AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
Affiliated to Savitribai Phule Pune University, Approved by AICTE,
New Delhi and Recognised by Govt. of Maharashtra
(ID No. PU/PN/ENGG/124/1998)
Accredited by NAAC with 'A' Grade

A
MINI-PROJECT REPORT
ON

STUDENT DATABASE

SUBMITTED TO SPPU, PUNE

IN PARTIAL FULFILLMENT OF THE THIRD YEAR IN COMPUTER ENGINEERING 2019-20

By

YASH JITENDRA KASAT
NIKHIL RAMDAS MANDALE
DANESH ADIL PAGDIWALLA

Guided by
Mrs. P. S. Gaikwad

AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
Affiliated to Savitribai Phule Pune University, Approved by AICTE,
New Delhi and Recognised by Govt. of Maharashtra
(ID No. PU/PN/ENGG/124/1998)
Accredited by NAAC with 'A' Grade

ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S
INSTITUTE OF INFORMATION TECHNOLOGY
PUNE – 01.

DEPARTMENT OF COMPUTER ENGINEERING

Mini Project Report On

"STUDENT DATABASE"

Submitted by

YASH JITENDRA KASAT
NIKHIL RAMDAS MANDALE
DANESH ADIL PAGDIWALLA

In partial fulfillment for the project of

Third Year of Engineering
(Computer Engineering)

Guided by
Name of the Guide
Mrs. P. S. Gaikwad

CERTIFICATE

This is to certify that, the Mini Project "**STUDENT DATABASE**" submitted by **Yash Jitendra Kasat (30), Nikhil Ramdas Mandale (37), Danesh Adil Pagdiwalla (97)** is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Third Year of Engineering (Computer Engineering) Degree of Savitribai Phule Pune University.

Place: Pune.
Date: 05/12/2019

Mrs. P. S. Gaikwad
(Project Guide)

Dr. S. N. Zaware
(Head Of Department)

ACKNOWLEDGMENT

We feel happiness in forwarding this project report as an image of sincere effort. The successful project reflects our work, efforts of our guide in giving us good information

Our sincere thanks to our guide respected Prof **Mrs. P. S. Gaikwad** who has been a consistent source of inspiration and guiding star in achieving our goal. We give our special thanks to respected Prof **Dr. S. N. Zaware (H.O.D)** for her consistent interest and her encouragement throughout the completion of our project.

TABLE OF CONTENTS **Page No.**

1: INTRODUCTION

Modern handheld devices such as smartphones have become increasingly powerful in recent years.
As mobile devices become more like PCs they will come to replace objects we tend to carry around such as checkbooks, credit cards, cameras, planners, mp3 players, etc.

We will be using them to accomplish our daily tasks. One application that falls into this category is to store data. Current database storage often offers decryption of data but few have been applied to online commerce.

The project has the following objectives:

- To provide Android Mobile Application that will enable storing a student database.
- It will help in showing the details and full information about the enrolled students.
- We can store the Roll No, Name, Email, Courses, Marks and other student-related information in the database.
- This application is more likely for a teacher, professors, etc.

2: LITERATURE SURVEY

Existing application:

The student database is an android application used for storing data of students.

Proposed application:

Features:

- Student Database is portable and can be accessed from anywhere.
- Any updation that needs to make it is easy and simple.
- Wrong entries can be deleted in a quick time.
- This application is safe and secure for future reference.

Drawbacks:

- When we store the data there is no internet connection back up, so we can store permanently.
- We have to clear the entries manually.
- Fewer fields of storing data.
- Design is somewhat peachy.

3: SOFTWARE REQUIREMENT SPECIFICATION

Name of System/Application Software	**Software Version**
Android Studio	AndroidStudio-ide-191.5900203
SDK, SDK manager (Software Development kit)	The Lowest Version is 2.2 and Above
SQLite	SQLite-3.19
XML	XML Version 1.0
Emulator	Emulator v2.3

4: OVERALL DESCRIPTION

4.1 Project Goals

Our project has three main goals. The first goal is to replicate the use of paper for storing data on an individual basis. Our second goal is to make it easy to update, delete, insert, etc. This involves improving the consumption of time. The Third goal is to be able to give the proper information to the individual for their needs. Milestones include installing the browser, configuring proper network settings.

