Name: Raghav Maheshwari
Roll NO: 53
Panel: A

Lab Assignment - 11 (JP)

Aim:
Android SQLite Connectivity CRUD operations

Objectives:
→ To understand different CRUD operations.
→ To learn Android application and database connectivity.

Problem statement:
Create simple database using SQLite for an Android application and perform CRUD (Create, update, Read and Delete) database operations.

- CRUD operations on Database:
CRUD is an acronym which stands for Create, Read, update and Delete. These are basic functions of persistent storage. CRUD operations can use forms or an interface view to retrieve and return data from a database. Table records are read based on primary key within input parameter.

- About SQLite:
SQLite is an open source SQL database that stores data to text file on a device Android comes with built-in

SQLite implementation - SQLite supports all relational database features. Main package is android.database.sqlite.

- SQLite database creation:
In order to create a database, you just need to call the method 'openOrCreateDatabase' with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object. Syntax as follows:
SQLiteDatabase mydatabase = openOrCreateDatabase("your database-name", MODE_PRIVATE, null);

- SQLite database Insertion:
We can create table or insert data into table using execSQL method defined in SQLiteDatabase class. Syntax as follows:
mydatabase.execSQL("CREATE TABLE IF NOT EXISTS STUDENTS (Username VARCHAR, Password VARCHAR);");
mydatabase.execSQL("INSERT INTO STUDENTS VALUES ('admin', 'admin');");

- SQLite database Reading:
We can retrieve anything from database using an object of Cursor class. We will call a method of this class called rawQuery and it will return resultset with cursor pointing to table. We can move cursor forward and retrieve data. Syntax as follows

```
Cursor resultSet = mydatabase.rawQuery("Select * from
STUDENTS", null);
resultSet.moveToFirst();
String username = resultSet.getString(0);
String password = resultSet.getString(1);
```

- Functions of Cursor class

i) getColumnCount():
This method returns total no. of columns of table.

ii) getColumnIndex(String columnName):
This method returns index number of a column by specifying name of the column.

iii) getColumnName(int columnIndex)
This method returns name of column by specifying index of column.

iv) getColumnNames():
This method returns array of all column.

v) getCount():
This method returns total no. of rows in cursor.

vi) getPosition():
This method returns current position of cursor in table.

vii) isClosed():
This method returns true if cursor is closed and returns false otherwise.

- Platform: Windows
- Conclusion: Thus, we have learnt to develop android application with connection to SQLite database.

- FAQ

1. Write a query to update fields of SQLite database by taking input from user.

Ans Let's say, we have a table named COMPANY which has fields ID, Name, Age, Address and Salary. Then, by using following query we can update Address field for a customer whose ID is 6.

UPDATE COMPANY SET Address = 'Pune' WHERE ID = 6

2. Write difference b/w SQLite and MySQL.

Ans

SQLite	MySQL
→ Developed by D. Richard Hip in August 2000.	→ Developed by Oracle in May 1995.
→ It was developed only in C Language.	→ It was developed in C and C++.
→ Does not support xml format.	→ Supports xml format.
→ Does not require a server to run, hence it is serverless.	→ Requires a server for functioning, hence it follows client / server architecture

3. Write steps to create database applications in Android.

Ans → Create a new Project in Android Studio with suitable name.

→ Select a Project Template, select empty activity and click next.

→ Include CRUD operations in your Java class by importing relevant packages and methods.

→ Design your layout of app by making appropriate changes to your Activity.xml file.

→ Create your emulator device and now you can run your app on virtual machine.

