MainActivity.java package com.example.ex

```
package com.example.exp4;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity:
public class MainActivity extends AppCompatActivity {
  private EditText nameInput, rollNoInput, depInput,
collegeInput, ageInput, locationInput, yearInput,
dsHostelInput;
  private Button addButton, readButton, updateButton,
deleteButton;
  private TextView resultTextView:
  private DatabaseManager dbManager;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
     setContentView(R.layout.activity main);
    // Initialize UI components
    nameInput = findViewById(R.id.name_input);
    rollNoInput = findViewById(R.id.roll no input);
     depInput = findViewById(R.id.dep input);
     collegeInput = findViewById(R.id.college input);
    ageInput = findViewById(R.id.age input);
     locationInput = findViewById(R.id.location input);
     yearInput = findViewById(R.id.year input);
     dsHostelInput = findViewById(R.id.ds hostel input);
    addButton = findViewById(R.id.add button);
    readButton = findViewById(R.id.read button);
     updateButton = findViewById(R.id.update button);
     deleteButton = findViewByld(R.id.delete button);
     resultTextView = findViewByld(R.id.result_text_view);
    dbManager = new DatabaseManager(this);
    // Insert example data if the database is empty
     insertExampleData();
    addButton.setOnClickListener(new
View.OnClickListener() {
       @Override
       public void onClick(View v) {
         addUser();
       }
    });
    readButton.setOnClickListener(new
View.OnClickListener() {
       @Override
```

```
public void onClick(View v) {
          readUser();
       }
    });
    updateButton.setOnClickListener(new
View.OnClickListener() {
       @Override
       public void onClick(View v) {
          updateUser();
       }
    });
    deleteButton.setOnClickListener(new
View.OnClickListener() {
       @Override
       public void onClick(View v) {
          deleteUser();
       }
    });
  }
  private void insertExampleData() {
     // Check if the database is empty before inserting
    if (dbManager.isDatabaseEmpty()) {
       dbManager.addUser("Sania", "221501", "AIML",
"REC", "19", "Santhome", "3rd", "Dayscholar");
       dbManager.addUser("Riya", "221502", "AIML",
"REC", "19", "Mylapore", "3rd", "Dayscholar");
       dbManager.addUser("Nandhini", "221503", "AIML",
"REC", "20", "Adyar", "3rd", "Hostel");
       dbManager.addUser("Mithra", "221504", "AIML",
"REC", "20", "T Nagar", "3rd", "Dayscholar");
       dbManager.addUser("Varsha", "221505", "AIML",
"REC", "19", "Nungambakkam", "3rd", "Hostel");
  }
  private void addUser() {
     String name = nameInput.getText().toString().trim();
    String rollNo = rollNoInput.getText().toString().trim();
    String dep = depInput.getText().toString().trim();
    String college = collegeInput.getText().toString().trim();
     String age = ageInput.getText().toString().trim();
    String location = locationInput.getText().toString().trim();
    String year = yearInput.getText().toString().trim();
     String dsHostel =
dsHostelInput.getText().toString().trim();
     if (name.isEmpty() || rollNo.isEmpty() || dep.isEmpty() ||
college.isEmpty() || age.isEmpty() ||
          location.isEmpty() || year.isEmpty() ||
dsHostel.isEmpty()) {
       Toast.makeText(MainActivity.this, "All fields are
required to add a user", Toast.LENGTH SHORT).show();
       return:
    }
     dbManager.addUser(name, rollNo, dep, college, age,
location, year, dsHostel);
```

```
Toast.makeText(MainActivity.this, "User added
                                                                                 location.isEmpty() || year.isEmpty() ||
successfully", Toast.LENGTH SHORT).show();
                                                                       dsHostel.isEmpty()) {
                                                                              Toast.makeText(MainActivity.this, "All fields are
                                                                       required to update a user", Toast.LENGTH SHORT).show();
  private void readUser() {
     String name = nameInput.getText().toString().trim();
                                                                            }
     String rollNo = rollNoInput.getText().toString().trim();
                                                                            dbManager.updateUser(name, rollNo, dep, college,
                                                                       age, location, year, dsHostel);
     if (name.isEmpty() || rollNo.isEmpty()) {
       Toast.makeText(MainActivity.this, "Name and Roll No
                                                                            Toast.makeText(MainActivity.this, "User updated
are required to read user details",
                                                                       successfully", Toast.LENGTH SHORT).show();
Toast.LENGTH SHORT).show();
                                                                         }
       return;
    }
                                                                         private void deleteUser() {
                                                                            String name = nameInput.getText().toString().trim();
     Cursor cursor = dbManager.getUserDetails(name,
                                                                            String rollNo = rollNoInput.getText().toString().trim();
rollNo);
                                                                            if (name.isEmpty() || rollNo.isEmpty()) {
     if (cursor != null && cursor.moveToFirst()) {
                                                                               Toast.makeText(MainActivity.this, "Name and Roll No
       String dep =
                                                                       are required to delete a user".
