PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043

Department of Electronics & Telecommunication Engineering

BATCH AND ROLL NO:
EXPERIMENT NO.: 03
TITLE: Design a mobile application to create the login page using sqlite / firebase
DATE OF PERFORMANCE:
DATE OF SUBMISSION:

Title: Design a mobile application to create the login page using sqlite /firebase

Requirements:

1 Android studio
2. Sqlite /firebase

Theory:

In the rapidly evolving landscape of mobile application development, creating a seamless and secure login experience is a fundamental aspect. The login page serves as the gateway for users to access the application's features and functionalities. Two widely utilized technologies for implementing login systems are SQLite and Firebase.

SQLite:

SQLite is a self-contained, serverless, and zero-configuration relational database engine. It is embedded into the mobile application to handle local data storage efficiently. For mobile applications, SQLite provides a lightweight and efficient solution to manage databases directly on the user's device. In this lab, we will explore the integration of SQLite to design a local database for storing user credentials securely.

Firebase:

Firebase, on the other hand, is a comprehensive mobile and web application development platform provided by Google. Firebase offers a real-time NoSQL database, allowing for seamless synchronization of data between different devices. Additionally, Firebase Authentication simplifies the process of user authentication, providing a secure and scalable solution for managing user logins in mobile applications.

Objective of the Lab:

The primary objective of this lab is to guide you through the process of designing a mobile application login page. You will have the opportunity to choose between two robust technologies: SQLite for local database storage or Firebase for a cloud-based solution. By the end of this lab, you should be proficient in implementing a secure and user-friendly login system in your mobile application.

Lab Prerequisites:

- Basic understanding of mobile application development concepts.
- Familiarity with the chosen development environment (e.g., Android Studio).
- Prior knowledge of programming languages such as Java (for Android)

Steps:

Using SQLite:

Step 1: Set Up SQLite Database

- Create a SQLite database to store user credentials.
- Define a table structure to hold user information, including fields such as username and password.
- Implement methods to create, read, update, and delete user records in the SQLite database.

Step 2: Design the Login Page UI

- Create a login page UI with input fields for username and password.
- Include a "Login" button that triggers the authentication process.

Step 3: Authenticate User

- Retrieve user input from the login page.
- Query the SQLite database to verify the entered username and password.
- Grant access if the credentials are valid; otherwise, display an error message.

Common Steps:

Step 1: Handle User Input

- Implement error handling for invalid inputs on the login page.
- Validate and sanitize user input to enhance security.

Step 2: Test Your Implementation

- Test the login functionality thoroughly, considering various scenarios (valid and invalid credentials, edge cases).
- Debug and resolve any issues that may arise during testing.

Step 3: Enhance Security

- Implement secure coding practices to protect user data.

XML Code:

```
<?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"

    tools:context=".MainActivity">


    <EditText

        android:id="@+id/editTextText"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:ems="10"

        android:hint="Username"

        android:inputType="text"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.497"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.176" />


    <TextView
```

```
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Registration Form"
    android:textColor="#7C40ED"
    android:textSize="24sp"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toTopOf="@+id/editTextText"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/editTextText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:ems="10"
    android:inputType="textPassword"
    android:hint="Password"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
```

```
app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/editTextText"

    app:layout_constraintVertical_bias="0.0" />


  <EditText

    android:id="@+id/editTextText3"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_marginTop="36dp"

    android:ems="10"

    android:hint="ReType Password"

    android:inputType="textPassword"

    app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintHorizontal_bias="0.497"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toBottomOf="@+id/editTextText2" />


  <Button

    android:id="@+id/button"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_marginTop="44dp"

    android:text="Submit"

    android:textSize="20sp"
```

```
app:layout_constraintEnd_toEndOf="parent"

    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toBottomOf="@+id/editTextText3" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

JAVA Code:

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Button;
import android.widget.Toast;

import com.example.myapplication.DBhelper;

public class MainActivity extends AppCompatActivity {
   EditText user,pass,repass;
   Button btn;

   DBhelper db;



   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      btn = findViewById(R.id.button);
      db = new DBhelper(this);
      user = findViewById(R.id.editTextText);
      pass = findViewById(R.id.editTextText2);
      repass = findViewById(R.id.editTextText3);

  btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
```

```
            String username = user.getText().toString();
            String password = pass.getText().toString();

  String repassword = repass.getText().toString();

            if(password.equals(repassword))
            {
               Boolean checkuser = db.checkUserName(username);
               if(checkuser == false)
               {
                  Boolean insert =  db.insertdata(username,password);
                  if (insert == true)
                  {
                     Toast.makeText(MainActivity.this,"Registration
Successful",Toast.LENGTH_SHORT).show();
                  }
                  else {
                     Toast.makeText(MainActivity.this,"Registration
Unsuccessful",Toast.LENGTH_SHORT).show();
                  }
               }
               else{
                  Toast.makeText(MainActivity.this,"Username already
exits",Toast.LENGTH_SHORT).show();
               }
            }
            else{
               Toast.makeText(MainActivity.this,"Password do not
match",Toast.LENGTH_SHORT).show();
            }


         }
      });
   }
}
```

