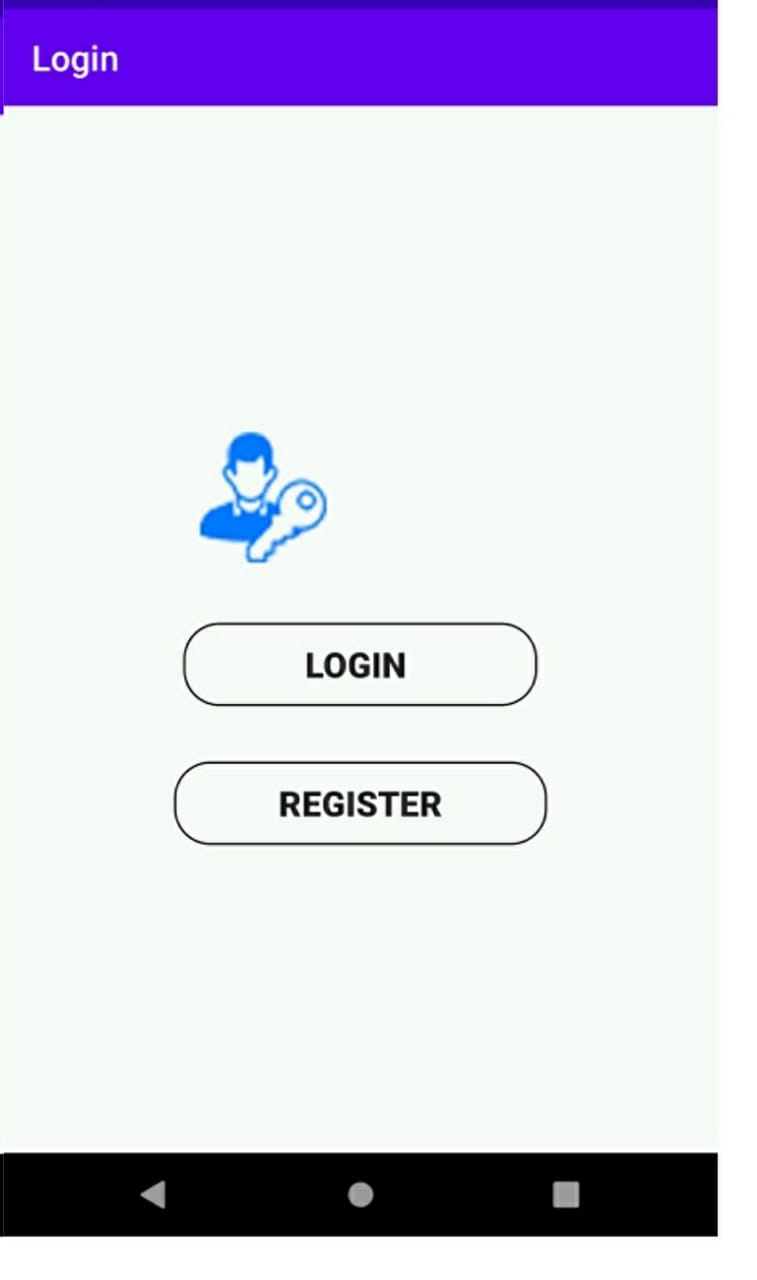
Name - Fazal Rehman

USN - 1NT17IS058

### **PROBLEM - Login Registration with SQlite Database in Android**

### (login screen)

**SCREEN-1**



## 

## 

**SCREEN-2**

## 

## 

## 

**SCREEN-3**

## 

## 

## 

## 

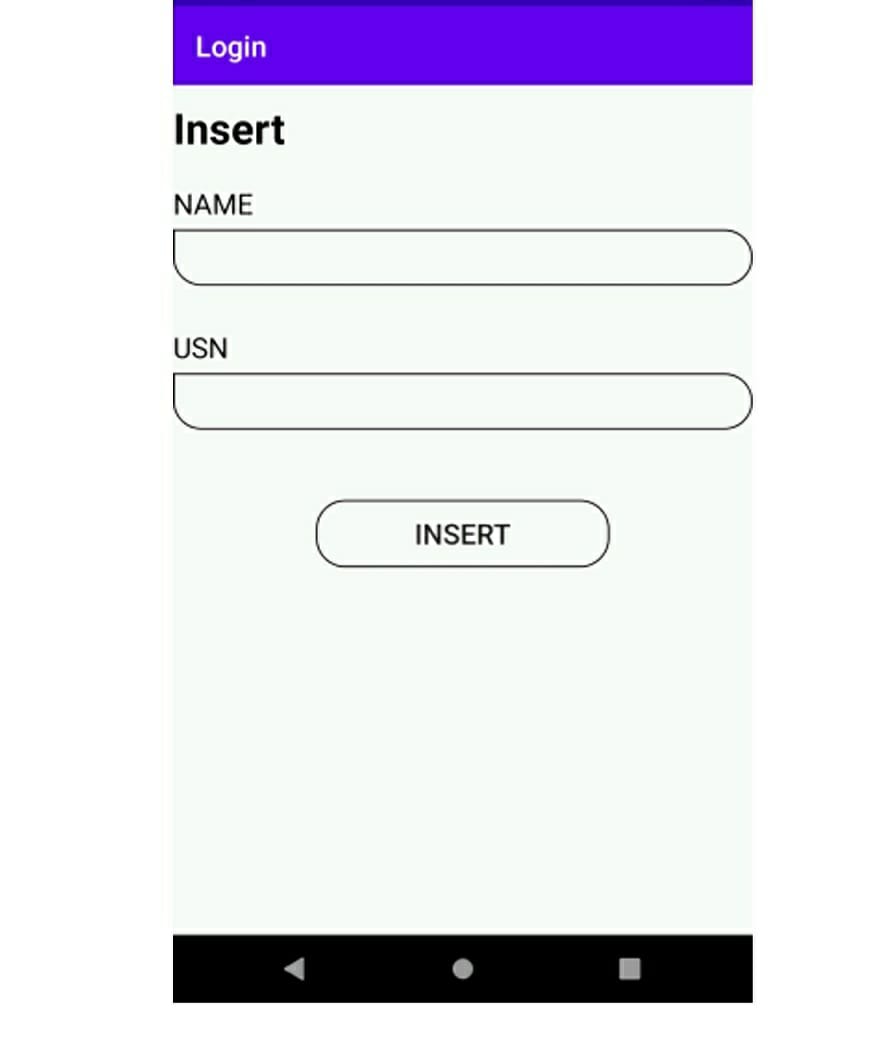
## 

## 

## 

## 

## **SCREEN-4**



## 

## 

## 

## **CODE -**

## **1) User.java :**

### **Model Class**

package com.example.login\_registration\_demo;

public class User {

private int id;

private String name;

private String email;

private String password;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

### 

### 

### **2)**

### 

### **DatabaseHelper .java**

### **Database Helper class**

package com.example.login\_registration\_demo;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {

// Database Version

private static final int DATABASE\_VERSION = 1;

// Database Name

private static final String DATABASE\_NAME = "UserManager.db";

// User table name

private static final String TABLE\_USER = "user";

// User Table Columns names

private static final String COLUMN\_USER\_ID = "user\_id";

private static final String COLUMN\_USER\_NAME = "user\_name";

private static final String COLUMN\_USER\_EMAIL = "user\_email";

