# Experiment No. - 10

**Student Name: Vivek Kumar** **UID: 21BCS8129**
**Branch: BE-CSE(LEET)** **Section/Group:20BCS-ST-801/B**
**Semester: 6th** **Date of Performance: 03/05/2023**
**Subject Name: Mobile Application Development Lab** **Subject Code: 20CSP-356**

1. **Aim:**
   **How to Create and Add Data to SQLite Database in Android?**

2. **Objective:**
   **Understanding and analyse the specific requirement, possibilities and challenges when developing for a mobile application context.**

3. **System Requirements:**
   - Microsoft Windows 7/8/10 (32-bit or 64-bit)
   - 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
   - 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
   - 1280 x 800 minimum screen resolution
   - Java JDK5 or later version
   - Java Runtime Environment (JRE) 6 or higher.

4. **Theory:**

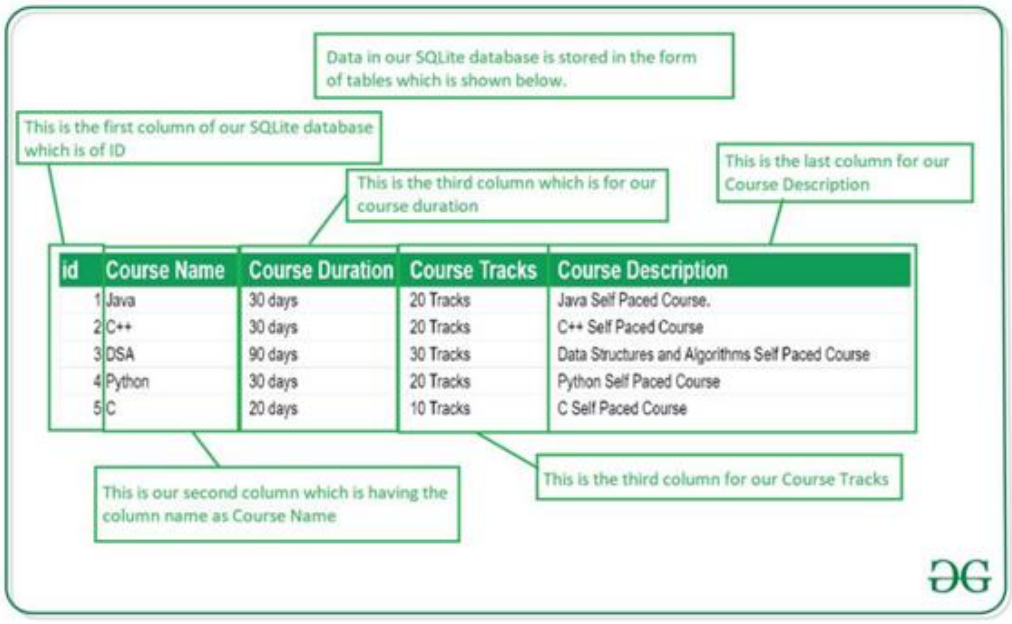
SQLite is another data storage available in Android where we can store data in the user's device and can use it any time when required

### What is SQLite Database?

SQLite Database is an open-source database provided in Android which is used to store data inside the user's device in the form of a Text file. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data. SQLite is an offline database that is locally stored in the user's device and we do not have to create any connection to connect to this database.

### How Data is Being Stored in the SQLite Database?

Data is stored in the SQLite database in the form of **tables**. When we stored this data in our SQLite database it is arranged in the form of tables that are similar to that of an excel sheet. Below is the representation of our SQLite database which we are storing in our SQLite database.




### Important Methods in SQLite Database

Below are the several important methods that we will be using in this SQLite database integration in Android.

| Method | Description |
|---|---|
| getColumnNames() | This method is used to get the Array of column names of our SQLite table. |
| getCount() | This method will return the number of rows in the cursor. |
| isClosed() | This method returns a Boolean value when our cursor is closed. |
| getColumnCount() | This method returns the total number of columns present in our table. |
| getColumnName(int columnIndex) | This method will return the name of the column when we passed the index of our column in it. |
| getColumnIndex(String columnName) | This method will return the index of our column from the name of the column. |
| getPosition() | This method will return the current position of our cursor in our table. |
| getPosition() | This method will return the current position of our cursor in our table. |

We will be building a simple application in which we will be adding data to the SQLite database. We will be creating a database for adding course name, course description, course duration, and course tracks. We will be saving all this data in our SQLite database. A sample video is given below to get an idea about what we are going to do in this article. Note that we are going to implement this project using the **Java** language.