cursor.getString(cursor.getColumnIndex("Dep"));
                                                                       Toast.LENGTH SHORT).show();
       String college =
                                                                              return;
cursor.getString(cursor.getColumnIndex("College"));
                                                                            }
       String age =
cursor.getString(cursor.getColumnIndex("Age"));
                                                                            dbManager.deleteUser(name, rollNo);
       String location =
                                                                            Toast.makeText(MainActivity.this, "User deleted
cursor.getString(cursor.getColumnIndex("Location"));
                                                                       successfully", Toast.LENGTH_SHORT).show();
       String year =
cursor.getString(cursor.getColumnIndex("Year"));
                                                                       }
       String dsHostel =
cursor.getString(cursor.getColumnIndex("Ds Hostel"));
                                                                       DatabaseHelper.java
       resultTextView.setText("Name: " + name + "\nRoll No:
                                                                       package com.example.exp4;
" + rollNo +
            "\nDepartment: " + dep +
                                                                       import android.content.ContentValues;
            "\nCollege: " + college +
                                                                       import android.content.Context;
            "\nAge: " + age +
                                                                       import android.database.Cursor;
            "\nLocation: " + location +
                                                                       import android.database.sqlite.SQLiteDatabase;
            "\nYear: " + year +
                                                                       import android.database.sqlite.SQLiteOpenHelper;
            "\nDS/Hostel: " + dsHostel);
       cursor.close();
                                                                       public class DatabaseHelper extends SQLiteOpenHelper {
    } else {
       resultTextView.setText("No user found with Name: " +
                                                                         private static final int DATABASE_VERSION = 1;
name + " and Roll No: " + rollNo);
                                                                         private static final String DATABASE_NAME =
                                                                       "usersManager";
  }
                                                                         private static final String TABLE_USERS = "users";
  private void updateUser() {
                                                                         // Column names
     String name = nameInput.getText().toString().trim();
                                                                         private static final String KEY ID = "id";
     String rollNo = rollNoInput.getText().toString().trim();
                                                                         private static final String KEY NAME = "name";
     String dep = depInput.getText().toString().trim();
                                                                         private static final String KEY ROLL NO = "roll no";
     String college = collegeInput.getText().toString().trim();
                                                                         private static final String KEY_DEP = "dep";
     String age = ageInput.getText().toString().trim();
                                                                         private static final String KEY_COLLEGE = "college";
     String location = locationInput.getText().toString().trim();
                                                                         private static final String KEY_AGE = "age";
     String year = yearInput.getText().toString().trim();
                                                                         private static final String KEY_LOCATION = "location";
     String dsHostel =
                                                                         private static final String KEY_YEAR = "year";
dsHostelInput.getText().toString().trim();
                                                                         private static final String KEY_DS_HOSTEL = "ds_hostel";
     if (name.isEmpty() || rollNo.isEmpty() || dep.isEmpty() ||
                                                                         public DatabaseHelper(Context context) {
college.isEmpty() || age.isEmpty() ||
```

```
public void updateUser(String name, String rollNo, String
    super(context, DATABASE NAME, null,
DATABASE VERSION);
                                                                  dep, String college, String age, String location, String year,
                                                                  String dsHostel) {
                                                                       SQLiteDatabase db = this.getWritableDatabase();
  @Override
  public void onCreate(SQLiteDatabase db) {
                                                                       ContentValues values = new ContentValues();
    String CREATE USERS TABLE = "CREATE TABLE"
                                                                       values.put(KEY DEP, dep);
+ TABLE USERS + "("
                                                                       values.put(KEY_COLLEGE, college);
         + KEY ID + " INTEGER PRIMARY KEY,"
                                                                       values.put(KEY AGE, age);
         + KEY NAME + "TEXT,"
                                                                       values.put(KEY LOCATION, location);
                                                                       values.put(KEY YEAR, year);
         + KEY ROLL NO + "TEXT,"
         + KEY_DEP + " TEXT,"
                                                                       values.put(KEY DS HOSTEL, dsHostel);
         + KEY_COLLEGE + " TEXT,"
                                                                       db.update(TABLE_USERS, values, KEY_NAME + "=?
