PRACTICAL-7
Implementation of SQLite Database

1.Complete CRUD operation in given application.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/txtID"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter ID"
        android:ems="10"/>
    <TextView
        android:id="@+id/fstTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name" />
    <EditText
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"/>
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Designation" />
    <EditText
        android:id="@+id/txtDesignation"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
<TextView
    android:id="@+id/thirdTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Location" />
<EditText
    android:id="@+id/txtLocation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10" />
<Button
    android:id="@+id/btnAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add" />
<Button
    android:id="@+id/btnUpdate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Update" />
<Button
    android:id="@+id/btnDelete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Delete" />
<Button
    android:id="@+id/btnDisplay"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Display" />
</LinearLayout>
```

details.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">
  <ListView
    android:id="@+id/user_list"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:dividerHeight="1dp" />
  <Button
    android:id="@+id/btnBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:text="Back" />
</LinearLayout>
```

PRACTICAL-7
Implementation of SQLite Database

list_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dip">
    <TextView
        android:id="@+id/id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="17sp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginEnd="16dp"
        android:layout_marginTop="16dp"
        android:text="ID"/>
    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="17sp"
        android:layout_below="@+id/id"
        android:layout_alignParentStart="true"
        android:text="Name"/>
    <TextView
        android:id="@+id/designation"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:layout_below="@+id/name"
        android:layout_alignParentStart="true"
        android:layout_marginTop="7dp"
        android:text="Designation"/>
    <TextView
        android:id="@+id/location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:layout_below="@+id/designation"
        android:layout_alignParentStart="true"
        android:layout_marginTop="7dp"
        android:text="Location"/>
</RelativeLayout>
```

DbHandler.java

```
package com.example.program1;


import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;
import java.util.HashMap;
public class DbHandler extends SQLiteOpenHelper {
    private static final int DB_VERSION = 1;
```

```
private static final String DB_NAME = "usersdb";
private static final String TABLE_Users = "userdetails";
private static final String KEY_ID = "id";
private static final String KEY_NAME = "name";
private static final String KEY_DESIGNATION = "designation";
private static final String KEY_LOCATION = "location";
public DbHandler(Context context) {
   super (context, DB_NAME, null, DB_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
   String CREATE_TABLE = "CREATE TABLE " + TABLE_Users + "("
         + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
         + KEY_NAME + " TEXT,"
         + KEY_DESIGNATION + " TEXT,"
         + KEY_LOCATION + " TEXT"
         + ")";
   db.execSQL(CREATE_TABLE);
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
   // Drop older table if exists
   db.execSQL("DROP TABLE IF EXISTS " + TABLE_Users);
   // Create tables again
   onCreate(db);
}
// Insert a new user
public long insertUserDetails(String name, String designation, String location) {
   SQLiteDatabase db = this.getWritableDatabase();
   ContentValues cValues = new ContentValues();
```

```
        cValues.put(KEY_NAME, name);
        cValues.put(KEY_DESIGNATION, designation);
        cValues.put(KEY_LOCATION, location);
        long newRowId = db.insert(TABLE_Users, null, cValues);
        db.close();
        return newRowId;
    }
    // Update user details
    public int updateUserDetails(int id, String name, String designation, String location) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cVals = new ContentValues();
        cVals.put(KEY_NAME, name);
        cVals.put(KEY_DESIGNATION, designation);
        cVals.put(KEY_LOCATION, location);
        return db.update(TABLE_Users, cVals, KEY_ID + " = ?", new
String[]{String.valueOf(id)});
    }
    // Delete a user by ID
    public void deleteUser(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_Users, KEY_ID + " = ?", new String[]{String.valueOf(id)});
        db.close();
    }
    // Get all users
    @SuppressLint("Range")
    public ArrayList<HashMap<String, String>> getAllUsers() {
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> userList = new ArrayList<>();
        String query = "SELECT * FROM " + TABLE_Users;
        Cursor cursor = db.rawQuery(query, null);
```