Our main use case will involve inserting the data on the go. The user will update as they want at any time within some seconds. Afterward, the user can use it for future reference.

There are many possibilities for what could be stored in a student database. We could denote Name, Roll No, Address, Email address, Courses that He/She enrolled, Marks, Grades, etc. Thus, we have streamlined the transfer of data in a simple and elegant manner.

4.2 Project Implementation

Our system architecture can be organized into four main components: Login, Database, Information of students, User Interface.

We created the login page for a single user from that only he/she can access the data. We have created the SQLite database with the help of SQLiteOpenHelper as a superclass then implemented functions like Insert, Update, Delete, View, View All, etc. To connect these functions to the main activity class we have used InputMethodManager and view. We implemented the OnclickListener method to view the contents.

To implement the methods we used functions like Curser, Context, Bundle, Button, Toast, ContentValues, etc.

Context –

A **Context** is a handle to the system; it provides services like resolving resources, obtaining access to databases and preferences, and so on. An **Android** app has activities. **Context** is like a handle to the environment your application is currently running in. The activity object inherits the **Context** object.

Curser –

Cursors are what contains the result set of a query made against a database in **Android**. The **Cursor** class has an API that allows an app to read (in a type-safe manner) the columns that were returned from the query as well as iterate over the rows of the result set.

Bundle –

A bundle is used to pass data between Activities. You can create a **bundle**, pass it to Intent that starts the activity which then can be used from the destination activity. **Bundle**:- A mapping from String values to various Parcelable types. **The bundle** is generally used for passing data between various activities of **android.**

Toast –

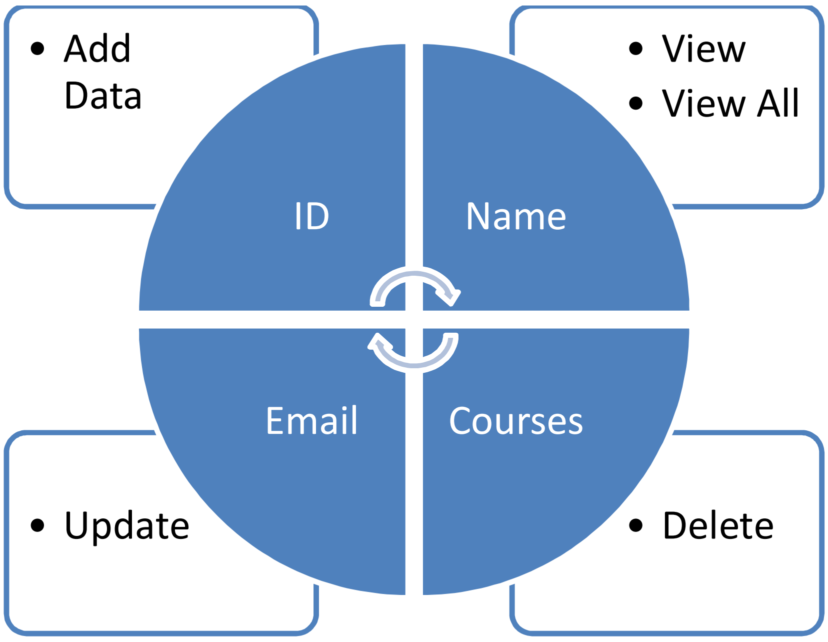
Android **Toast** can be used to display information for a short period of time. A **toast** contains a message to be displayed quickly and disappears after some time. The **android**.widget.**Toast** class is the subclass of java.lang.Object class. You can also create custom **toast** as well for example **toast** displaying image.

ContentValues –

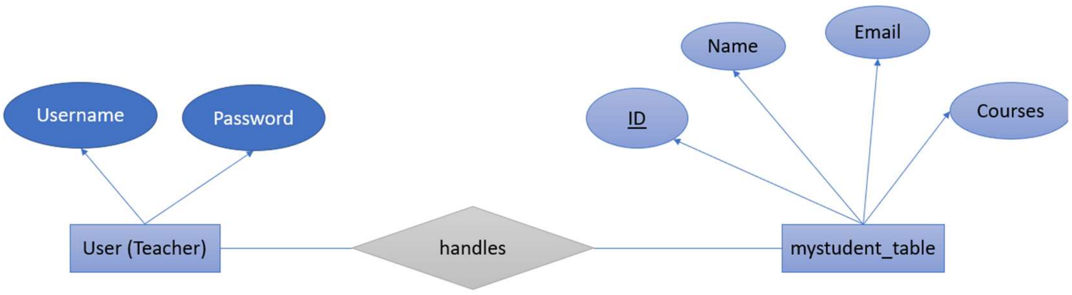
Public constructors. **ContentValues**(). Creates an empty set of values using the default initial size. **ContentValues**(int size).