1) **MainActivity.java Class**

```
package com.example.exno5;

import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      Marks=(EditText)findViewById(R.id.Marks);
      Insert=(Button)findViewById(R.id.Insert);
      Delete=(Button)findViewById(R.id.Delete);
      Update=(Button)findViewById(R.id.Update);
      View=(Button)findViewById(R.id.View);
      ViewAll=(Button)findViewById(R.id.ViewAll);

      Insert.setOnClickListener(this);
```

```
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

        // Creating database and table
        db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE,
null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno
VARCHAR,name VARCHAR,marks VARCHAR);");
    }
    public void onClick(View view)
    {
        // Inserting a record to the Student table
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().length()==0||
                    Name.getText().toString().trim().length()==0||
                    Marks.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter all values");
                return;
            }
            db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+
                    "','"+Marks.getText()+"');");
            showMessage("Success", "Record added");
            clearText();
        }
        // Deleting a record from the Student table
        if(view==Delete)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
```

```
            return;
         }
         Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
         if(c.moveToFirst())
         {
            db.execSQL("DELETE FROM student WHERE
rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Deleted");
         }
         else
         {
            showMessage("Error", "Invalid Rollno");
         }
         clearText();
      }
      // Updating a record in the Student table
      if(view==Update)
      {
         // Checking for empty roll number
         if(Rollno.getText().toString().trim().length()==0)
         {
            showMessage("Error", "Please enter Rollno");
            return;
         }
         Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
         if(c.moveToFirst()) {
            db.execSQL("UPDATE student SET name='" + Name.getText() +
"',marks='" + Marks.getText() +
                  "' WHERE rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Modified");
         }
         else {
            showMessage("Error", "Invalid Rollno");
         }
         clearText();
```

```
        }
        // Display a record from the Student table
        if(view==View)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                Name.setText(c.getString(1));
                Marks.setText(c.getString(2));
            }
            else
            {
                showMessage("Error", "Invalid Rollno");
                clearText();
            }
        }
        // Displaying all the records
        if(view==ViewAll)
        {
            Cursor c=db.rawQuery("SELECT * FROM student", null);
            if(c.getCount()==0)
            {
                showMessage("Error", "No records found");
                return;
            }
            StringBuffer buffer=new StringBuffer();
            while(c.moveToNext())
            {
                buffer.append("Rollno: "+c.getString(0)+"\n");
                buffer.append("Name: "+c.getString(1)+"\n");
```

```
            buffer.append("Marks: "+c.getString(2)+"\n\n");
          }
          showMessage("Student Details", buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

2) Activity.xml file code:

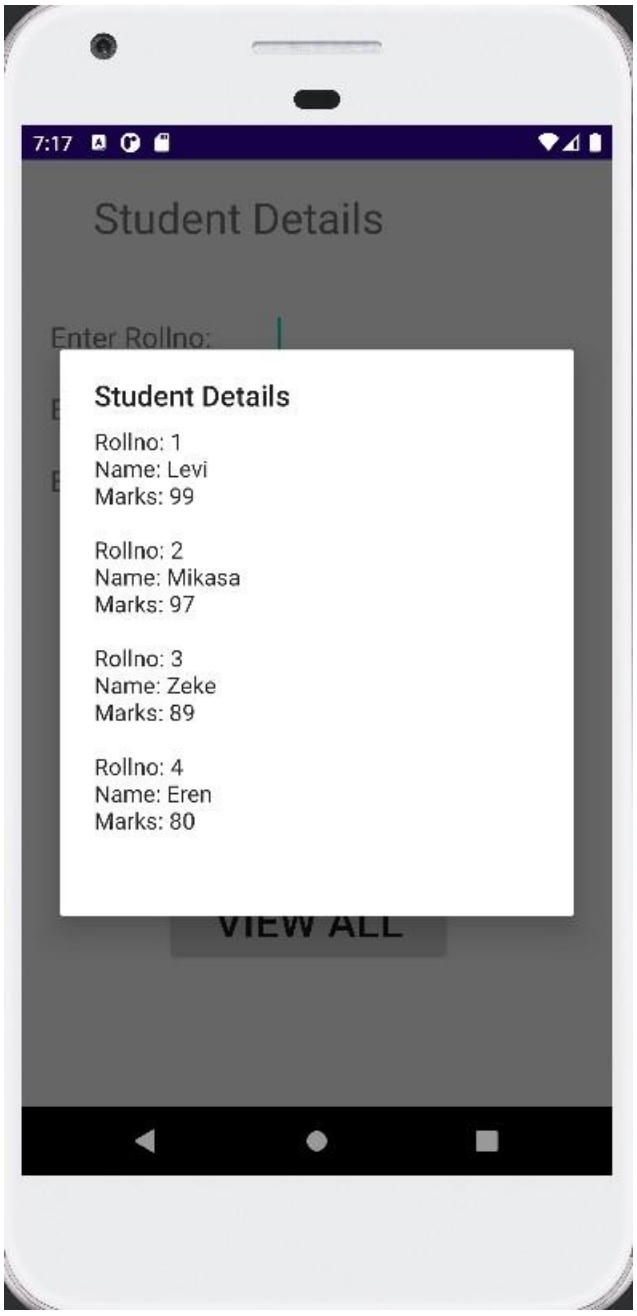
```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />
```

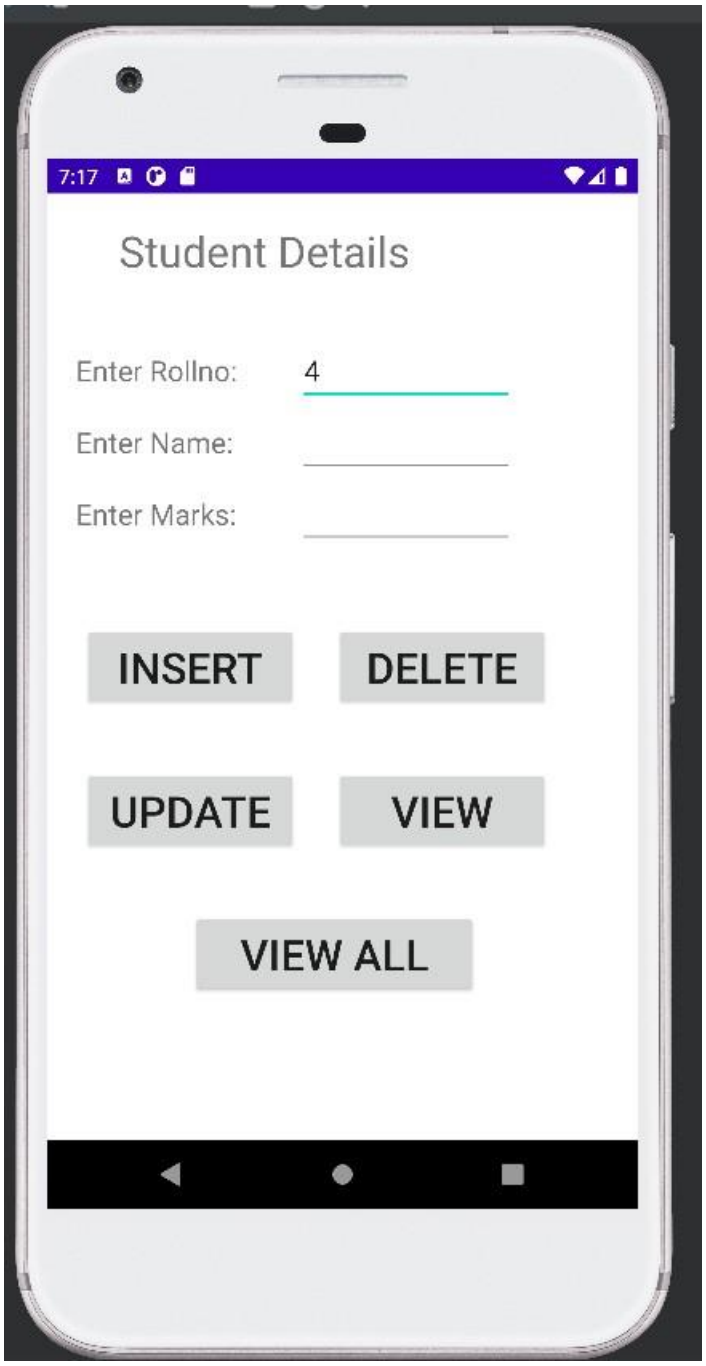
```
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="110dp"
   android:text="Enter Rollno:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Rollno"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="100dp"
   android:inputType="number"
   android:textSize="20sp" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="160dp"
   android:text="Enter Name:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Name"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="150dp"
   android:inputType="text"
   android:textSize="20sp" />