5. **Steps/Program:**
   - You will use Android Studio IDE to create an Android application and name it as tutorials point under a package in.innovateria.ws10application2
   - Modify src/MainActivity.java and add the appropriate code for the Option Page.
   - Create 3 Extra Activities such as RegisterActivity, Login Activity and Home Activity.
   - Add the Respective XML design for the all Pages such as Login and Register Design.
   - Create a DBHelper Class and extends SQLiteOpenHelper Class into it and Implements all the Required methods such as Insert, Update, Get, and Create functions.
   - Create a User Model to handler the Data from the Database or SQLite.
   - Create a Opbject for DBHelper class and While Registering the user use we need to always use the dbHelper.insetUserData(name1,number1,email1,pass1) function with all required fields.
   - Create and Modify the Drawable and res/values/strings.xml. Android studio takes care of default constants.
   - Complete the Login Functionality for the User in The LoginActivity page.
   - No need to Modify AndroidManifest.xml and Add the Permissions.
   - Run the application to launch Android emulator and verify the result of the changes done in the application.

6. **Code:**

**activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@drawable/background_main"
    app:layout_goneMarginBottom="10dp">
    <include layout="@layout/toolbar"
        android:id="@+id/tool_main"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        tools:ignore="MissingConstraints"
        >

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_marginBottom="10dp"
            app:tint="#000000"
            tools:srcCompat="@drawable/logo" />
```

```
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="10dp"
        android:fontFamily="cursive"
        android:gravity="center"
        android:padding="10dp"
        android:text="Innovateria"
        android:textSize="60sp" />

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@drawable/btn_view"
        android:fontFamily="sans-serif-black"
        android:padding="10dp"
        android:text="Login"
        android:textColor="#1B1515"
        android:textSize="18sp" />

    <Button
        android:id="@+id/btnSignUp"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:background="@drawable/btn_view"
        android:fontFamily="sans-serif-black"
        android:padding="10dp"
        android:text="Sign Up"
        android:textColor="#1B1515"
        android:textSize="18sp" />
  </LinearLayout>
</RelativeLayout>
```

**MainActivity.java**

