MainActivity.java

```
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 = findViewById(R.id.delete_button);
        resultTextView = findViewById(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
successfully", Toast.LENGTH_SHORT).show();
    }

    private void readUser() {
        String name = nameInput.getText().toString().trim();
        String rollNo = rollNoInput.getText().toString().trim();

        if (name.isEmpty() || rollNo.isEmpty()) {
            Toast.makeText(MainActivity.this, "Name and Roll No
are required to read user details",
Toast.LENGTH_SHORT).show();
            return;
        }

        Cursor cursor = dbManager.getUserDetails(name,
rollNo);

        if (cursor != null && cursor.moveToFirst()) {
            String dep =
cursor.getString(cursor.getColumnIndex("Dep"));
            String college =
cursor.getString(cursor.getColumnIndex("College"));
            String age =
cursor.getString(cursor.getColumnIndex("Age"));
            String location =
cursor.getString(cursor.getColumnIndex("Location"));
            String year =
cursor.getString(cursor.getColumnIndex("Year"));
            String dsHostel =
cursor.getString(cursor.getColumnIndex("Ds_Hostel"));

            resultTextView.setText("Name: " + name + "\nRoll No:
" + rollNo +
                    "\nDepartment: " + dep +
                    "\nCollege: " + college +
                    "\nAge: " + age +
                    "\nLocation: " + location +
                    "\nYear: " + year +
                    "\nDS/Hostel: " + dsHostel);
            cursor.close();
        } else {
            resultTextView.setText("No user found with Name: " +
name + " and Roll No: " + rollNo);
        }
    }

    private void updateUser() {
        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 update a user", Toast.LENGTH_SHORT).show();
            return;
        }

        dbManager.updateUser(name, rollNo, dep, college,
age, location, year, dsHostel);
        Toast.makeText(MainActivity.this, "User updated
successfully", Toast.LENGTH_SHORT).show();
    }

    private void deleteUser() {
        String name = nameInput.getText().toString().trim();
        String rollNo = rollNoInput.getText().toString().trim();

        if (name.isEmpty() || rollNo.isEmpty()) {
            Toast.makeText(MainActivity.this, "Name and Roll No
are required to delete a user",
Toast.LENGTH_SHORT).show();
            return;
        }

        dbManager.deleteUser(name, rollNo);
        Toast.makeText(MainActivity.this, "User deleted
successfully", Toast.LENGTH_SHORT).show();
    }
}
```

DatabaseHelper.java

```
package com.example.exp4;

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

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME =
"usersManager";
    private static final String TABLE_USERS = "users";

    // Column names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "name";
    private static final String KEY_ROLL_NO = "roll_no";
    private static final String KEY_DEP = "dep";
    private static final String KEY_COLLEGE = "college";
    private static final String KEY_AGE = "age";
    private static final String KEY_LOCATION = "location";
    private static final String KEY_YEAR = "year";
    private static final String KEY_DS_HOSTEL = "ds_hostel";

    public DatabaseHelper(Context context) {
```

```
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_USERS_TABLE = "CREATE TABLE "
+ TABLE_USERS + "("
                + KEY_ID + " INTEGER PRIMARY KEY,"
                + KEY_NAME + " TEXT,"
                + KEY_ROLL_NO + " TEXT,"
                + KEY_DEP + " TEXT,"
                + KEY_COLLEGE + " TEXT,"
                + KEY_AGE + " TEXT,"
                + KEY_LOCATION + " TEXT,"
                + KEY_YEAR + " TEXT,"
                + KEY_DS_HOSTEL + " TEXT" + ")";
        db.execSQL(CREATE_USERS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " +
TABLE_USERS);
        // Create tables again
        onCreate(db);
    }

    // Method to add a new user
    public void addUser(String name, String rollNo, String
dep, String college, String age, String location, String year,
String dsHostel) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_NAME, name);
        values.put(KEY_ROLL_NO, rollNo);
        values.put(KEY_DEP, dep);
        values.put(KEY_COLLEGE, college);
        values.put(KEY_AGE, age);
        values.put(KEY_LOCATION, location);
        values.put(KEY_YEAR, year);
        values.put(KEY_DS_HOSTEL, dsHostel);

        db.insert(TABLE_USERS, null, values);
        db.close(); // Closing database connection
    }

    // Method to get user details by name and roll number
    public Cursor getUserDetails(String name, String rollNo) {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.query(TABLE_USERS, null, KEY_NAME +
"=? AND " + KEY_ROLL_NO + "=?",
                new String[]{name, rollNo}, null, null, null);
    }

    // Method to update user details
    public void updateUser(String name, String rollNo, String
dep, String college, String age, String location, String year,
String dsHostel) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_DEP, dep);
        values.put(KEY_COLLEGE, college);
        values.put(KEY_AGE, age);
        values.put(KEY_LOCATION, location);
        values.put(KEY_YEAR, year);
        values.put(KEY_DS_HOSTEL, dsHostel);

        db.update(TABLE_USERS, values, KEY_NAME + "=?
AND " + KEY_ROLL_NO + "=?",
                new String[]{name, rollNo});
        db.close(); // Closing database connection
    }

    // Method to delete a user
    public void deleteUser(String name, String rollNo) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_USERS, KEY_NAME + "=? AND " +
KEY_ROLL_NO + "=?", new String[]{name, rollNo});
        db.close();
    }

    // Method to check if the database is empty
    public boolean isDatabaseEmpty() {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT COUNT(*)
FROM " + TABLE_USERS, null);
        boolean isEmpty = true;

        if (cursor != null) {
            cursor.moveToFirst();
            isEmpty = cursor.getInt(0) == 0;
            cursor.close();
        }

        db.close();
        return isEmpty;
    }
}
```