private static final String COLUMN\_USER\_PASSWORD = "user\_password";

// create table sql query

private String CREATE\_USER\_TABLE = "CREATE TABLE " + TABLE\_USER + "("

+ COLUMN\_USER\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + COLUMN\_USER\_NAME + " TEXT,"

+ COLUMN\_USER\_EMAIL + " TEXT," + COLUMN\_USER\_PASSWORD + " TEXT" + ")";

// drop table sql query

private String DROP\_USER\_TABLE = "DROP TABLE IF EXISTS " + TABLE\_USER;

DatabaseHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(CREATE\_USER\_TABLE);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

//Drop User Table if exist

db.execSQL(DROP\_USER\_TABLE);

// Create tables again

onCreate(db);

}

public void addUser(User user) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(COLUMN\_USER\_NAME, user.getName());

values.put(COLUMN\_USER\_EMAIL, user.getEmail());

values.put(COLUMN\_USER\_PASSWORD, user.getPassword());

// Inserting Row

db.insert(TABLE\_USER, null, values);

db.close();

}

public boolean checkUser(String email) {

// array of columns to fetch

String[] columns = {

COLUMN\_USER\_ID

};

SQLiteDatabase db = this.getReadableDatabase();

// selection criteria

String selection = COLUMN\_USER\_EMAIL + " = ?";

// selection argument

String[] selectionArgs = {email};