```
        while (cursor.moveToNext()) {
            HashMap<String, String> user = new HashMap<>();
            user.put("id", cursor.getString(cursor.getColumnIndex(KEY_ID)));
            user.put("name", cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            user.put("designation",
cursor.getString(cursor.getColumnIndex(KEY_DESIGNATION)));
            user.put("location", cursor.getString(cursor.getColumnIndex(KEY_LOCATION)));
            userList.add(user);
        }
        cursor.close();
        return userList;
    }
    // Get a user by ID
    @SuppressLint("Range")
    public HashMap<String, String> getUserById(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        HashMap<String, String> user = new HashMap<>();
        String query = "SELECT * FROM " + TABLE_Users + " WHERE " + KEY_ID + " =
?";
        Cursor cursor = db.rawQuery(query, new String[]{String.valueOf(id)});
        if (cursor.moveToNext()) {
            user.put("id", cursor.getString(cursor.getColumnIndex(KEY_ID)));
            user.put("name", cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            user.put("designation",
cursor.getString(cursor.getColumnIndex(KEY_DESIGNATION)));
            user.put("location", cursor.getString(cursor.getColumnIndex(KEY_LOCATION)));
        }
        cursor.close();
        return user;
    }
}
```

PRACTICAL-7
Implementation of SQLite Database

Details.java

```
package com.example.program1;


import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import java.util.HashMap;
public class Details extends AppCompatActivity {
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.details);
        DbHandler db = new DbHandler(this);
        ArrayList<HashMap<String, String>> userList = db.getAllUsers();
        ListView lv = findViewById(R.id.user_list);
        ListAdapter adapter = new SimpleAdapter(
                Details.this, userList, R.layout.list_row,
                new String[]{"id", "name", "designation", "location"},
                new int[]{R.id.id, R.id.name, R.id.designation, R.id.location}
        );
        lv.setAdapter(adapter);
        Button back = findViewById(R.id.btnBack);
        back.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            intent = new Intent(Details.this, MainActivity.class);
            startActivity(intent);
        }
    });
  }
}
```

MainActivity.java

```
package com.example.program1;


import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
  EditText id, name, designation, location;
  Button saveBtn, addBtn, updateBtn, deleteBtn, displayBtn;
  Intent intent;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
// Initialize EditText for entering the ID
    id = findViewById(R.id.txtID);
    name = findViewById(R.id.txtName);
    designation = findViewById(R.id.txtDesignation);
```

```
        location = findViewById(R.id.txtLocation);
        addBtn = findViewById(R.id.btnAdd);
        updateBtn = findViewById(R.id.btnUpdate);
        deleteBtn = findViewById(R.id.btnDelete);
        displayBtn = findViewById(R.id.btnDisplay);
// Click listener for the "Add" button
        addBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name1 = name.getText().toString();
                String designation1 = designation.getText().toString();
                String location1 = location.getText().toString();
                DbHandler dbHandler = new DbHandler(MainActivity.this);
                long newRowId = dbHandler.insertUserDetails(name1, designation1, location1);
                if (newRowId != -1) {
// Insertion successful
                    Toast.makeText(getApplicationContext(), "Details Added Successfully",
                            Toast.LENGTH_SHORT).show();
// Optionally, you can clear the input fields here
                    name.setText("");
                    designation.setText("");
                    location.setText("");
                } else {
// Insertion failed
                    Toast.makeText(getApplicationContext(), "Failed to add details",
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
// Click listener for the "Update" button
```