5. SYSTEM FEATURES

5.1 Data flow diagram –



5.2 ER Diagram –



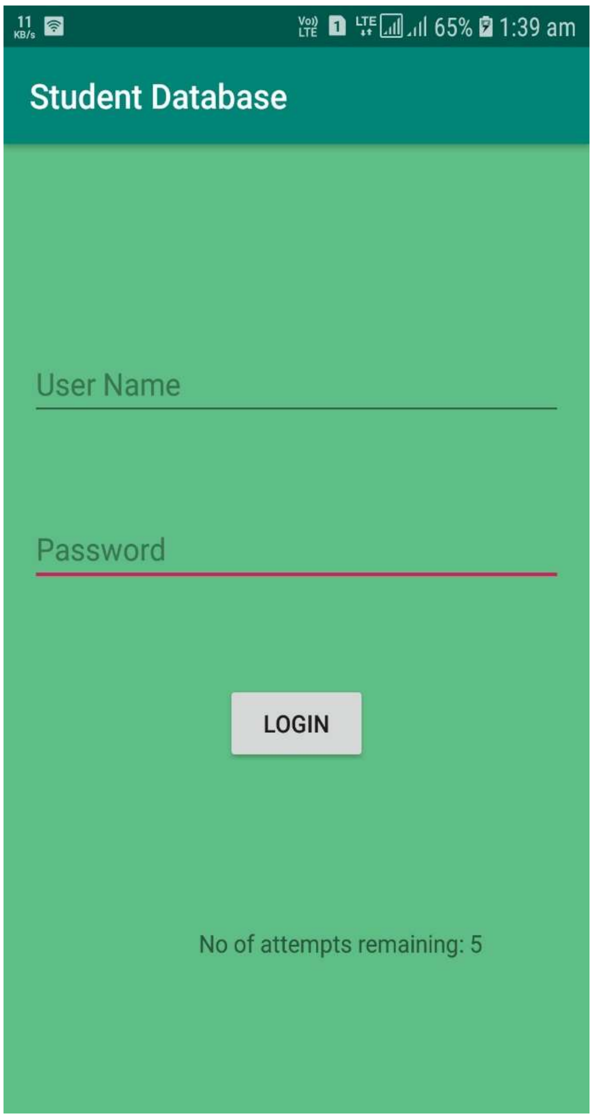
6. EXTERNAL INTERFACE REQUIREMENTS

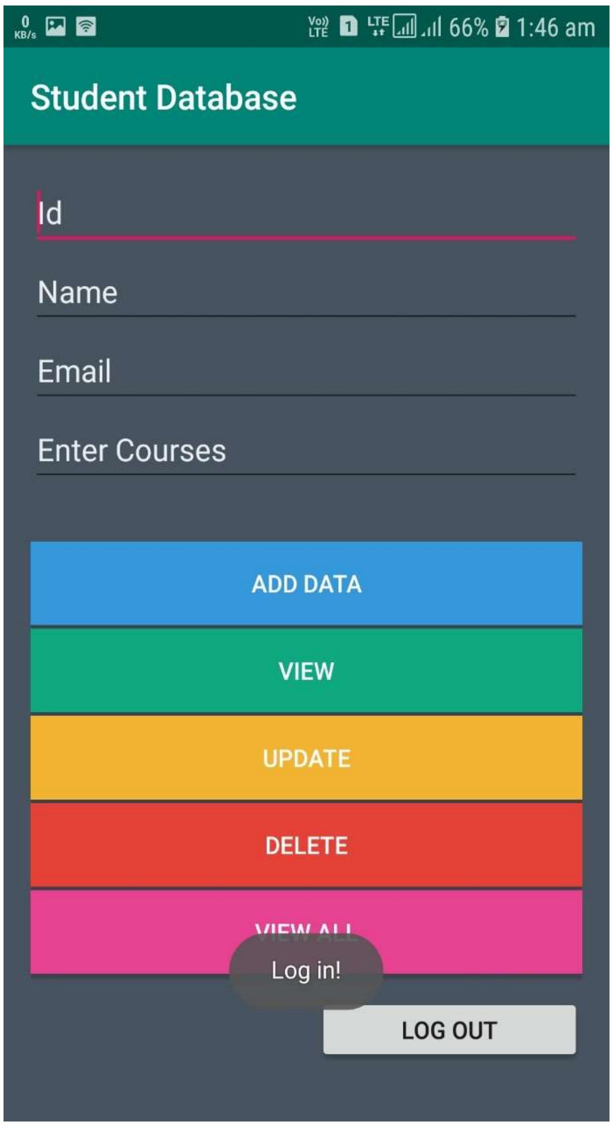