Cursor cursor = db.query(TABLE\_USER, //Table to query

columns, //columns to return

selection, //columns for the WHERE clause

selectionArgs, //The values for the WHERE clause

null, //group the rows

null, //filter by row groups

null); //The sort order

int cursorCount = cursor.getCount();

cursor.close();

db.close();

if (cursorCount > 0) {

return true;

}

return false;

}

public boolean checkUser(String email, String password) {

// array of columns to fetch

String[] columns = {COLUMN\_USER\_ID};

SQLiteDatabase db = this.getReadableDatabase();

String selection = COLUMN\_USER\_EMAIL + " = ?" + " AND " + COLUMN\_USER\_PASSWORD + " = ?";

String[] selectionArgs = {email, password};

Cursor cursor = db.query(TABLE\_USER, //Table to query

columns, //columns to return

selection, //columns for the WHERE clause

selectionArgs, //The values for the WHERE clause

null, //group the rows

null, //filter by row groups

null); //The sort order

int cursorCount = cursor.getCount();

cursor.close();

db.close();

if (cursorCount > 0) {

return true;

}

return false;

}

}

### **4) activity\_login.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.core.widget.NestedScrollView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/nestedScrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorBackground"

android:paddingLeft="20dp"

android:paddingTop="20dp"

android:paddingRight="20dp"

android:paddingBottom="20dp">

<androidx.appcompat.widget.LinearLayoutCompat

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp">

<EditText

android:id="@+id/et\_Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_email"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_Password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<EditText

android:id="@+id/et\_Password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/appCompatButtonLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:background="@color/colorTextHint"

android:text="@string/text\_login" />

<androidx.appcompat.widget.AppCompatTextView

android:id="@+id/textViewLinkRegister"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:gravity="center"

android:text="@string/text\_not\_member"

android:textSize="16dp" />

</androidx.appcompat.widget.LinearLayoutCompat>

</androidx.core.widget.NestedScrollView>

### **5) activity\_registration.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.core.widget.NestedScrollView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/nestedScrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorBackground"

android:paddingLeft="20dp"

android:paddingTop="20dp"

android:paddingRight="20dp"

android:paddingBottom="20dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_Name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp">

<EditText

android:id="@+id/et\_Name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_name"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<EditText

android:id="@+id/et\_Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_email"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_Password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<EditText

android:id="@+id/et\_Password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/lay\_ConfPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<EditText

android:id="@+id/et\_ConfPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_confirm\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white" />

</com.google.android.material.textfield.TextInputLayout>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/appCompatButtonRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:background="@color/colorTextHint"

android:text="@string/text\_register" />

<androidx.appcompat.widget.AppCompatTextView

android:id="@+id/appCompatTextViewLoginLink"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:gravity="center"

android:text="Already a member? Login"

android:textSize="16dp" />

</LinearLayout>

</androidx.core.widget.NestedScrollView>

### **6) activity\_user\_list.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorBackground"

tools:context=".UserListActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@id/textview"

android:layout\_centerInParent="true"

android:layout\_margin="20dp"

android:text="Successfully login"

android:textColor="@color/colorText"

android:textSize="16dp" />

<TextView

android:id="@+id/textview"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="Email ID"

android:textColor="@color/colorText"

android:textSize="16dp" />

</RelativeLayout>

### **7) InputValidation .java**

package com.example.login\_registration\_demo;

import android.app.Activity;

import android.content.Context;

import android.view.View;

import android.view.WindowManager;

import android.view.inputmethod.InputMethodManager;

import android.widget.EditText;

import com.google.android.material.textfield.TextInputLayout;

public class InputValidation {

private Context context;

InputValidation(Context context) {

this.context = context;

}

boolean isInputEditTextFilled(EditText editText, TextInputLayout textInputLayout, String message) {

String value = editText.getText().toString().trim();

if (value.isEmpty()) {

textInputLayout.setError(message);

hideKeyboardFrom(editText);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

boolean isInputEditTextEmail(EditText editText, TextInputLayout textInputLayout, String message) {

String value = editText.getText().toString().trim();

if (value.isEmpty() || !android.util.Patterns.EMAIL\_ADDRESS.matcher(value).matches()) {

textInputLayout.setError(message);

hideKeyboardFrom(editText);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

boolean isInputEditTextMatches(EditText editText1, EditText editText2, TextInputLayout textInputLayout, String message) {

String value1 = editText1.getText().toString().trim();

String value2 = editText2.getText().toString().trim();

if (!value1.contentEquals(value2)) {

textInputLayout.setError(message);

hideKeyboardFrom(editText2);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

private void hideKeyboardFrom(View view) {

InputMethodManager imm = (InputMethodManager) context.getSystemService(Activity.INPUT\_METHOD\_SERVICE);

imm.hideSoftInputFromWindow(view.getWindowToken(), WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN);

