

|  |
| --- |
| **BATCH AND ROLL NO: R6, 42257** |
| **EXPERIMENT NO.: 03** |
| **TITLE:** Design a mobile application to create the login page using sqlite /firebase |
| **DATE OF PERFORMANCE:** |
| **DATE OF CHECKING:** |

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

# Requirements:

1 Android studio

1. Sqlite /firebase

# Theory:

In software applications, it is mostly required to store users and app data locally. Android SDK provides several APIs for developers to save user and app data, So SQLite is one of the ways of storing data. For many applications, SQLite is the backbone of the app whether it’s used directly or via some third-party wrapper. SQLite is a lightweight database that comes with android. It is an Open-Source embedded SQL database engine. This provides a relational database management structure for storing user-defined records in the form of tables.

Here are the general steps to create login page using SQLite in Android Studio:

* 1. Create a Home Screen JAVA Activity, Which will hold ‘Sign In‘and ‘Sign Up‘options.
  2. Create XML layouts for home screen and ‘Sign In‘and ‘Sign Up‘Screens.
  3. Create a SQLite Database in the application’s context, so that we can save all required user data (‘Sign In‘and ‘Sign Up‘details).
  4. Code Logic in Application’s Java files.
  5. Run Android Login Registration app on Device/Emulator.



# Code:

import android.content.Intent; import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.Button; import android.widget.EditText; import android.widget.Toast;

public class LoginActivity extends AppCompatActivity { private EditText mUsernameEditText;

private EditText mPasswordEditText;

private Button mLoginButton; private SQLiteDatabase mDatabase;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_login);

// Initialize the UI components

mUsernameEditText = findViewById(R.id.username); mPasswordEditText = findViewById(R.id.password); mLoginButton = findViewById(R.id.login);

// Open the database connection

mDatabase = openOrCreateDatabase("mydatabase.db", MODE\_PRIVATE, null);

// Create the users table if it doesn't exist

mDatabase.execSQL("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY

AUTOINCREMENT, username TEXT, password TEXT)");

// Set the click listener for the login button mLoginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username = mUsernameEditText.getText().toString(); String password = mPasswordEditText.getText().toString();

// Query the database for the user with the given username and password

Cursor cursor = mDatabase.rawQuery("SELECT \* FROM users WHERE username = ? AND password = ?", new String[]{username, password});

// Check if the query returned any results if (cursor.getCount() > 0) {

// The user is authenticated, start the main activity

Intent intent = new Intent(LoginActivity.this, MainActivity.class); startActivity(intent);

finish();

} else {

// The user is not authenticated, display an error message

Toast.makeText(LoginActivity.this, "Invalid username or password", Toast.LENGTH\_SHORT).show();

}

}

});

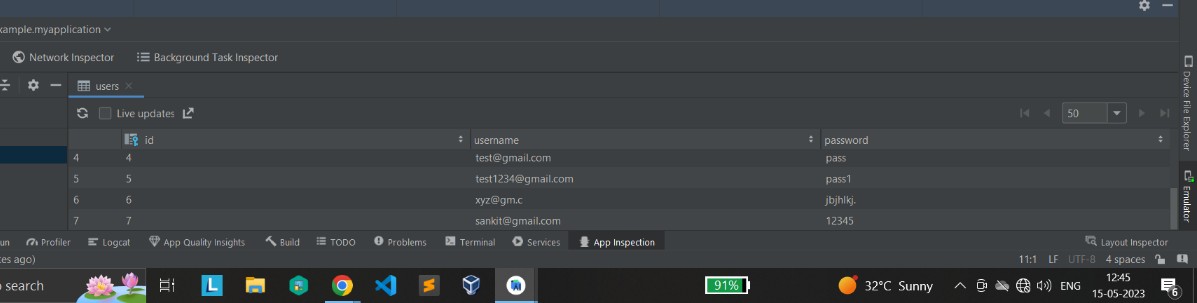
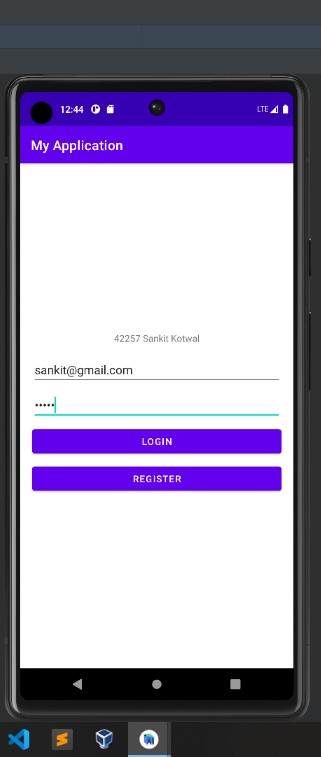
}

}

// Close the cursor and database connection cursor.close();

mDatabase.close();





# Output:

**Conclusion:**