```
package in.innovateria.ws10application;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
  Button login, Reg;
```

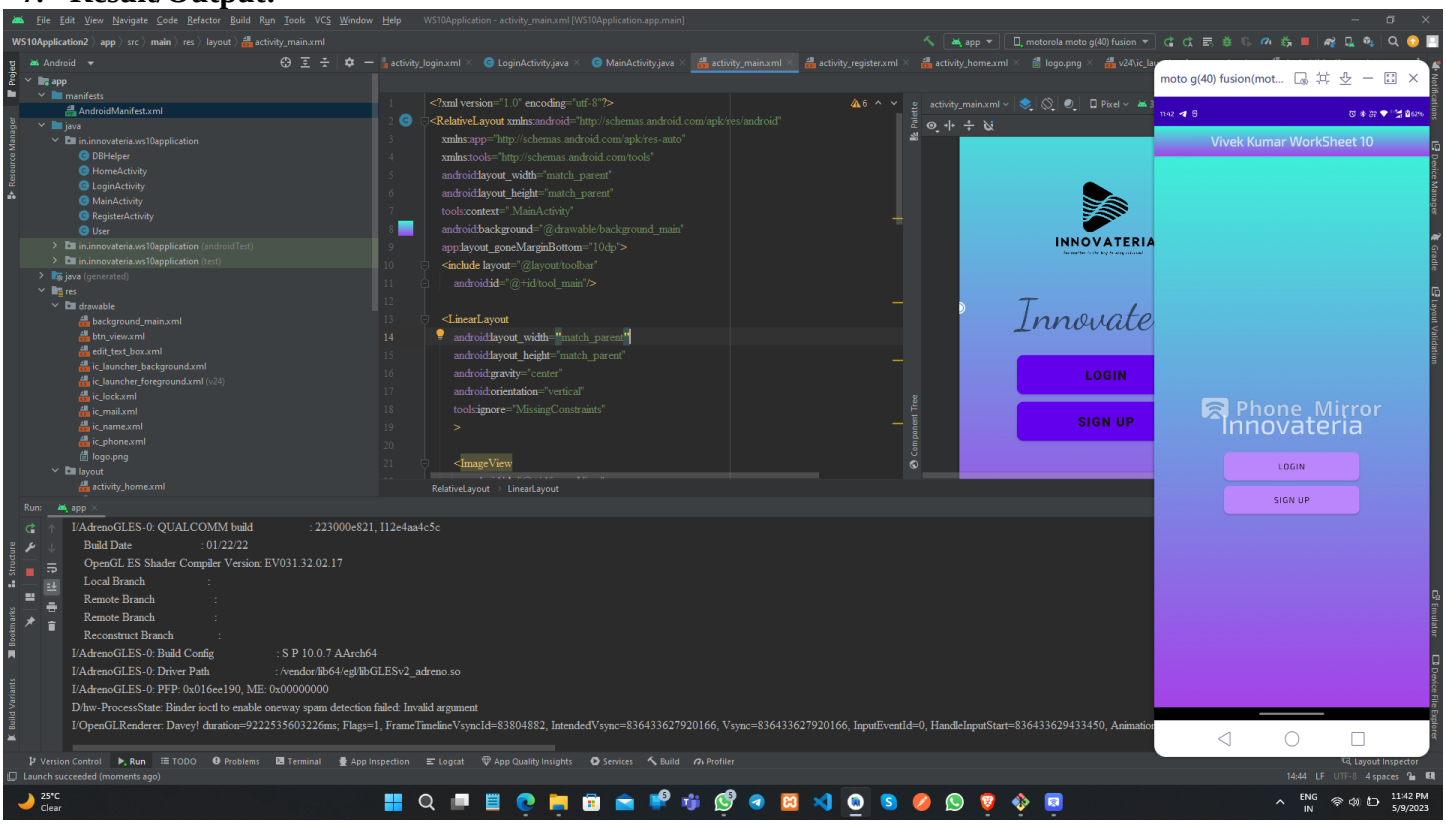
```
    Toolbar toolbar;
    DBHelper dbHelper;

    @Override
    public void onBackPressed() {
        MainActivity.this.finish();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dbHelper = new DBHelper(this);
        login = (Button) findViewById(R.id.btnLogin);
        toolbar = (Toolbar) findViewById(R.id.tool_main);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        });
        Reg = findViewById(R.id.btnSignUp);
        Reg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