}

}

### **8) RegistrationActivity.java**

package com.example.login\_registration\_demo;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.AppCompatButton;

import androidx.appcompat.widget.AppCompatTextView;

import androidx.core.widget.NestedScrollView;

import com.google.android.material.snackbar.Snackbar;

import com.google.android.material.textfield.TextInputLayout;

public class RegisterActivity extends AppCompatActivity implements View.OnClickListener {

private final AppCompatActivity activity = RegisterActivity.this;

private NestedScrollView nestedScrollView;

private TextInputLayout lay\_Name;

private TextInputLayout lay\_Email;

private TextInputLayout lay\_Password;

private TextInputLayout lay\_ConfPassword;

private EditText et\_Name;

private EditText et\_Email;

private EditText et\_Password;

private EditText et\_ConfPassword;

private AppCompatButton appCompatButtonRegister;

private AppCompatTextView appCompatTextViewLoginLink;

private InputValidation inputValidation;

private DatabaseHelper databaseHelper;

private User user;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

getSupportActionBar().hide();

initViews();

initListeners();

initObjects();

}

private void initViews() {

nestedScrollView = findViewById(R.id.nestedScrollView);

lay\_Name = findViewById(R.id.lay\_Name);

lay\_Email = findViewById(R.id.lay\_Email);

lay\_Password = findViewById(R.id.lay\_Password);

lay\_ConfPassword = findViewById(R.id.lay\_ConfPassword);

et\_Name = findViewById(R.id.et\_Name);

et\_Email = findViewById(R.id.et\_Email);

et\_Password = findViewById(R.id.et\_Password);

et\_ConfPassword = findViewById(R.id.et\_ConfPassword);

appCompatButtonRegister = findViewById(R.id.appCompatButtonRegister);

appCompatTextViewLoginLink = findViewById(R.id.appCompatTextViewLoginLink);

}

private void initListeners() {

appCompatButtonRegister.setOnClickListener(this);

appCompatTextViewLoginLink.setOnClickListener(this);

}

private void initObjects() {

inputValidation = new InputValidation(activity);

databaseHelper = new DatabaseHelper(activity);

user = new User();

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.appCompatButtonRegister:

postDataToSQLite();

break;

case R.id.appCompatTextViewLoginLink:

finish();

break;

}

}

private void postDataToSQLite() {

if (!inputValidation.isInputEditTextFilled(et\_Name, lay\_Name, getString(R.string.error\_message\_name))) {

return;

}

if (!inputValidation.isInputEditTextFilled(et\_Email, lay\_Email, getString(R.string.error\_message\_email))) {

return;

}

if (!inputValidation.isInputEditTextEmail(et\_Email, lay\_Email, getString(R.string.error\_message\_email))) {

return;

}

if (!inputValidation.isInputEditTextFilled(et\_Password, lay\_Password, getString(R.string.error\_message\_password))) {

return;

}

if (!inputValidation.isInputEditTextMatches(et\_Password, et\_ConfPassword,

lay\_ConfPassword, getString(R.string.error\_password\_match))) {

return;

}

if (!databaseHelper.checkUser(et\_Email.getText().toString().trim())) {

user.setName(et\_Name.getText().toString().trim());

user.setEmail(et\_Email.getText().toString().trim());

user.setPassword(et\_Password.getText().toString().trim());

databaseHelper.addUser(user);

// Snack Bar to show success message that record saved successfully

Snackbar.make(nestedScrollView, getString(R.string.success\_message), Snackbar.LENGTH\_LONG).show();

emptyInputEditText();

} else {

// Snack Bar to show error message that record already exists

Snackbar.make(nestedScrollView, getString(R.string.error\_email\_exists), Snackbar.LENGTH\_LONG).show();

}

}

/\*\*

\* This method is to empty all input edit text

\*/

private void emptyInputEditText() {

et\_Name.setText(null);

et\_Email.setText(null);

et\_Password.setText(null);

et\_ConfPassword.setText(null);