6.1 Hardware interface -

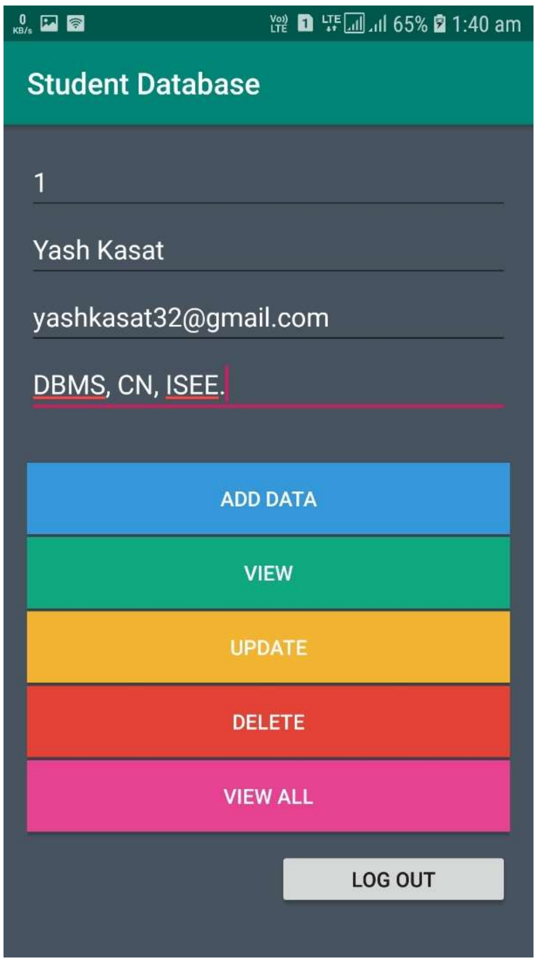
Peripheral/Device	**Configuration**
Android Phone	RAM: 126 MB Android Version: 2.3.6 and above Processor: 600 Mhz