<TextView
   android:layout_width="wrap_content"
```

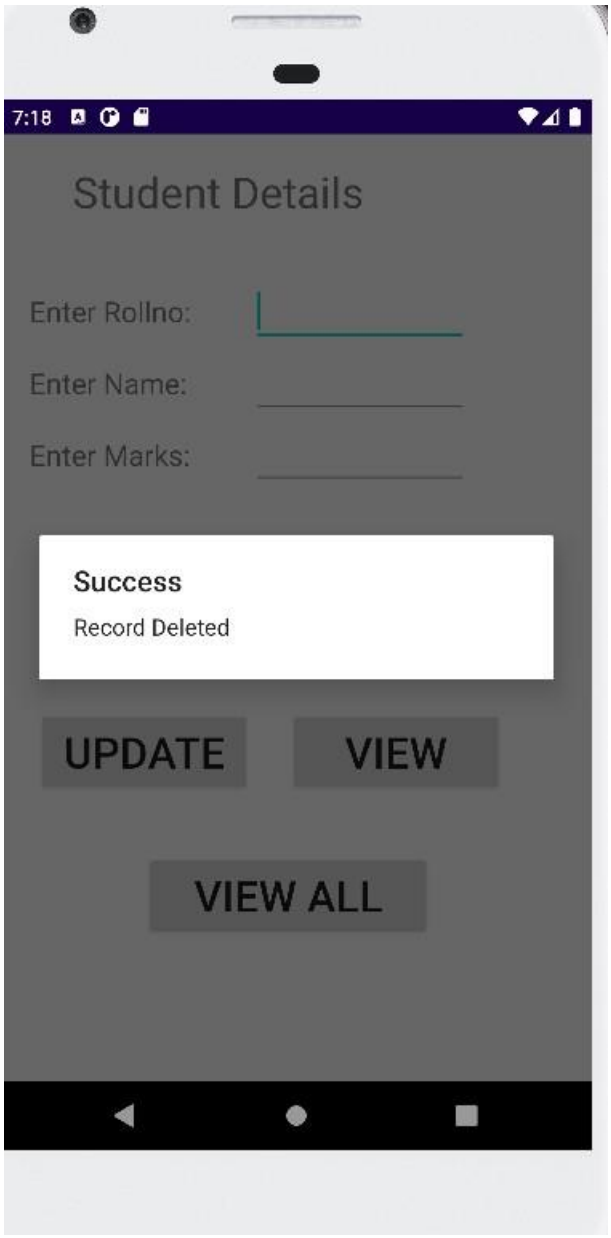
```
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
```

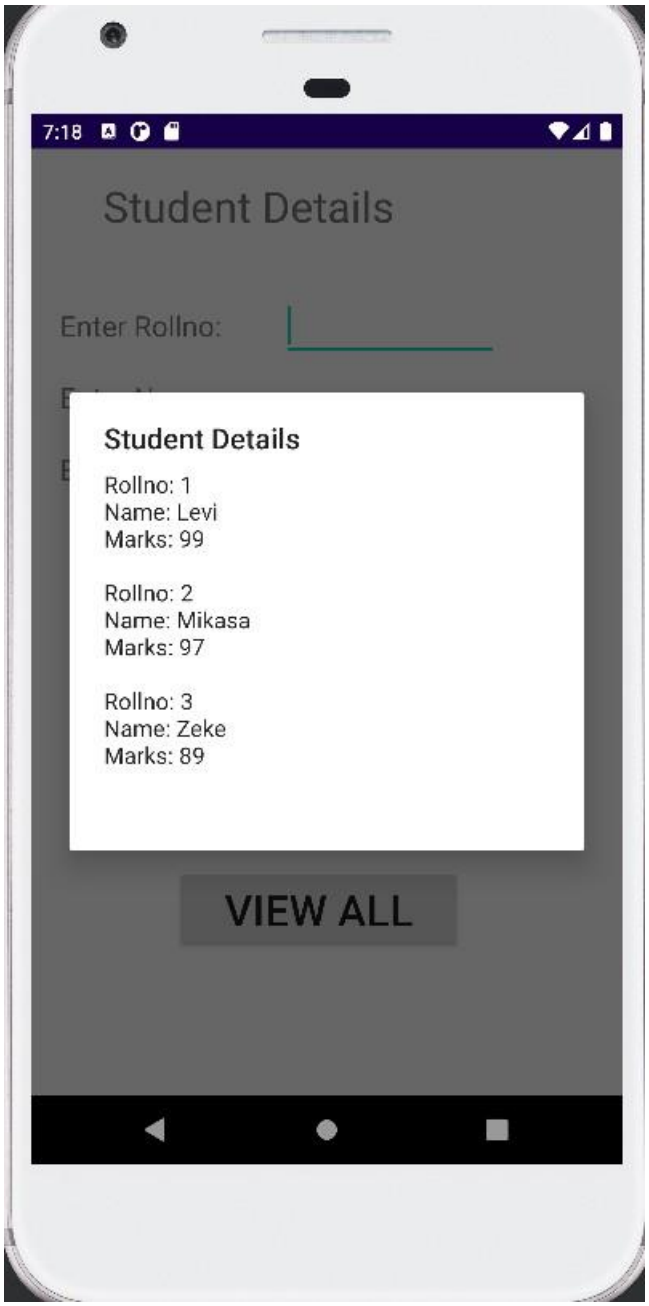
```
        android:layout_x="25dp"
        android:layout_y="400dp"
        android:text="Update"
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