         + KEY_AGE + " TEXT,"
         + KEY_LOCATION + " TEXT,"
                                                                  AND " + KEY ROLL NO + "=?",
                                                                            new String[]{name, rollNo});
         + KEY_YEAR + " TEXT,"
         + KEY_DS_HOSTEL + " TEXT" + ")";
                                                                       db.close(); // Closing database connection
    db.execSQL(CREATE USERS TABLE);
                                                                    }
  }
                                                                     // Method to delete a user
  @Override
                                                                     public void deleteUser(String name, String rollNo) {
  public void on Upgrade (SQLite Database db, int old Version,
                                                                       SQLiteDatabase db = this.getWritableDatabase();
                                                                       db.delete(TABLE_USERS, KEY_NAME + "=? AND " +
int newVersion) {
    // Drop older table if existed
                                                                  KEY_ROLL_NO + "=?", new String[]{name, rollNo});
    db.execSQL("DROP TABLE IF EXISTS " +
                                                                       db.close();
TABLE_USERS);
                                                                    }
    // Create tables again
    onCreate(db);
                                                                     // Method to check if the database is empty
  }
                                                                     public boolean isDatabaseEmpty() {
                                                                       SQLiteDatabase db = this.getReadableDatabase();
  // Method to add a new user
                                                                       Cursor cursor = db.rawQuery("SELECT COUNT(*)
  public void addUser(String name, String rollNo, String
                                                                  FROM " + TABLE_USERS, null);
dep, String college, String age, String location, String year,
                                                                       boolean isEmpty = true;
String dsHostel) {
    SQLiteDatabase db = this.getWritableDatabase();
                                                                       if (cursor != null) {
                                                                         cursor.moveToFirst();
                                                                         isEmpty = cursor.getInt(0) == 0;
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, name);
                                                                         cursor.close();
    values.put(KEY_ROLL_NO, rollNo);
    values.put(KEY_DEP, dep);
                                                                       db.close();
    values.put(KEY_COLLEGE, college);
    values.put(KEY_AGE, age);
                                                                       return isEmpty;
    values.put(KEY_LOCATION, location);
    values.put(KEY_YEAR, year);
                                                                  }
    values.put(KEY_DS_HOSTEL, dsHostel);
                                                                  DatabaseManager.java
    db.insert(TABLE_USERS, null, values);
    db.close(); // Closing database connection
                                                                  package com.example.exp4;
                                                                  import android.content.ContentValues;
  // Method to get user details by name and roll number
                                                                  import android.content.Context;
  public Cursor getUserDetails(String name, String rollNo) {
                                                                  import android.database.Cursor;
    SQLiteDatabase db = this.getReadableDatabase();
                                                                  import android.database.sqlite.SQLiteDatabase;
    return db.query(TABLE_USERS, null, KEY_NAME +
                                                                  import android.database.sqlite.SQLiteOpenHelper;
"=? AND " + KEY_ROLL_NO + "=?",
         new String[]{name, rollNo}, null, null, null);
                                                                  public class DatabaseManager extends SQLiteOpenHelper {
                                                                     private static final String DATABASE NAME =
  // Method to update user details
                                                                  "StudentDB";
                                                                     private static final int DATABASE VERSION = 1;
```

```
private static final String TABLE USERS = "users";
                                                                       SQLiteDatabase db = this.getReadableDatabase();
                                                                       return db.query(TABLE USERS, null, KEY NAME +
  // Column names
                                                                  "=? AND " + KEY ROLL NO + "=?",
  private static final String KEY NAME = "Name";
                                                                           new String[]{name, rollNo}, null, null, null);
  private static final String KEY ROLL NO = "Roll no";
                                                                    }
  private static final String KEY DEP = "Dep";
  private static final String KEY COLLEGE = "College";
                                                                    public void updateUser(String name, String rollNo, String
  private static final String KEY AGE = "Age";
                                                                  dep. String college, String age, String location, String year,