```
        updateBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String idStr = id.getText().toString();
                if (!idStr.isEmpty()) {
                    int userId = Integer.parseInt(idStr);
                    String name1 = name.getText().toString();
                    String designation1 = designation.getText().toString();
                    String location1 = location.getText().toString();
                    DbHandler dbHandler = new DbHandler(MainActivity.this);
                    int rowsUpdated = dbHandler.updateUserDetails(userId, name1, designation1,
                            location1);
                    if (rowsUpdated > 0) {
                        Toast.makeText(getApplicationContext(), "Details Updated Successfully",
                                Toast.LENGTH_SHORT).show();
// Optionally, you can clear the input fields here
                        id.setText("");
                        name.setText("");
                        designation.setText("");
                        location.setText("");
                    } else {
                        Toast.makeText(getApplicationContext(), "No user with this ID found",
                                Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(getApplicationContext(), "Please enter an ID to update",
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
```

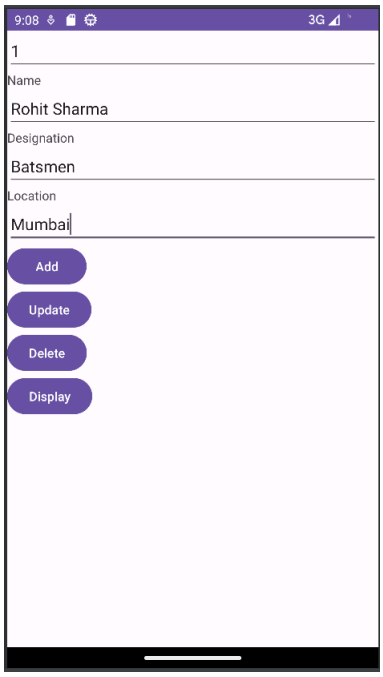
```
// Click listener for the "Delete" button
        deleteBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String idStr = id.getText().toString();
                if (!idStr.isEmpty()) {
                    int userId = Integer.parseInt(idStr);
                    DbHandler dbHandler = new DbHandler(MainActivity.this);
                    dbHandler.deleteUser(userId);
// Optionally, you can clear the input fields here
                    id.setText("");
                    name.setText("");
                    designation.setText("");
                    location.setText("");
                    Toast.makeText(getApplicationContext(),"User_Deleted_Successfully",
                            Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(getApplicationContext(), "Please enter an ID to delete",
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
// Click listener for the "Display" button
        displayBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                intent = new Intent(MainActivity.this, Details.class);
                startActivity(intent);
            }
        });
```

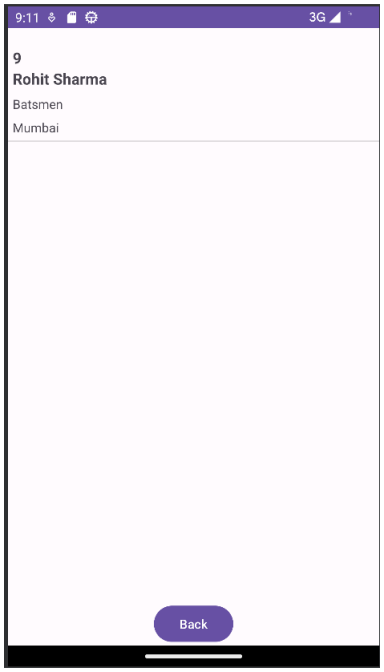
```
    }
}
```

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">

    <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.Program1"
        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>
        <activity android:name=".Details"></activity>
    </application>

</manifest>
```

OUTPUT





2. Create an application which will handle Student Details.

• Create a database with name “MyDb” •

Create a student table with id, rollno, name and marks.

• Create a screen to allow user to input student details, and store the details to table. Display error message if data is empty.