7. USER INTERFACE –

Screenshots –





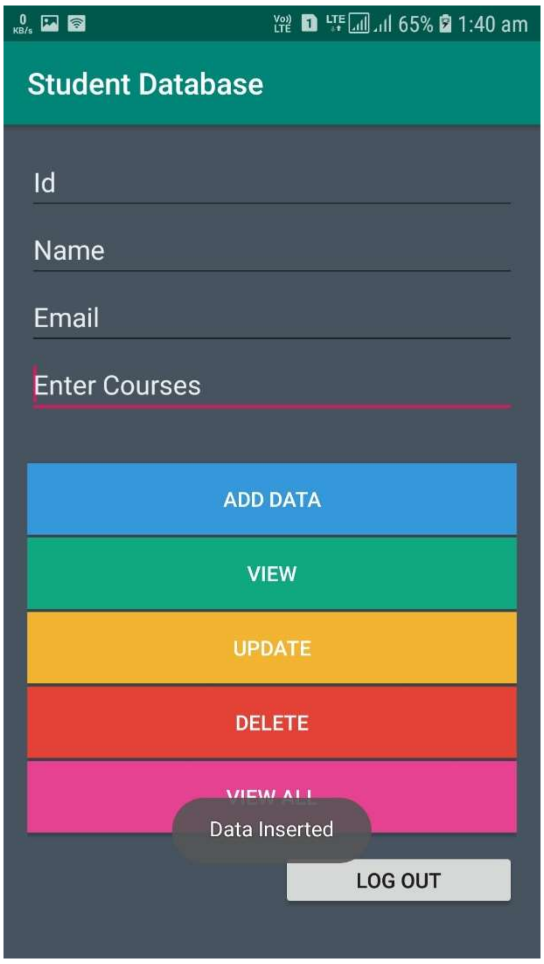


1

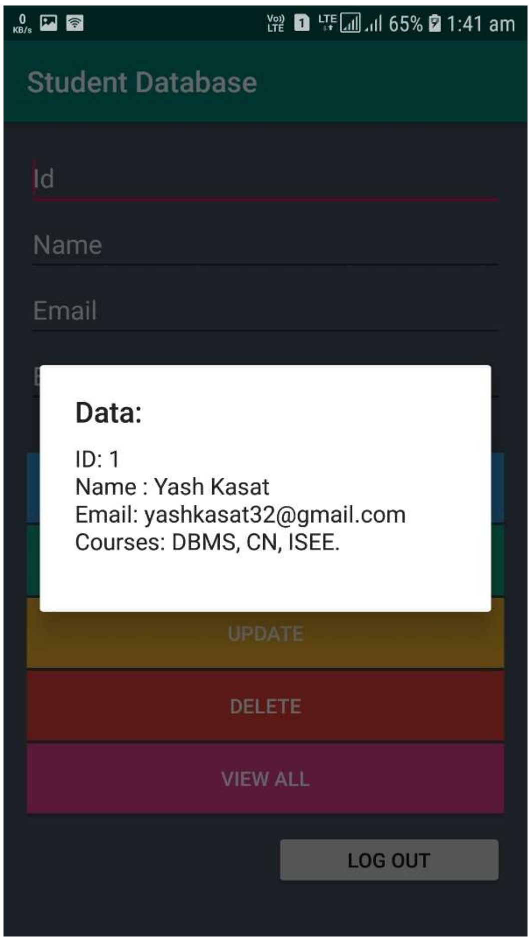
Yash Kasat

yashkasat32@gmail.com

DBMS, CN, ISEE.

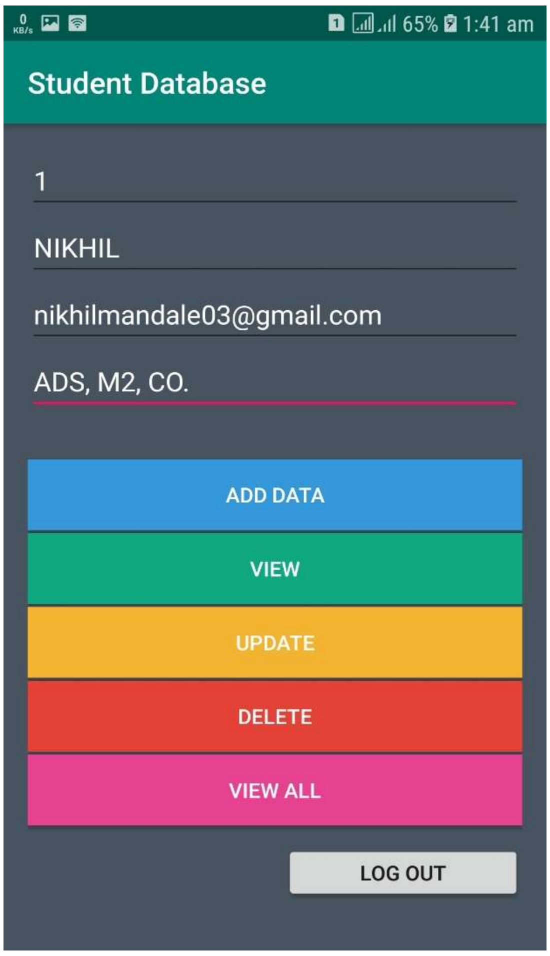


Data Inserted



Data:

ID: 1
Name : Yash Kasat
Email: yashkasat32@gmail.com
Courses: DBMS, CN, ISEE.