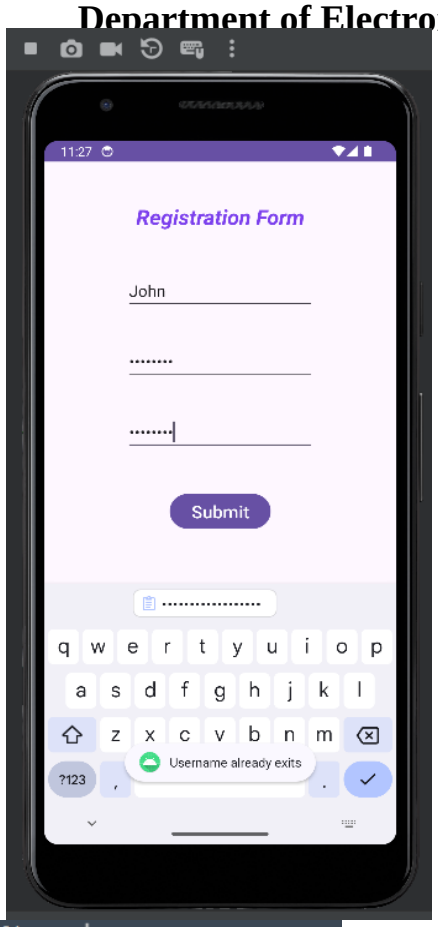
1) DBHelper.java

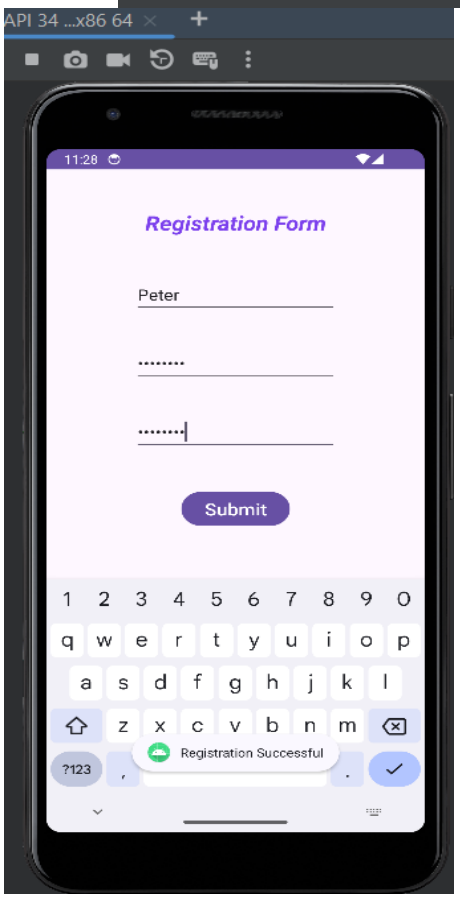
package com.example.myapplication;

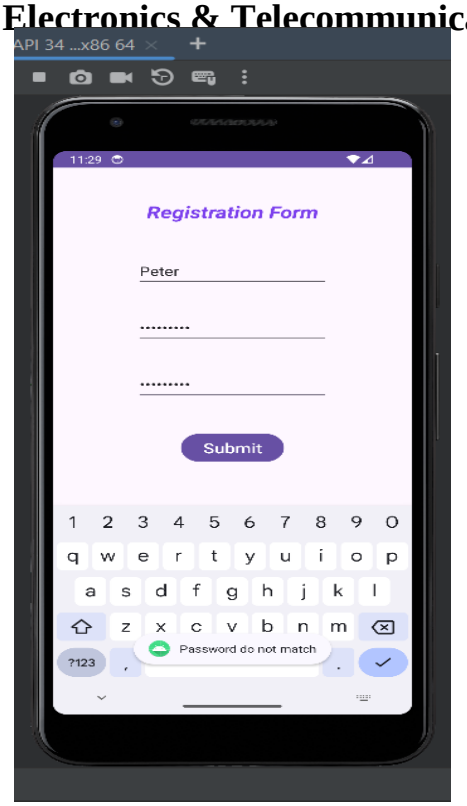
```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.view.View;

import androidx.annotation.Nullable;

public class DBhelper extends SQLiteOpenHelper {
public DBhelper(Context context) {
super(context, "Login.db", null, 1);
}



@Override
public void onCreate(SQLiteDatabase db) {
db.execSQL("create table users(username Text primary key ,
password Text)");
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
db.execSQL("drop table if exists users");
}
public Boolean insertdata(String user,String password)
{
SQLiteDatabase db = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();
contentValues.put("username",user);
contentValues.put("password",password);

long result = db.insert("users",null,contentValues);
if(result==-1)
{
return false;//insertion is failed
}
else{
return true;
}
}

public Boolean checkUserName(String user)
{
SQLiteDatabase db = this.getWritableDatabase();
```