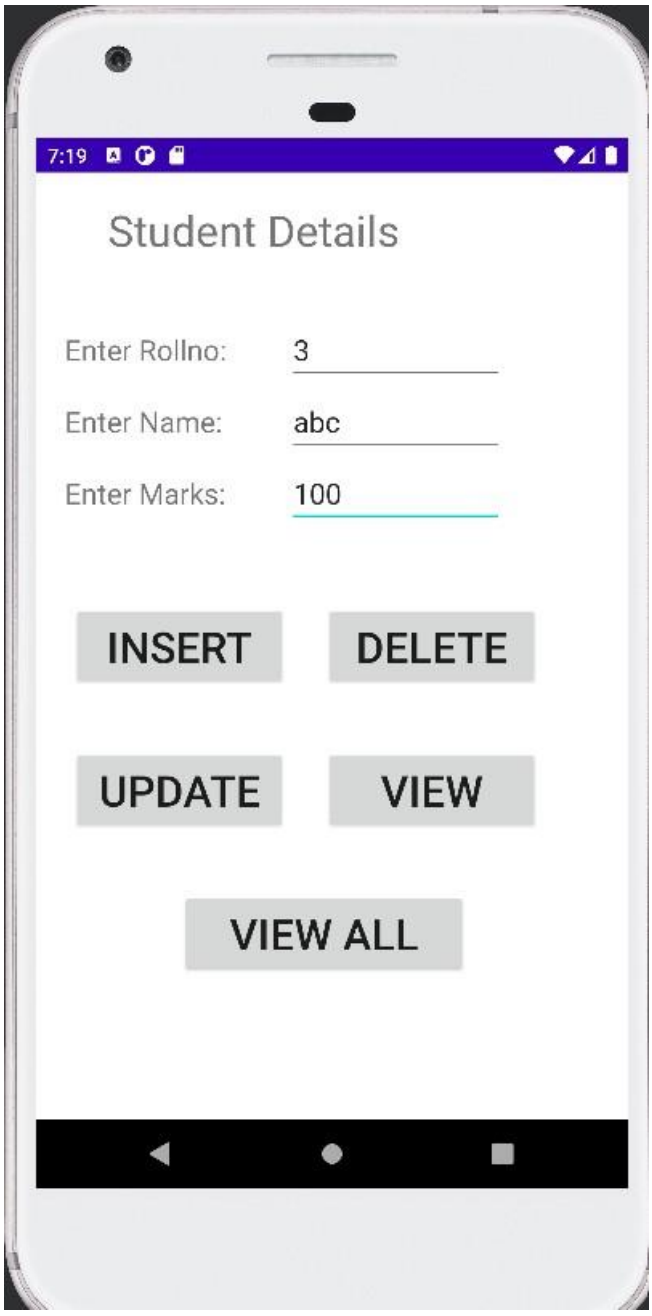
Output:

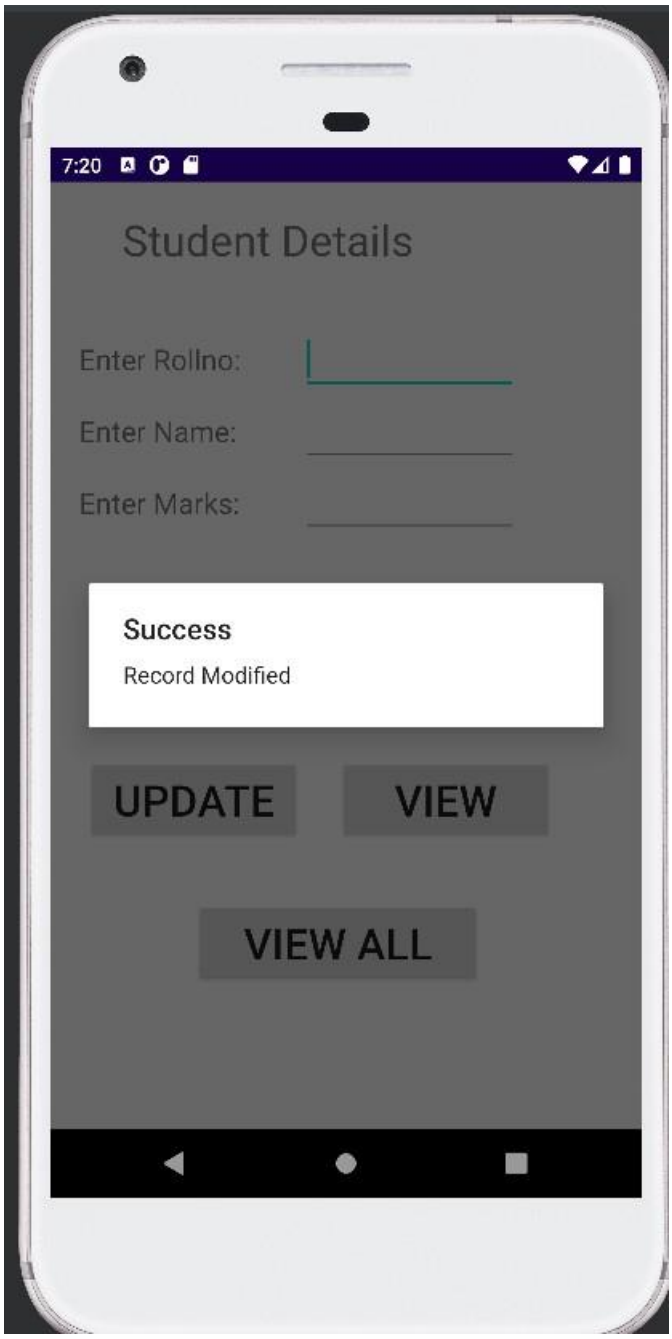

7:17
Student Details
Enter Rollno:
Student Details
Rollno: 1
Name: Levi
Marks: 99
Rollno: 2
Name: Mikasa
Marks: 97
Rollno: 3
Name: Zeke
Marks: 89
Rollno: 4
Name: Eren
Marks: 80
VIEW ALL