DatabaseManager.java

```
package com.example.exp4;

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

public class DatabaseManager extends SQLiteOpenHelper {

    private static final String DATABASE_NAME =
"StudentDB";
    private static final int DATABASE_VERSION = 1;
```

```
    private static final String TABLE_USERS = "users";

    // Column names
    private static final String KEY_NAME = "Name";
    private static final String KEY_ROLL_NO = "Roll_no";
    private static final String KEY_DEP = "Dep";
    private static final String KEY_COLLEGE = "College";
    private static final String KEY_AGE = "Age";
    private static final String KEY_LOCATION = "Location";
    private static final String KEY_YEAR = "Year";
    private static final String KEY_DS_HOSTEL =
"Ds_Hostel";

    public DatabaseManager(Context context) {
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_USERS_TABLE = "CREATE TABLE "
+ TABLE_USERS + "("
                + KEY_NAME + " TEXT,"
                + KEY_ROLL_NO + " TEXT PRIMARY KEY,"
                + KEY_DEP + " TEXT,"
                + KEY_COLLEGE + " TEXT,"
                + KEY_AGE + " TEXT,"
                + KEY_LOCATION + " TEXT,"
                + KEY_YEAR + " TEXT,"
                + KEY_DS_HOSTEL + " TEXT" + ")";
        db.execSQL(CREATE_USERS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " +
TABLE_USERS);
        onCreate(db);
    }

    public void addUser(String name, String rollNo, String
dep, String college, String age, String location, String year,
String dsHostel) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_NAME, name);
        values.put(KEY_ROLL_NO, rollNo);
        values.put(KEY_DEP, dep);
        values.put(KEY_COLLEGE, college);
        values.put(KEY_AGE, age);
        values.put(KEY_LOCATION, location);
        values.put(KEY_YEAR, year);
        values.put(KEY_DS_HOSTEL, dsHostel);

        db.insert(TABLE_USERS, null, values);
        db.close();
    }

    public Cursor getUserDetails(String name, String rollNo) {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.query(TABLE_USERS, null, KEY_NAME +
"=? AND " + KEY_ROLL_NO + "=?",
                new String[]{name, rollNo}, null, null, null);
    }

    public void updateUser(String name, String rollNo, String
dep, String college, String age, String location, String year,
String dsHostel) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_DEP, dep);
        values.put(KEY_COLLEGE, college);
        values.put(KEY_AGE, age);
        values.put(KEY_LOCATION, location);
        values.put(KEY_YEAR, year);
        values.put(KEY_DS_HOSTEL, dsHostel);

        db.update(TABLE_USERS, values, KEY_NAME + "=?
AND " + KEY_ROLL_NO + "=?",
                new String[]{name, rollNo});
        db.close();
    }

    public void deleteUser(String name, String rollNo) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_USERS, KEY_NAME + "=? AND " +
KEY_ROLL_NO + "=?", new String[]{name, rollNo});
        db.close();
    }

    public boolean isDatabaseEmpty() {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT COUNT(*)
FROM " + TABLE_USERS, null);
        boolean isEmpty = true;

        if (cursor != null) {
            cursor.moveToFirst();
            isEmpty = cursor.getInt(0) == 0;
            cursor.close();
        }

        db.close();
        return isEmpty;
    }
}
```