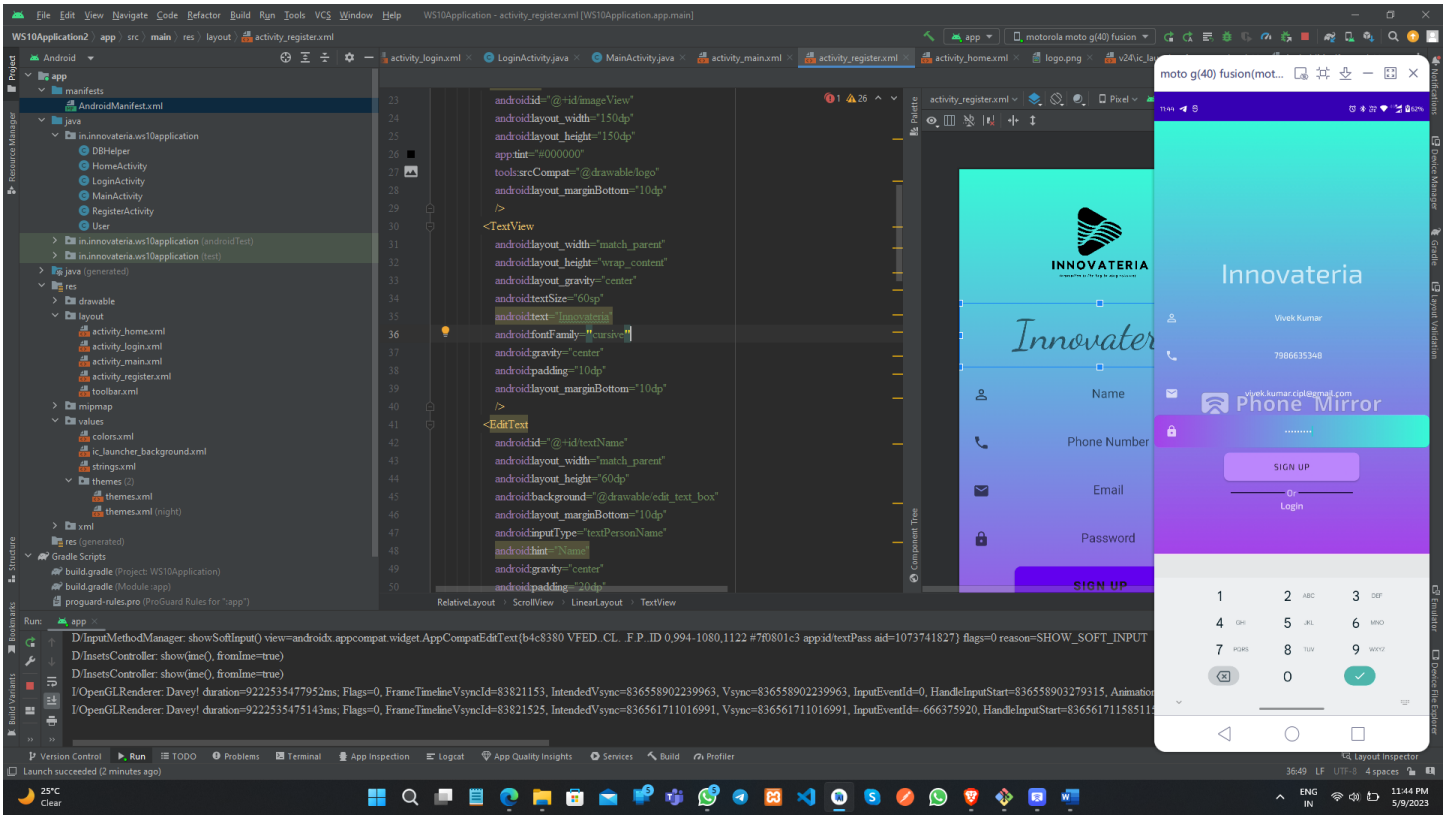
**DBHelper.class**

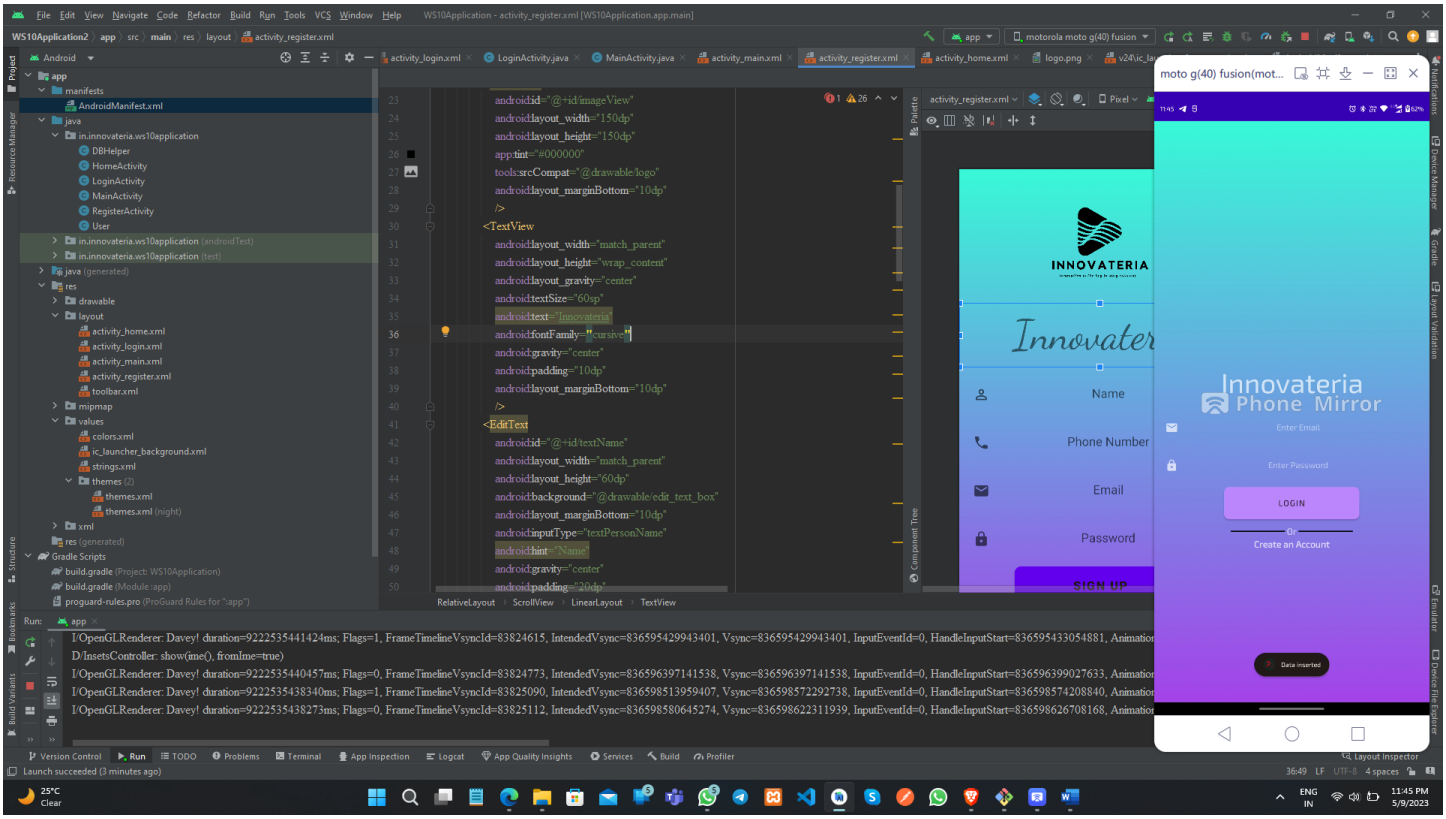
```
package in.innovateria.ws10application;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHelper extends SQLiteOpenHelper {
    public DBHelper(Context context ) {
        super(context,"UserData",null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase DB) {
        DB.execSQL("create Table UserDetails(userID TEXT primary key,name TEXT,password PASS-
WORD,number NUMBER)");
```