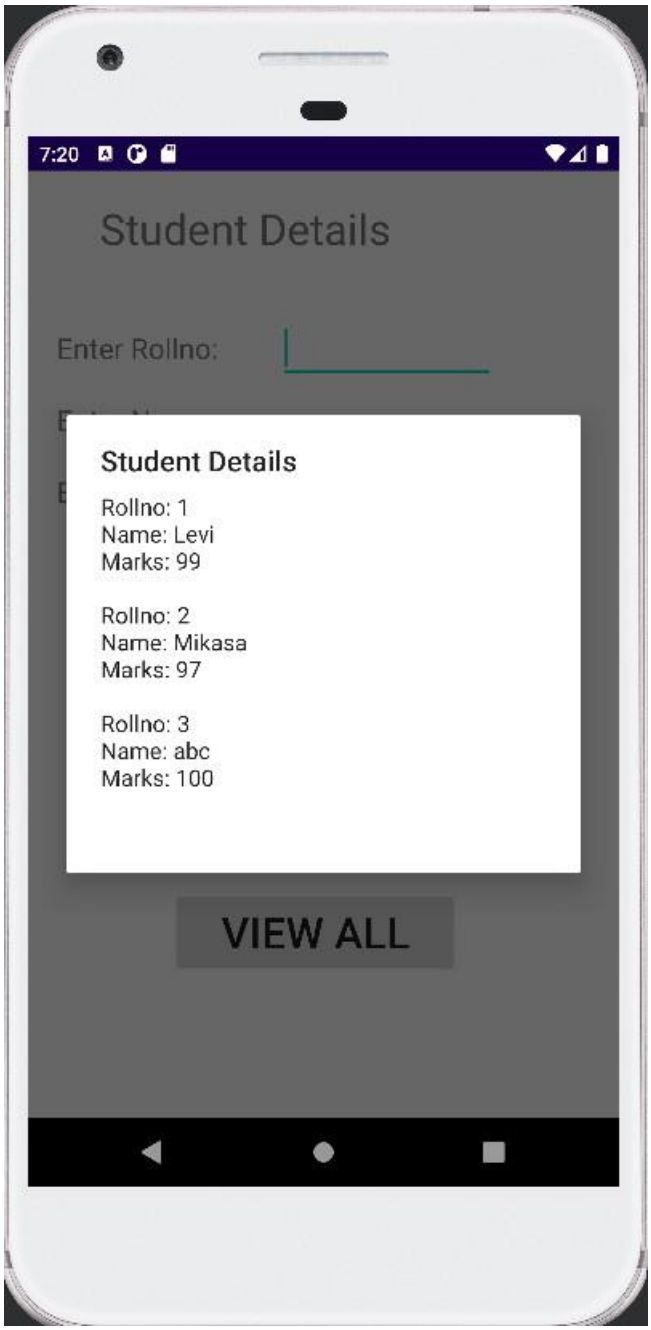

7:17
Student Details
Enter Rollno:
4
Enter Name:
Enter Marks:
INSERT
DELETE
UPDATE
VIEW
VIEW ALL