• Display all students entered. • Allow the user to edit or remove the student

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">
        <EditText
            android:id="@+id/txtRollNo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
    android:hint="Roll Number" />
<EditText
    android:id="@+id/txtName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name" />
<EditText
    android:id="@+id/txtCMarks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="C Marks" />
<EditText
    android:id="@+id/txtJavaMarks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Java Marks" />
<EditText
    android:id="@+id/txtPythonMarks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Python Marks" />
<Button
    android:id="@+id/btnAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add" />
<Button
    android:id="@+id/btnUpdate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="Update" />
    <Button
        android:id="@+id/btnDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete" />
    <Button
        android:id="@+id/btnDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Display" />
</LinearLayout>
```

details.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">
  <ListView
    android:id="@+id/studentListView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:dividerHeight="1dp" />
  <Button
    android:id="@+id/btnBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Back" />
```

```
</LinearLayout>
```

list_row.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="wrap_content"
    android:orientation="vertical"
    android:padding="5dip" >
    <TextView
        android:id="@+id/rollNo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="17dp" />
    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="17dp" />
    <TextView
        android:id="@+id/cMarks"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="7dp"
        android:textColor="#343434"
        android:textSize="14dp" />
    <TextView
        android:id="@+id/javaMarks"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:textColor="#343434"
        android:textSize="14dp" />
    <TextView
        android:id="@+id/pythonMarks"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#343434"
        android:textSize="14dp" />
</LinearLayout>
```

DbHandler.java

```
package com.example.student;
import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;
import java.util.HashMap;
public class DbHandler extends SQLiteOpenHelper {
    private static final int DB_VERSION = 1;
    private static final String DB_NAME = "MyDb";
    private static final String TABLE_Students = "studentdetails";
    private static final String KEY_ID = "id";
    private static final String KEY_ROLLNO = "rollno";
    private static final String KEY_NAME = "name";
```

```
    private static final String KEY_CMarks = "c_marks";
    private static final String KEY_JavaMarks = "java_marks";
    private static final String KEY_PythonMarks = "python_marks";


    public DbHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE = "CREATE TABLE " + TABLE_Students + "("
                + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
                + KEY_ROLLNO + " TEXT,"
                + KEY_NAME + " TEXT,"
                + KEY_CMarks + " INTEGER,"
                + KEY_JavaMarks + " INTEGER,"
                + KEY_PythonMarks + " INTEGER"
                + ")";
        db.execSQL(CREATE_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
// Drop older table if exists
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_Students);
// Create tables again
        onCreate(db);
    }
    // Adding new Student Details
    void insertStudentDetails(String rollNo, String name, int cMarks, int javaMarks, int
            pythonMarks) {
// Get the Data Repository in write mode
```

```
        SQLiteDatabase db = this.getWritableDatabase();
// Create a new map of values, where column names are the keys
        ContentValues cValues = new ContentValues();
        cValues.put(KEY_ROLLNO, rollNo);
        cValues.put(KEY_NAME, name);
        cValues.put(KEY_CMarks, cMarks);
        cValues.put(KEY_JavaMarks, javaMarks);
        cValues.put(KEY_PythonMarks, pythonMarks);
// Insert the new row, returning the primary key value of the new row
        long newRowId = db.insert(TABLE_Students, null, cValues);
        db.close();
    }
    @SuppressLint("Range")
    public ArrayList<HashMap<String, String>> GetStudents() {
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> studentList = new ArrayList<>();
        String query = "SELECT rollno, name, c_marks, java_marks, python_marks, id FROM
" +
                TABLE_Students;
        Cursor cursor = db.rawQuery(query, null);
        while (cursor.moveToNext()) {
            HashMap<String, String> student = new HashMap<>();
            student.put("rollNo", cursor.getString(cursor.getColumnIndex(KEY_ROLLNO)));
            student.put("name", cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            student.put("cMarks", cursor.getString(cursor.getColumnIndex(KEY_CMarks)));
            student.put("javaMarks",
cursor.getString(cursor.getColumnIndex(KEY_JavaMarks)));
            student.put("pythonMarks",
                    cursor.getString(cursor.getColumnIndex(KEY_PythonMarks)));
            student.put("id", cursor.getString(cursor.getColumnIndex(KEY_ID)));
```