}

}

### **9) LoginActivity.java**

package com.example.login\_registration\_demo;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.AppCompatButton;

import androidx.appcompat.widget.AppCompatTextView;

import androidx.core.widget.NestedScrollView;

import com.google.android.material.snackbar.Snackbar;

import com.google.android.material.textfield.TextInputLayout;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener {

private final AppCompatActivity activity = LoginActivity.this;

private NestedScrollView nestedScrollView;

private TextInputLayout lay\_Email;

private TextInputLayout lay\_Password;

private EditText et\_Email;

private EditText et\_Password;

private AppCompatButton appCompatButtonLogin;

private AppCompatTextView textViewLinkRegister;

private InputValidation inputValidation;

private DatabaseHelper databaseHelper;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

getSupportActionBar().hide();

initViews();

initListeners();

initObjects();

}

private void initViews() {

nestedScrollView = findViewById(R.id.nestedScrollView);

lay\_Email = findViewById(R.id.lay\_Email);

lay\_Password = findViewById(R.id.lay\_Password);

et\_Email = findViewById(R.id.et\_Email);

et\_Password = findViewById(R.id.et\_Password);

appCompatButtonLogin = findViewById(R.id.appCompatButtonLogin);

textViewLinkRegister = findViewById(R.id.textViewLinkRegister);

}

private void initListeners() {

appCompatButtonLogin.setOnClickListener(this);

textViewLinkRegister.setOnClickListener(this);

}

private void initObjects() {

databaseHelper = new DatabaseHelper(activity);

inputValidation = new InputValidation(activity);

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.appCompatButtonLogin:

verifyFromSQLite();

break;

case R.id.textViewLinkRegister:

Intent intentRegister = new Intent(getApplicationContext(), RegisterActivity.class);

startActivity(intentRegister);

break;

}

}

private void verifyFromSQLite() {

if (!inputValidation.isInputEditTextFilled(et\_Email, lay\_Email, getString(R.string.error\_message\_email))) {

return;

}

if (!inputValidation.isInputEditTextEmail(et\_Email, lay\_Email, getString(R.string.error\_message\_email))) {

return;

}

if (!inputValidation.isInputEditTextFilled(et\_Password, lay\_Password, getString(R.string.error\_message\_email))) {

return;

}

if (databaseHelper.checkUser(et\_Email.getText().toString().trim()

, et\_Password.getText().toString().trim())) {

Intent accountsIntent = new Intent(activity, UserListActivity.class);

accountsIntent.putExtra("EMAIL", et\_Email.getText().toString().trim());

emptyInputEditText();

startActivity(accountsIntent);

} else {

// Snack Bar to show success message that record is wrong

Snackbar.make(nestedScrollView, getString(R.string.error\_valid\_email\_password), Snackbar.LENGTH\_LONG).show();

}

}

private void emptyInputEditText() {

et\_Email.setText(null);

et\_Password.setText(null);

}

}

### **10) UserListactivity.java**

package com.example.login\_registration\_demo;

import android.os.Bundle;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class UserListActivity extends AppCompatActivity {

TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_user\_list);

textView = findViewById(R.id.textview);

String emailfromIntent = getIntent().getStringExtra("EMAIL");

textView.setText(emailfromIntent);

}

}

### **11) color.xml**

### <?xml version="1.0" encoding="utf-8"?>

### <resources>

### <color name="colorPrimary">#6200EE</color>

### <color name="colorPrimaryDark">#6200EE</color>

### <color name="colorAccent">#FFFFFF</color>

### <color name="colorBackground">#FF0266</color>

### <color name="colorText">#FFFFFF</color>

### <color name="colorTextHint">#51d8c7</color>

### </resources>

### **12) String.xml**

<resources>

<string name="app\_name">Login Register</string>

<string name="text\_accounts">All Accounts</string>

<string name="hint\_name">Name</string>

<string name="hint\_email">Email</string>

<string name="hint\_password">Password</string>

<string name="hint\_confirm\_password">Confirm Password</string>

<string name="text\_login">Login</string>

<string name="text\_register">Register</string>

<string name="error\_message\_name">Enter Full Name</string>

<string name="error\_message\_email">Enter Valid Email</string>

<string name="error\_message\_age">Enter Age</string>

<string name="error\_message\_password">Enter Password</string>

<string name="success\_message">Registration Successful</string>

<string name="text\_not\_member">No account yet? Create one</string>

<string name="text\_already\_member">Already a member? Login</string>

<string name="error\_email\_exists">Email Already Exists</string>

<string name="error\_password\_match">Password Does Not Matches</string>

<string name="error\_valid\_email\_password">Wrong Email or Password</string>

<string name="action\_settings">Settings</string>

<string name="text\_hello">Hello,</string>

<string name="text\_title">Android Tutorials Hub</string>

</resources>