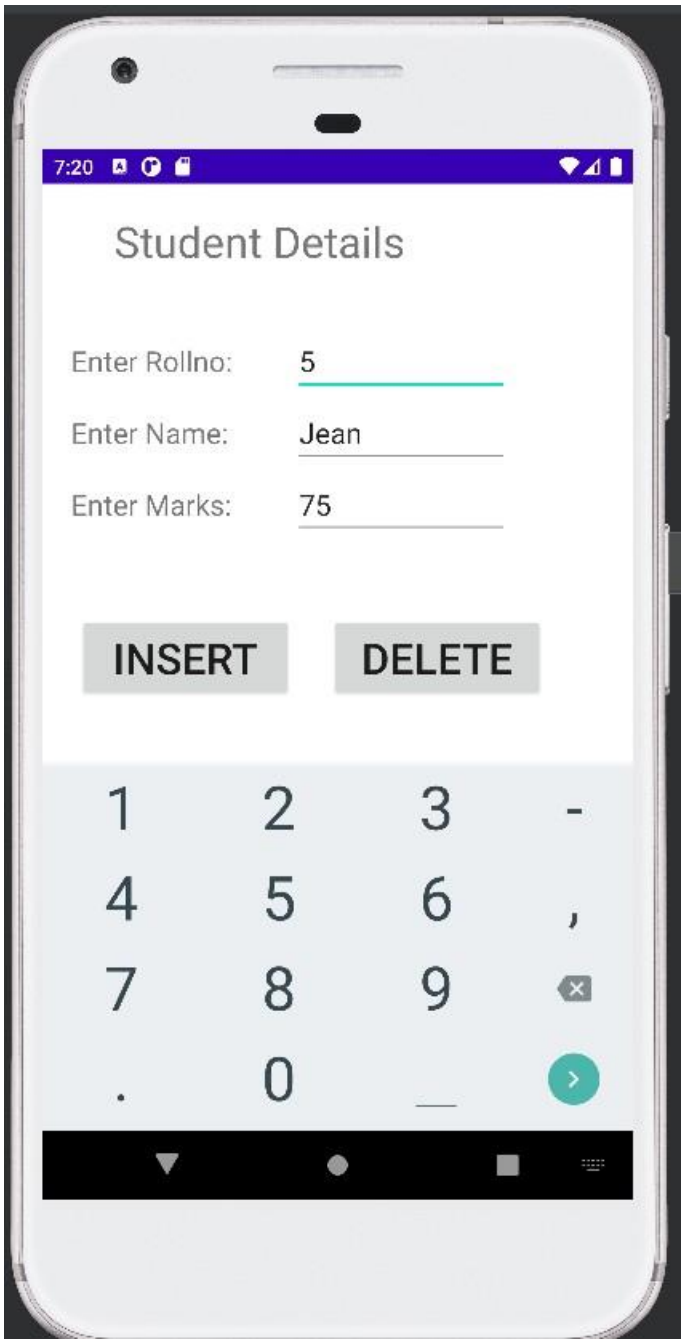

7:18
Student Details
Enter Rollno:
Enter Name:
Enter Marks:
Success
Record Deleted
UPDATE
VIEW
VIEW ALL

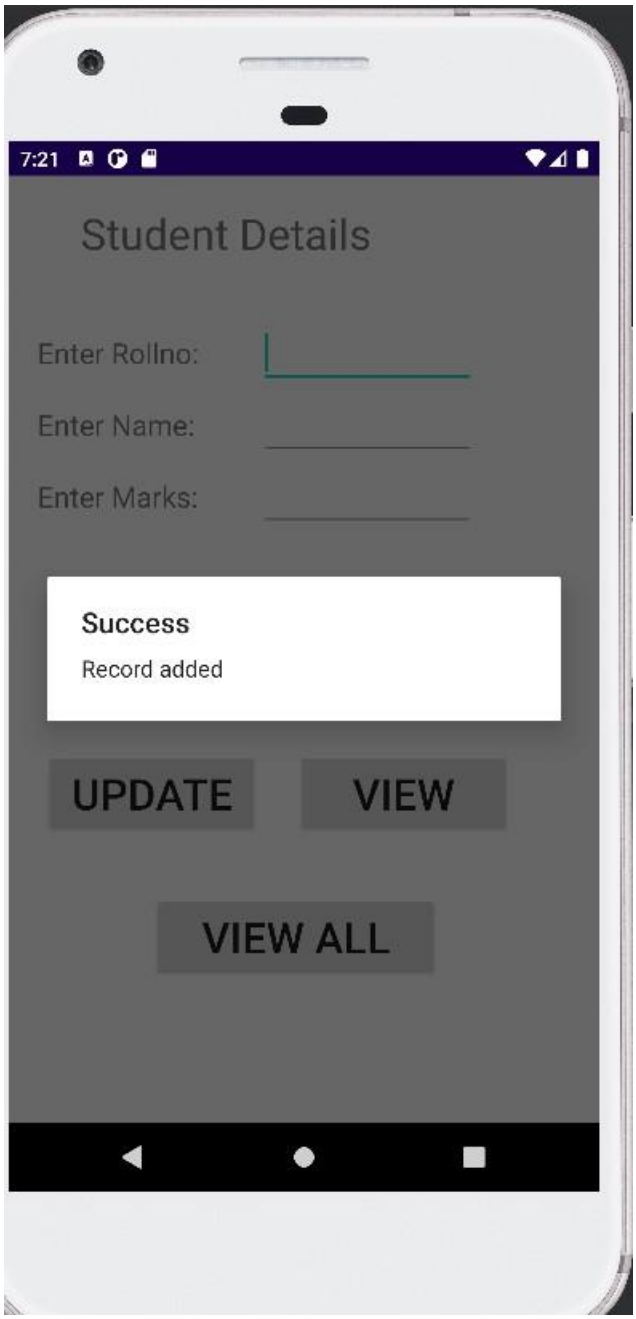

7:18
Student Details
Enter Rollno:
Student Details
Rollno: 1
Name: Levi
Marks: 99
Rollno: 2
Name: Mikasa
Marks: 97
Rollno: 3
Name: Zeke
Marks: 89
VIEW ALL