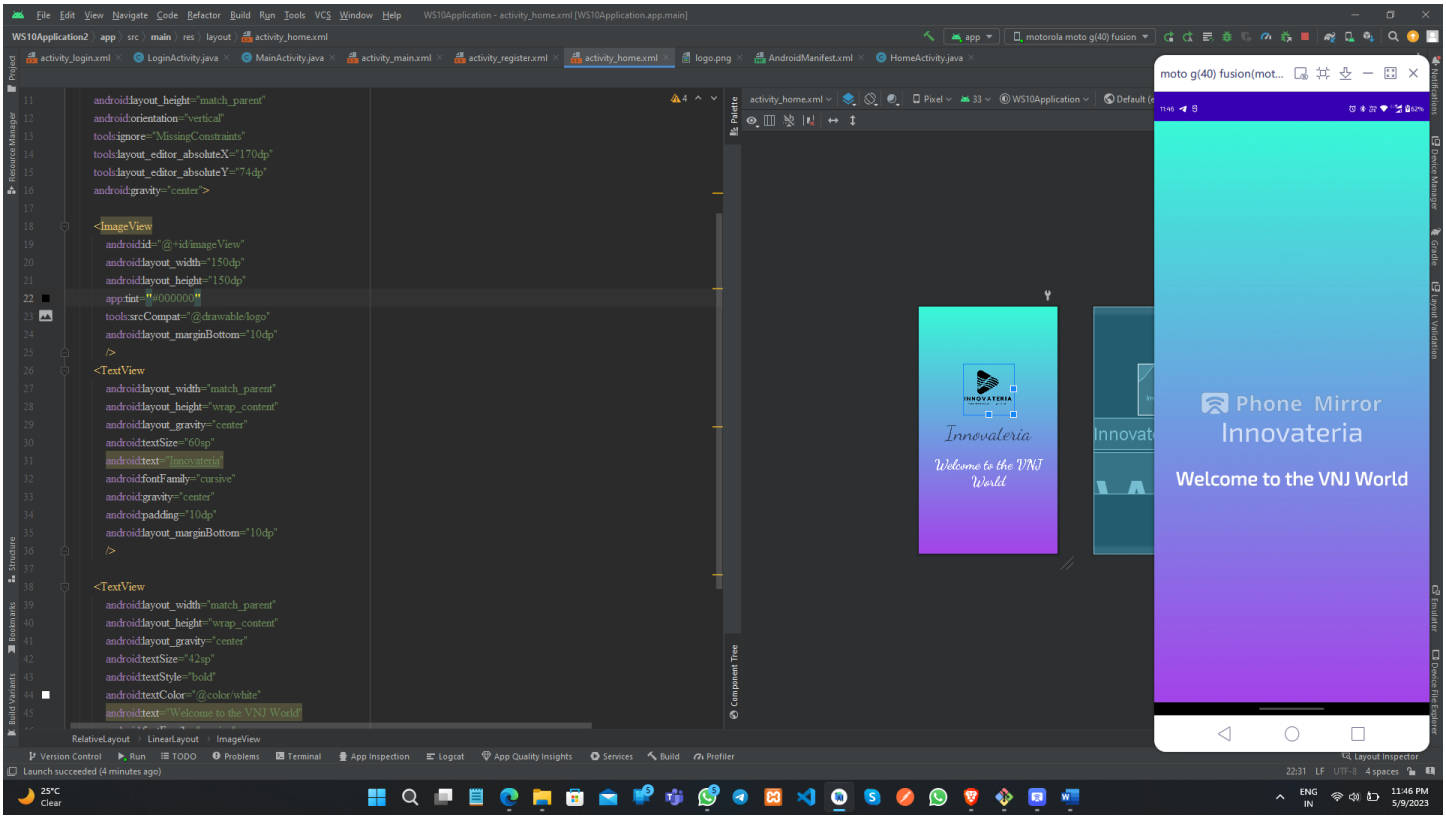
```
    }
    @Override
    public void onUpgrade(SQLiteDatabase DB, int i, int i1) {
        DB.execSQL("drop Table if exists UserDetails");
    }
    public Boolean insetUserData(String name,String number,String email,String password){
        SQLiteDatabase DB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("userID",email);
        contentValues.put("name",name);
        contentValues.put("password",password);
        contentValues.put("number",number);
        long result= DB.insert("UserDetails",null,contentValues);
        if (result == -1){
            return false;
        }else {
            return true;
        }
    }
    public Cursor getData(){
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor cursor = DB.rawQuery("Select * from Userdetails ",null);
        return cursor;
    }
}
```

## 7. Result/Output:




Innovateria
Name
Phone Number
Email
Password
SIGN UP
Vivek Kumar
7986635348
Phone Mirror
SIGN UP
Or
Login


Innovateria
Name
Phone Number
Email
Password
SIGN UP
Innovateria
Phone Mirror
Enter Email
Enter Password
LOGIN
Or
Create an Account
Data inserted




**Learning outcomes (What I have learnt):**

- To design an android application which uses SQLite Database and Create and Login & Register Application.
- Learnt about running application on android studio.
- Creating Application by Implementing SQLite Database Completed the Execution of the Experiment in the Lab.