PRACTICAL 6

**AIM :** Android App GridLayout and with Database Connectivity using sqlite for updating student records.

**Scenario :** Create an Android App which would add student records in sqlite table, delete, modify and view specific student already stored records of given rollno. View All to display all records stored in the table. Create the UI using GridLayout and add appropriate validation popup dialog boxes.

|  |  |
| --- | --- |
|  |  |

**Part B (to be completed by students)**

**(Students must submit the soft copy as per the following segments. The soft copy must be uploaded on the Blackboard. The filename should be Batch\_RollNo\_Exp\_No)**

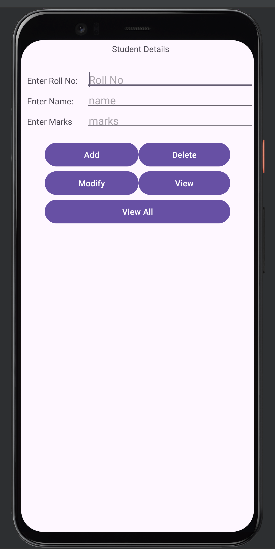
|  |  |
| --- | --- |
| **Roll No.: K005** | **Name: Jal Bafana** |
| **Prog/Yr/Sem: Cyber Security/2nd/4th** | **Batch: K1** |
| **Date of Experiment:** | **Date of Submission:** |

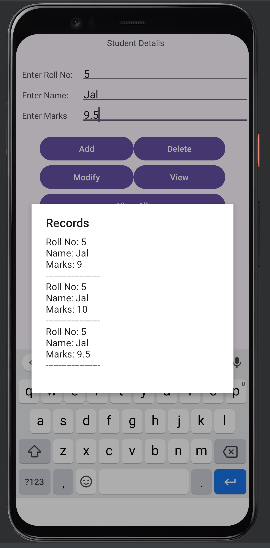
1. **Program Scenario and Program code:** (Write Scenario and Paste your program code (Java, xml resource and layout)).

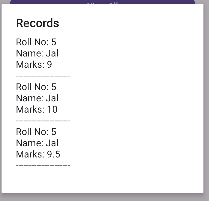
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity" >  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:shrinkColumns="0"  
 android:stretchColumns="1">  
 <TableRow android:padding="5dip">  
 <TextView  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:layout\_span = "2"  
 android:gravity="center\_horizontal"  
 android:text="Student Details"/>  
 </TableRow>  
 <TableRow>  
 <TextView  
 android:layout\_height="wrap\_content"  
 android:layout\_column="0"  
 android:layout\_marginLeft="10dp"  
 android:text="Enter Roll No: "/>  
 <EditText  
 android:id="@+id/txt\_roll"  
 android:layout\_column="1"  
 android:layout\_marginLeft="10dp"  
 android:hint="Roll No"  
 android:padding="5dp"/>  
 </TableRow>  
  
 <TableRow>  
 <TextView  
 android:layout\_height="wrap\_content"  
 android:layout\_column="0"  
 android:layout\_marginLeft="10dp"  
 android:text="Enter Name: "/>  
 <EditText  
 android:id="@+id/txt\_name"  
 android:layout\_column="1"  
 android:layout\_marginLeft="10dp"  
 android:hint="name"  
 android:padding="5dp"/>  
 </TableRow>  
  
 <TableRow>  
 <TextView  
 android:layout\_height="wrap\_content"  
 android:layout\_column="0"  
 android:layout\_marginLeft="10dp"  
 android:text="Enter Marks "/>  
 <EditText  
 android:id="@+id/txt\_marks"  
 android:layout\_column="1"  
 android:layout\_marginLeft="10dp"  
 android:hint="marks"  
 android:padding="5dp"/>  
 </TableRow>  
  