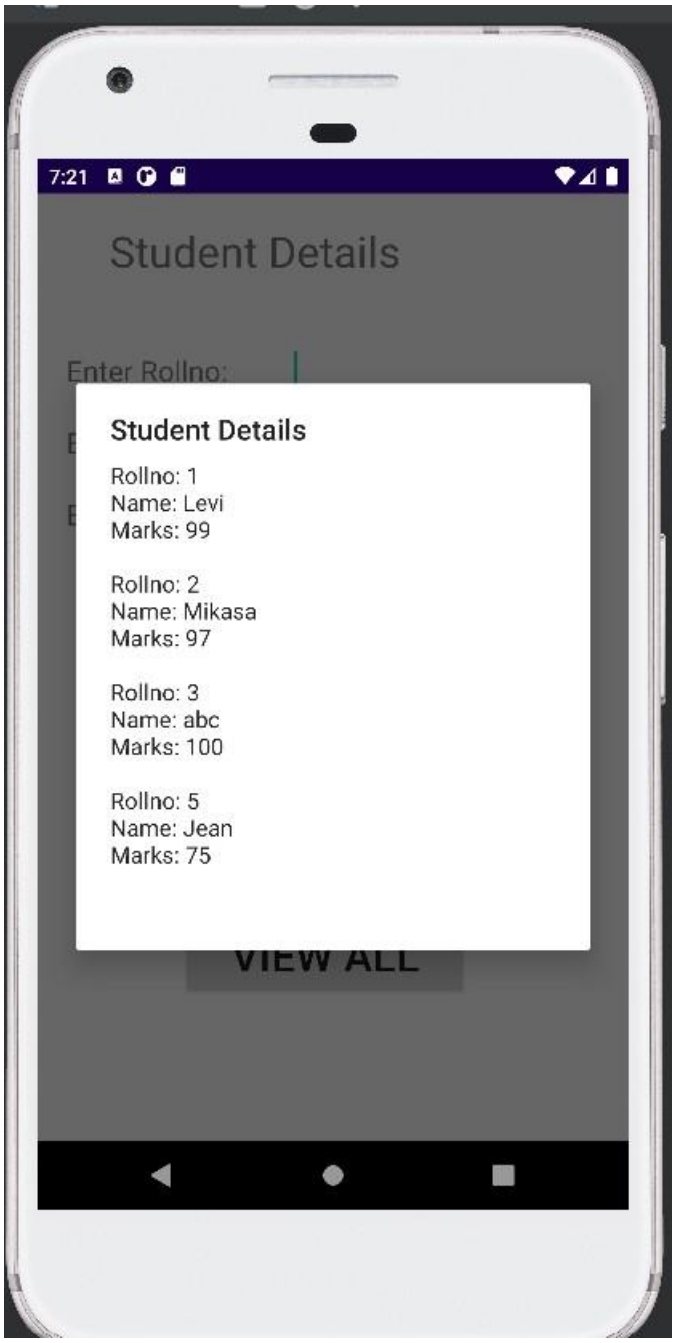

7:19
Student Details
Enter Rollno:
3
Enter Name:
abc
Enter Marks:
100
INSERT
DELETE
UPDATE
VIEW
VIEW ALL


7:20
Student Details
Enter Rollno:
Enter Name:
Enter Marks:
Success
Record Modified
UPDATE
VIEW
VIEW ALL


7:20
Student Details
Enter Rollno:
Student Details
Rollno: 1
Name: Levi
Marks: 99
Rollno: 2
Name: Mikasa
Marks: 97
Rollno: 3
Name: abc
Marks: 100
VIEW ALL


7:20
Student Details
Enter Rollno:
5
Enter Name:
Jean
Enter Marks:
75
INSERT
DELETE
1
2
3
-
4
5
6
,
7
8
9
.
0


7:21
Student Details
Enter Rollno:
Enter Name:
Enter Marks:
Success
Record added
UPDATE
VIEW
VIEW ALL


7:21
Student Details
Enter Rollno:
Student Details
Rollno: 1
Name: Levi
Marks: 99
Rollno: 2
Name: Mikasa
Marks: 97
Rollno: 3
Name: abc
Marks: 100
Rollno: 5
Name: Jean
Marks: 75
VIEW ALL