  private static final String KEY LOCATION = "Location";
                                                                  String dsHostel) {
  private static final String KEY YEAR = "Year";
                                                                       SQLiteDatabase db = this.getWritableDatabase();
  private static final String KEY_DS_HOSTEL =
"Ds Hostel";
                                                                       ContentValues values = new ContentValues();
                                                                       values.put(KEY_DEP, dep);
                                                                       values.put(KEY_COLLEGE, college);
  public DatabaseManager(Context context) {
    super(context, DATABASE NAME, null,
                                                                       values.put(KEY AGE, age);
DATABASE_VERSION);
                                                                       values.put(KEY LOCATION, location);
                                                                       values.put(KEY_YEAR, year);
  }
                                                                       values.put(KEY DS HOSTEL, dsHostel);
  @Override
  public void onCreate(SQLiteDatabase db) {
                                                                       db.update(TABLE_USERS, values, KEY_NAME + "=?
    String CREATE USERS TABLE = "CREATE TABLE"
                                                                  AND " + KEY ROLL NO + "=?".
+ TABLE_USERS + "("
                                                                           new String[]{name, rollNo});
         + KEY_NAME + " TEXT,"
                                                                       db.close();
         + KEY_ROLL_NO + " TEXT PRIMARY KEY,"
                                                                    }
         + KEY_DEP + " TEXT,"
         + KEY_COLLEGE + " TEXT,"
                                                                    public void deleteUser(String name, String rollNo) {
         + KEY_AGE + " TEXT,"
                                                                       SQLiteDatabase db = this.getWritableDatabase();
                                                                       db.delete(TABLE USERS, KEY NAME + "=? AND " +
         + KEY LOCATION + " TEXT,"
         + KEY YEAR + "TEXT,"
                                                                  KEY ROLL NO + "=?", new String[]{name, rollNo});
         + KEY_DS_HOSTEL + " TEXT" + ")";
                                                                       db.close();
    db.execSQL(CREATE USERS TABLE);
                                                                    }
  }
                                                                    public boolean isDatabaseEmpty() {
  @Override
                                                                       SQLiteDatabase db = this.getReadableDatabase();
  public void onUpgrade(SQLiteDatabase db, int oldVersion,
                                                                       Cursor cursor = db.rawQuery("SELECT COUNT(*)
int newVersion) {
                                                                  FROM " + TABLE_USERS, null);
    db.execSQL("DROP TABLE IF EXISTS " +
                                                                       boolean isEmpty = true;
TABLE_USERS);
    onCreate(db);
                                                                       if (cursor != null) {
                                                                         cursor.moveToFirst();
  }
                                                                         isEmpty = cursor.getInt(0) == 0;
  public void addUser(String name, String rollNo, String
                                                                         cursor.close();
dep, String college, String age, String location, String year,
String dsHostel) {
    SQLiteDatabase db = this.getWritableDatabase();
                                                                       db.close();
                                                                       return isEmpty;
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, name);
                                                                  }
    values.put(KEY ROLL NO, rollNo);
    values.put(KEY_DEP, dep);
                                                                  AndroidManifest.xml
    values.put(KEY_COLLEGE, college);
    values.put(KEY_AGE, age);
                                                                  <?xml version="1.0" encoding="utf-8"?>
    values.put(KEY_LOCATION, location);
    values.put(KEY_YEAR, year);
                                                                  xmlns:android="http://schemas.android.com/apk/res/android"
    values.put(KEY_DS_HOSTEL, dsHostel);
                                                                    xmlns:tools="http://schemas.android.com/tools">
                                                                     <uses-permission
    db.insert(TABLE_USERS, null, values);
                                                                  android:name="android.permission.INTERNET"/>
    db.close();
                                                                     <uses-permission
                                                                  android:name="android.permission.WRITE EXTERNAL ST
                                                                  ORAGE"/>
  public Cursor getUserDetails(String name, String rollNo) {
```

```
<application
    android:allowBackup="true"
                                                                         android:hint="College" />
android:dataExtractionRules="@xml/data extraction rules"
    android:fullBackupContent="@xml/backup rules"
                                                                       <EditText