 </TableLayout>  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:stretchColumns="\*"  
 android:layout\_marginTop="16dp"  
 android:paddingHorizontal="40dp">  
 <TableRow>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btn\_add"  
 android:onClick="add"  
 android:text="Add"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="delete"  
 android:text="Delete"  
 android:id="@+id/btn\_delete"/>  
 </TableRow>  
  
 <TableRow>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Modify"  
 android:onClick="modify"  
 android:id="@+id/btn\_modify"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="View"  
 android:onClick="view"  
 android:id="@+id/btn\_view"/>  
 </TableRow>  
  
 <TableRow>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="View All"  
 android:onClick="viewall"  
 android:layout\_span="2"  
 android:id="@+id/btn\_viewall"/>  
 </TableRow>  
  
 </TableLayout>  
  
</LinearLayout>

package com.example.k005\_db\_lab;  
import android.os.Bundle;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.text.Html;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
  
public class MainActivity extends AppCompatActivity {  
  
 EditText txtroll, txtname, txtmarks;  
  
 Button btnadd, btnmodify, btnview, btndelete, btnviewall;  
  
 SQLiteDatabase db;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 btnadd = findViewById(R.id.*btn\_add*);  
 btndelete = findViewById(R.id.*btn\_delete*);  
 btnmodify = findViewById(R.id.*btn\_modify*);  
 btnview = findViewById(R.id.*btn\_view*);  
 btnviewall = findViewById(R.id.*btn\_viewall*);  
  
 txtroll = findViewById(R.id.*txt\_roll*);  
 txtname = findViewById(R.id.*txt\_name*);  
 txtmarks = findViewById(R.id.*txt\_marks*);  
  
 db = openOrCreateDatabase("Studentdb", Context.*MODE\_PRIVATE*, null);  
 db.execSQL("CREATE TABLE IF NOT EXISTS " + "student(rollno text, name text, marks real);");  
 }  
 public void add(View v){  
 if(check()){  
 db.execSQL("insert into student values('"  
 +txtroll.getText().toString()  
 +"','"+ txtname.getText().toString()  
 +"',"+ Float.*parseFloat*(txtmarks.getText().toString()) +");");  
 showmsg("Added", "Record saved");  
 }  
 }  
  
 public void delete(View v){  
 if(txtroll.getText().toString().isEmpty()){  
 showmsg("Error", "Roll No not entered");  
 }  
 else{  
 Cursor c = db.rawQuery("select \* from student where rollno ='"  
 +txtroll.getText().toString()+"';",null);  
 if(c.moveToFirst()){  
 db.execSQL("delete from student where rollno ='"  
 +txtroll.getText().toString()+"';");  
 showmsg("Deleted",  
 "Record Deleted" + txtroll.getText().toString());  
 }  
 else{  
 showmsg("Error",txtroll.getText().toString()+"does not exist");  
 }  
 }  
  
 }  
 public boolean check(){  
 if (txtroll.getText().toString().isEmpty() || txtname.getText().toString().isEmpty() || txtmarks.getText().toString().isEmpty()){  
 showmsg("Error","Fields cannot be blank");  
 return false;  
 }else{  
 return true;  
 }  
 }  
  
 public void showmsg(String title, String msg){  
 {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle(title);  
 builder.setMessage(msg);  
 builder.show();  
 }  
 }  
  
 public void viewall(View v){  
 Cursor c = db.rawQuery("select \* from student; ",null);  
 if(c.getCount() == 0){  
 showmsg("Error", "Table Empty");  
 }else{  
 StringBuffer buffer = new StringBuffer();  
 while(c.moveToNext()){  
 buffer.append("Roll No: "+c.getString(0)+"\n");  
 buffer.append("Name: "+c.getString(1)+"\n");  
 buffer.append("Marks: "+c.getString(2)+"\n");  
 buffer.append("--------------------- \n");  
 }  
 showmsg("Records", buffer.toString());  
 }  
 }  
  