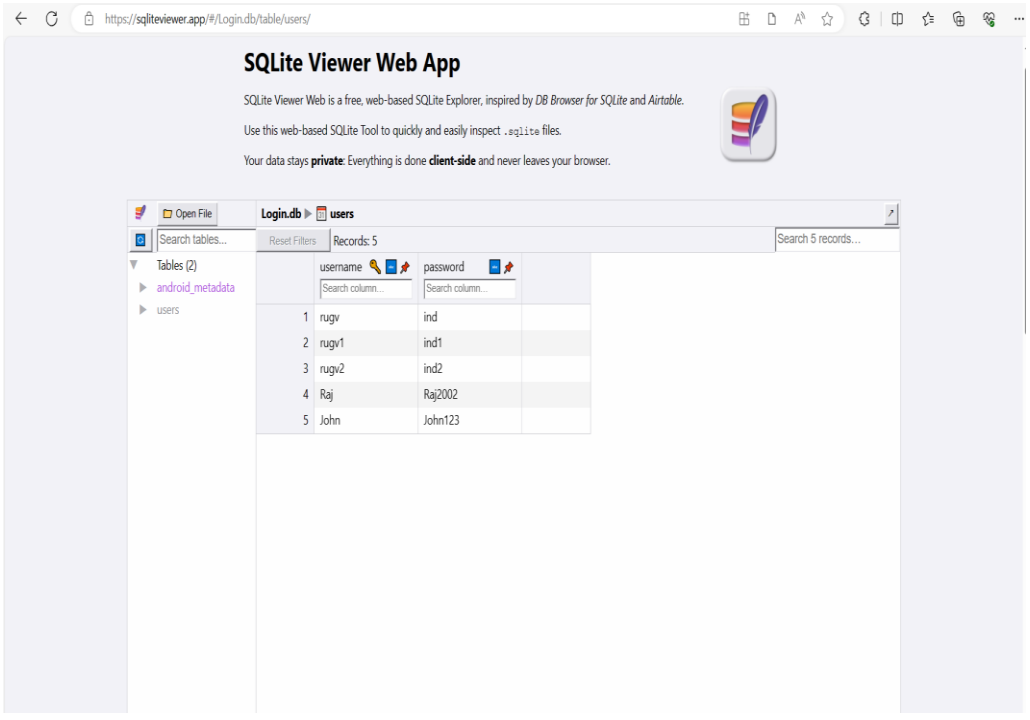
```
Cursor cursor = db.rawQuery("select * from users where
username=?", new String[]{user});
if(cursor.getCount()>0)
{
return true;
}
else
{
return false;
}



}


}
```

Output:


11:27
Registration Form
John
Submit
Username already exits


API 34 ...x86 64
11:28
Registration Form
Peter
Submit
Registration Successful


API 34 ...x86 64
11:29
Registration Form
Peter
Submit
Password do not match


https://sqliteviewer.app/#/Login.db/table/users/
SQLite Viewer Web App
SQLite Viewer Web is a free, web-based SQLite Explorer, inspired by DB Browser for SQLite and Airtable.
Use this web-based SQLite Tool to quickly and easily inspect .sqlite files.
Your data stays private: Everything is done client-side and never leaves your browser.
Open File
Login.db
users
Search tables...
Reset Filters
Records: 5
Search 5 records...
Tables (2)
android_metadata
users
username
password
Search column...
1 rugv ind
2 rugv1 ind1
3 rugv2 ind2
4 Raj Raj2002
5 John John123

Conclusion:

……

……

……