```
            studentList.add(student);
        }
        return studentList;
    }
    @SuppressLint("Range")
    public ArrayList<HashMap<String, String>> GetStudentById(int studentId) {
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> studentList = new ArrayList<>();
        String query = "SELECT rollno, name, c_marks, java_marks, python_marks FROM " +
                TABLE_Students;
        Cursor cursor = db.query(TABLE_Students, new String[]{KEY_ROLLNO,
KEY_NAME,
                        KEY_CMarks,
                        KEY_JavaMarks, KEY_PythonMarks},
                KEY_ID + "=?", new String[]{String.valueOf(studentId)}, null, null, null, null);
        if (cursor.moveToNext()) {
            HashMap<String, String> student = new HashMap<>();
            student.put("rollNo", cursor.getString(cursor.getColumnIndex(KEY_ROLLNO)));
            student.put("name", cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            student.put("cMarks", cursor.getString(cursor.getColumnIndex(KEY_CMarks)));
            student.put("javaMarks",
cursor.getString(cursor.getColumnIndex(KEY_JavaMarks)));
            student.put("pythonMarks",
                    cursor.getString(cursor.getColumnIndex(KEY_PythonMarks)));
            student.put("id", cursor.getString(cursor.getColumnIndex(KEY_ID)));
            studentList.add(student);
        }
        return studentList;
    }
    public void DeleteStudent(int studentId) {
```

```
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_Students, KEY_ID + " = ?", new
String[]{String.valueOf(studentId)});
        db.close();
    }
    public int updateStudentDetails(String rollNo, String name, int cMarks, int javaMarks, int
pythonMarks, int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cValues = new ContentValues();
        cValues.put(KEY_ROLLNO, rollNo);
        cValues.put(KEY_NAME, name);
        cValues.put(KEY_CMarks, cMarks);
        cValues.put(KEY_JavaMarks, javaMarks);
        cValues.put(KEY_PythonMarks, pythonMarks);
        int count = db.update(TABLE_Students, cValues, KEY_ID + " = ?", new
                String[]{String.valueOf(id)});
        db.close();
        return count;
    }
    // Delete Student Details
    public void deleteStudent(int studentId) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_Students, KEY_ID + " = ?", new
String[]{String.valueOf(studentId)});
        db.close();
    }
    // Get Student Details (All)
    @SuppressLint("Range")
    public ArrayList<HashMap<String, String>> getStudents() {
        SQLiteDatabase db = this.getWritableDatabase();
        ArrayList<HashMap<String, String>> studentList = new ArrayList<>();
```

```
        String query = "SELECT rollno, name, c_marks, java_marks, python_marks, id FROM
" +
                TABLE_Students;
        Cursor cursor = db.rawQuery(query, null);
        while (cursor.moveToNext()) {
            HashMap<String, String> student = new HashMap<>();
            student.put("rollNo", cursor.getString(cursor.getColumnIndex(KEY_ROLLNO)));
            student.put("name", cursor.getString(cursor.getColumnIndex(KEY_NAME)));
            student.put("cMarks", cursor.getString(cursor.getColumnIndex(KEY_CMarks)));
            student.put("javaMarks",
cursor.getString(cursor.getColumnIndex(KEY_JavaMarks)));
            student.put("pythonMarks",
                    cursor.getString(cursor.getColumnIndex(KEY_PythonMarks)));
            student.put("id", cursor.getString(cursor.getColumnIndex(KEY_ID)));
            studentList.add(student);
        }
        return studentList;
    }
}
```

DetailsActivity.java

```
package com.example.student;


import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;
```

PRACTICAL-7
Implementation of SQLite Database