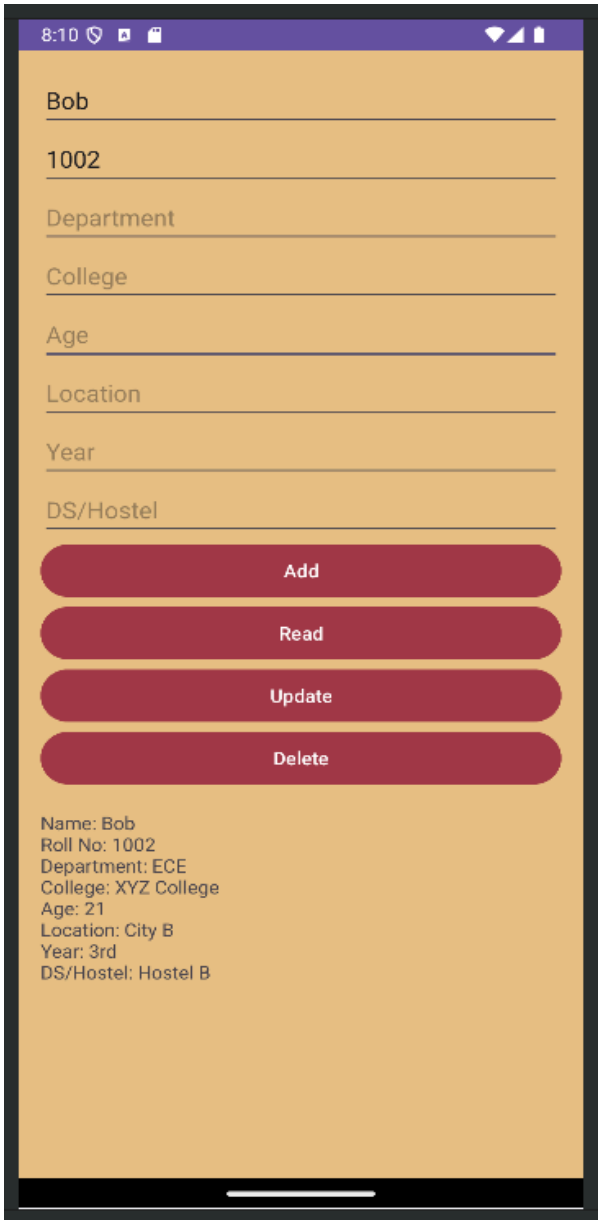
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission
android:name="android.permission.INTERNET"/>
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_ST
ORAGE"/>
```

```
    <application
        android:allowBackup="true"

android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Exp4"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />

            </intent-filter>
        </activity>
    </application>

</manifest>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/name_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name" />

    <EditText
        android:id="@+id/roll_no_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Roll No" />

    <EditText
        android:id="@+id/dep_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Department" />

    <EditText
        android:id="@+id/college_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="College" />

    <EditText
        android:id="@+id/age_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age" />

    <EditText
        android:id="@+id/location_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Location" />

    <EditText
        android:id="@+id/year_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Year" />

    <EditText
        android:id="@+id/ds_hostel_input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="DS/Hostel" />

    <Button
        android:id="@+id/add_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add User" />

    <Button
        android:id="@+id/read_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read User" />

    <Button
        android:id="@+id/update_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update User" />

    <Button
        android:id="@+id/delete_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete User" />

    <TextView
        android:id="@+id/result_text_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="16dp"
        android:text="Results will be shown here" />
</LinearLayout>
```


8:10
Bob
1002
Department
College
Age
Location
Year
DS/Hostel
Add
Read
Update
Delete
Name: Bob
Roll No: 1002
Department: ECE
College: XYZ College
Age: 21
Location: City B
Year: 3rd
DS/Hostel: Hostel B