 public void view(View v){  
 if(txtroll.getText().toString().isEmpty()){  
 showmsg("Error", "Roll No cannot be blank");  
 }  
 else{  
 Cursor c = db.rawQuery("select \* from student where rollno ='"+txtroll.getText().toString()+"';",null);  
 if(c.moveToFirst()){  
 txtname.setText(c.getString(1));  
 txtmarks.setText(c.getString(2));  
 }else{  
 showmsg("Error","does not exist");  
 }  
 }  
 }  
  
 public void modify(View v){  
 if(v == btnmodify) {  
 int r=Integer.*parseInt*(txtroll.getText().toString());  
 String n= txtname.getText().toString();  
 float m=Float.*parseFloat*(txtmarks.getText().toString());  
 if(txtroll.getText().toString().isEmpty() ||txtname.getText().toString().isEmpty() || txtmarks.getText().toString().isEmpty()) {  
 showmsg("Error","Input all fields");  
 }  
 else {  
 Cursor c=db.rawQuery("select \* from student where rollno="+r+";",null);  
 if(c.getCount()==0) {  
 showmsg("Error","Rollno not existing");  
 }  
 else { //update student set name='n', marks=67 where rollno=r  
 db.execSQL("update student set name='" + n + "' , marks="+ m +" where rollno=" + r +";");  
 showmsg("Success","Record updated");  
 }  
 }  
 }  
 }  
  
}

1. **Output:** (Paste your program input and output screen shots).

****

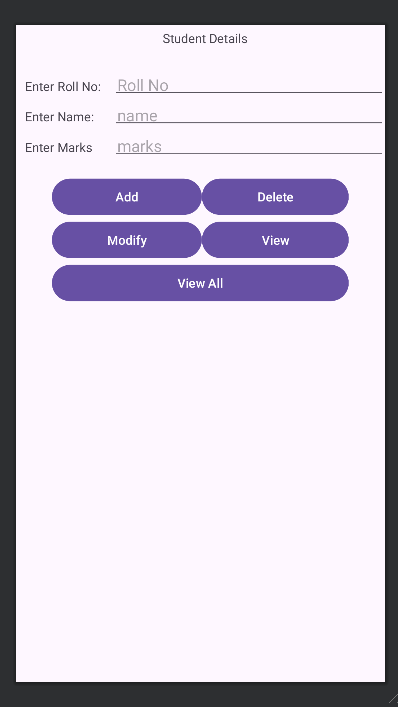
****

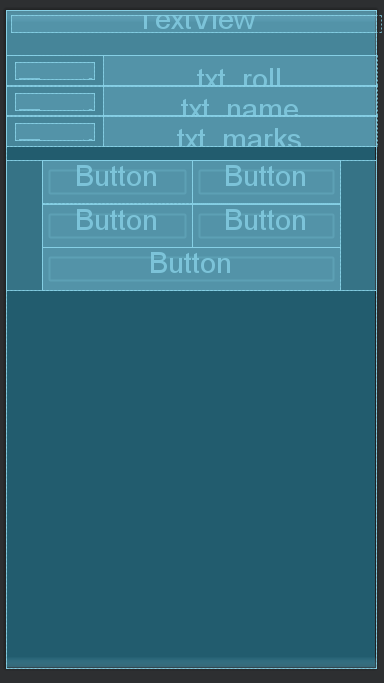
****

1. **Observations:** A brief description of the design aspects and working of the code in your own words.

The app uses MaterialApp with a custom theme and sets the home screen to Demo.  
The app’s root widget is BMIApp, and the Demo widget likely handles the main UI.

1. **Questions:** Draw & Explain with respect to layouts the Scene Graph of the experiment.

****

****

1. **Conclusion (Learning Outcomes):** How were the outcomes defined for the experiment in Part A fulfilled through the scenarios?

The app is structured with a modern Material Design approach and focuses on BMI functionality.  
It seems prepared to build a user-friendly interface for BMI calculation.