```
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import java.util.HashMap;
public class DetailsActivity extends AppCompatActivity {
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.details);
        DbHandler db = new DbHandler(this);
        ArrayList<HashMap<String, String>> studentList = db.GetStudents();
        ListView lv = findViewById(R.id.studentListView);
// Create a SimpleAdapter to populate the ListView with student data
        SimpleAdapter adapter = new SimpleAdapter(DetailsActivity.this, studentList,
R.layout.list_row,
                new String[]{"rollNo", "name", "cMarks", "javaMarks", "pythonMarks"},
                new int[]{R.id.rollNo, R.id.name, R.id.cMarks, R.id.javaMarks,
R.id.pythonMarks}
        );
        lv.setAdapter(adapter);
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                HashMap<String, String> selectedStudent = studentList.get(position);
                String studentId = selectedStudent.get("id");
                Toast.makeText(DetailsActivity.this, "Selected Student ID: " + studentId,
                        Toast.LENGTH_SHORT).show();
            }
        });
        Button backButton = findViewById(R.id.btnBack);
```

```
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                intent = new Intent(DetailsActivity.this, MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

MainActivity.java

```
package com.example.student;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText rollNo, name, cMarks, javaMarks, pythonMarks;
    Button addButton, updateButton, deleteButton, displayButton;
    Intent intent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rollNo = findViewById(R.id.txtRollNo);
        name = findViewById(R.id.txtName);
        cMarks = findViewById(R.id.txtCMarks);
```

```
javaMarks = findViewById(R.id.txtJavaMarks);
pythonMarks = findViewById(R.id.txtPythonMarks);
addButton = findViewById(R.id.btnAdd);
updateButton = findViewById(R.id.btnUpdate);
deleteButton = findViewById(R.id.btnDelete);
displayButton = findViewById(R.id.btnDisplay);
addButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String rollNo1 = rollNo.getText().toString();
        String name1 = name.getText().toString();
        String cMarks1 = cMarks.getText().toString();
        String javaMarks1 = javaMarks.getText().toString();
        String pythonMarks1 = pythonMarks.getText().toString();
        if (rollNo1.isEmpty() || name1.isEmpty()) {
            Toast.makeText(getApplicationContext(), "Roll No and Name are required.",
                    Toast.LENGTH_SHORT).show();
            return;
        }
        DbHandler dbHandler = new DbHandler(MainActivity.this);
        dbHandler.insertStudentDetails(rollNo1, name1, Integer.parseInt(cMarks1),
                Integer.parseInt(javaMarks1), Integer.parseInt(pythonMarks1));
        Toast.makeText(getApplicationContext(), "Student Details Inserted Successfully",
                Toast.LENGTH_SHORT).show();
    }
});
updateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String rollNo1 = rollNo.getText().toString();
```

```
            String name1 = name.getText().toString();
            String cMarks1 = cMarks.getText().toString();
            String javaMarks1 = javaMarks.getText().toString();
            String pythonMarks1 = pythonMarks.getText().toString();
// Get the student ID that you want to update (you should have this information)
            int studentIdToUpdate = 1; // Replace with the actual student ID
            DbHandler dbHandler = new DbHandler(MainActivity.this);
            int updatedRowCount = dbHandler.updateStudentDetails(rollNo1, name1,
                    Integer.parseInt(cMarks1), Integer.parseInt(javaMarks1),
                    Integer.parseInt(pythonMarks1),
                    studentIdToUpdate);
            if (updatedRowCount > 0) {
                Toast.makeText(getApplicationContext(), "Student Details Updated
Successfully",
                        Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(getApplicationContext(), "Failed to update student details",
                        Toast.LENGTH_SHORT).show();
            }
        }
    });
    deleteButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
// Get the student ID that you want to delete (you should have this information)
            int studentIdToDelete = 1; // Replace with the actual student ID
            DbHandler dbHandler = new DbHandler(MainActivity.this);
            dbHandler.deleteStudent(studentIdToDelete);
            Toast.makeText(getApplicationContext(), "Student Details Deleted Successfully",
                    Toast.LENGTH_SHORT).show();
```

```
        }
    });
    displayButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            intent = new Intent(MainActivity.this, DetailsActivity.class);
            startActivity(intent);
        }
    });
  }
}
```

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">

    <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.Student"
        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>
    <activity android:name=".DetailsActivity"></activity>
</application>

</manifest>
```

OUTPUT