     android:icon="@mipmap/ic launcher"
                                                                         android:id="@+id/age input"
    android:label="@string/app name"
                                                                         android:layout width="match parent"
    android:roundlcon="@mipmap/ic_launcher_round"
                                                                         android:layout height="wrap content"
    android:supportsRtl="true"
                                                                         android:hint="Age" />
    android:theme="@style/Theme.Exp4"
    tools:targetApi="31">
                                                                       <EditText
                                                                         android:id="@+id/location input"
     <activity
                                                                         android:layout width="match parent"
       android:name=".MainActivity"
       android:exported="true">
                                                                         android:layout height="wrap content"
       <intent-filter>
                                                                         android:hint="Location" />
         <action
android:name="android.intent.action.MAIN" />
                                                                       <EditText
                                                                         android:id="@+id/vear input"
                                                                         android:layout width="match parent"
         <category
android:name="android.intent.category.LAUNCHER" />
                                                                         android:layout height="wrap content"
                                                                         android:hint="Year" />
       </intent-filter>
     </activity>
                                                                       <EditText
                                                                         android:id="@+id/ds_hostel_input"
  </application>
                                                                         android:layout_width="match_parent"
                                                                         android:layout height="wrap content"
</manifest>
                                                                         android:hint="DS/Hostel" />
Activity main.xml
                                                                       <Button
                                                                         android:id="@+id/add button"
<?xml version="1.0" encoding="utf-8"?>
                                                                         android:layout width="match parent"
<LinearLavout
                                                                         android:layout height="wrap content"
xmlns:android="http://schemas.android.com/apk/res/android"
                                                                         android:text="Add User" />
  android:layout width="match parent"
  android:layout_height="match_parent"
                                                                       <Button
  android:orientation="vertical"
                                                                         android:id="@+id/read button"
  android:padding="16dp">
                                                                         android:layout width="match parent"
                                                                         android:layout height="wrap content"
  <EditText
                                                                         android:text="Read User" />
    android:id="@+id/name input"
    android:layout_width="match_parent"
                                                                       <Button
    android:layout_height="wrap_content"
                                                                         android:id="@+id/update_button"
    android:hint="Name" />
                                                                         android:layout width="match parent"
                                                                         android:layout height="wrap content"
  <EditText
                                                                         android:text="Update User" />
    android:id="@+id/roll no input"
    android:layout width="match parent"
                                                                       <Button
    android:layout_height="wrap_content"
                                                                         android:id="@+id/delete button"
    android:hint="Roll No" />
                                                                         android:layout_width="match_parent"
                                                                         android:layout_height="wrap_content"
  <EditText
                                                                         android:text="Delete User" />
    android:id="@+id/dep_input"
    android:layout_width="match_parent"
                                                                       <TextView
    android:layout_height="wrap_content"
                                                                         android:id="@+id/result_text_view"
    android:hint="Department" />
                                                                         android:layout width="match parent"
                                                                         android:layout height="wrap content"
  <FditText
                                                                         android:paddingTop="16dp"
    android:id="@+id/college_input"
                                                                         android:text="Results will be shown here" />
```

</LinearLayout>

android:layout width="match parent"

android:layout_height="wrap_content"