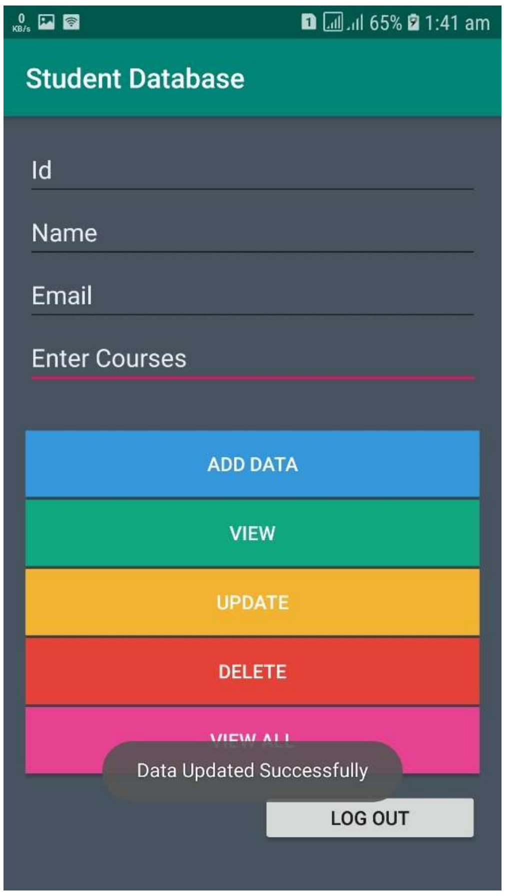


1

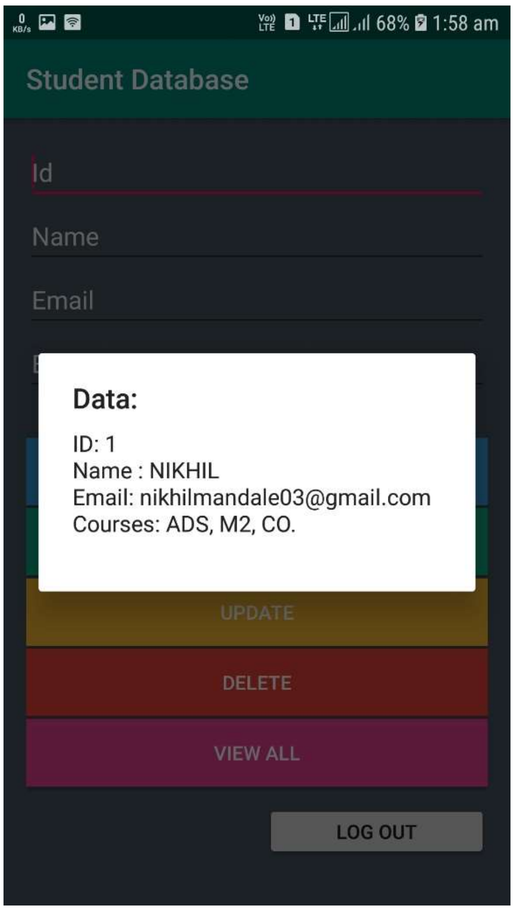
NIKHIL

nikhilmandale03@gmail.com

ADS, M2, CO.

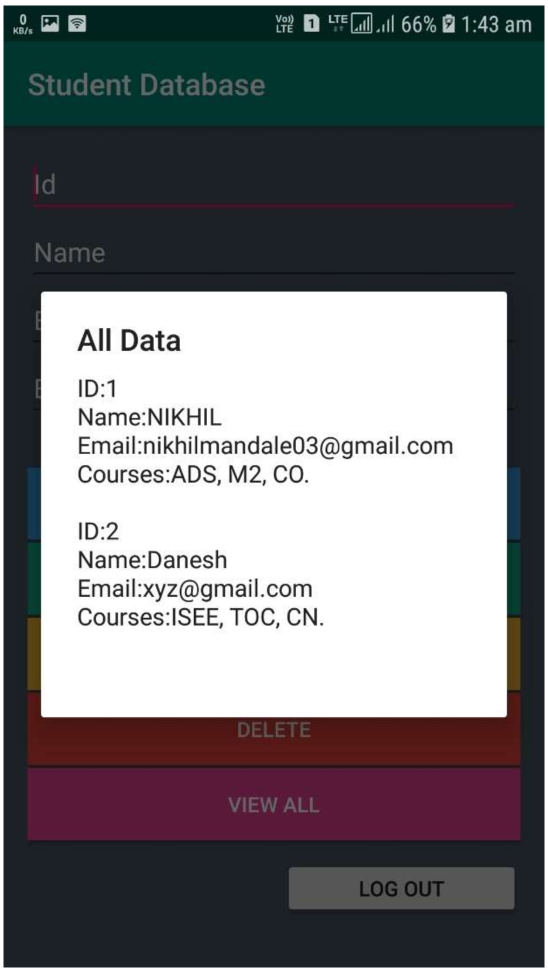


Data Updated Successfully



Data:

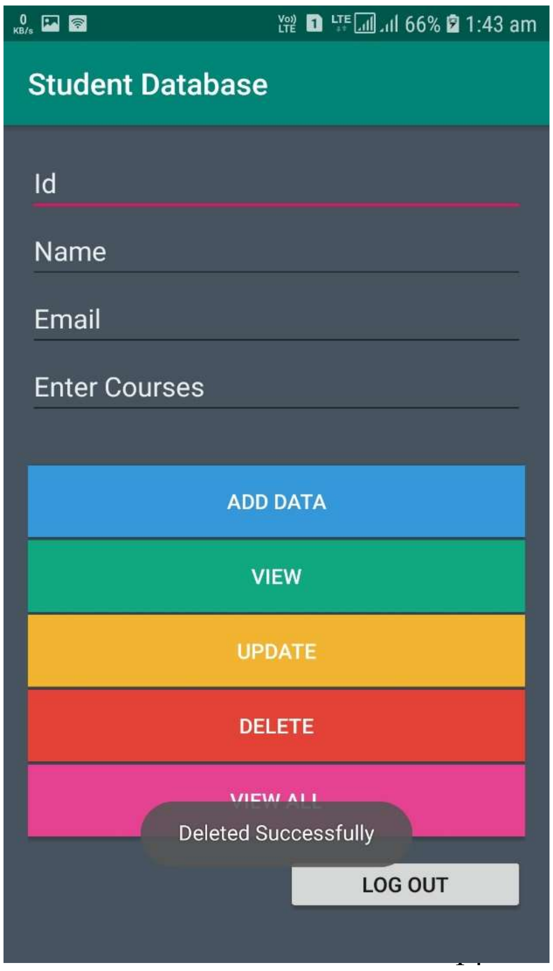
ID: 1
Name : NIKHIL
Email: nikhilmandale03@gmail.com
Courses: ADS, M2, CO.



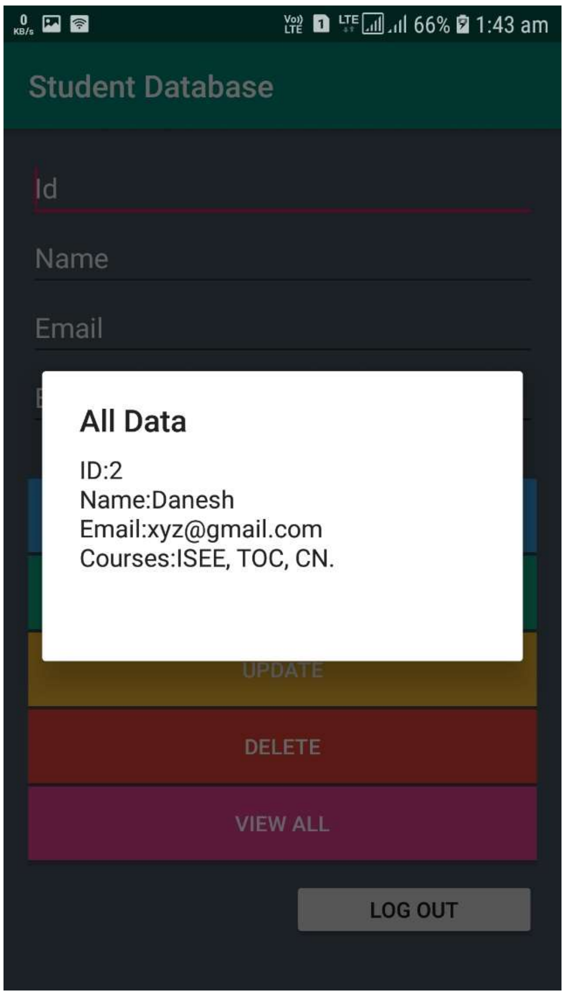
All Data

ID:1
Name:NIKHIL
Email:nikhilmandale03@gmail.com
Courses:ADS, M2, CO.

ID:2
Name:Danesh
Email:xyz@gmail.com
Courses:ISEE, TOC, CN.

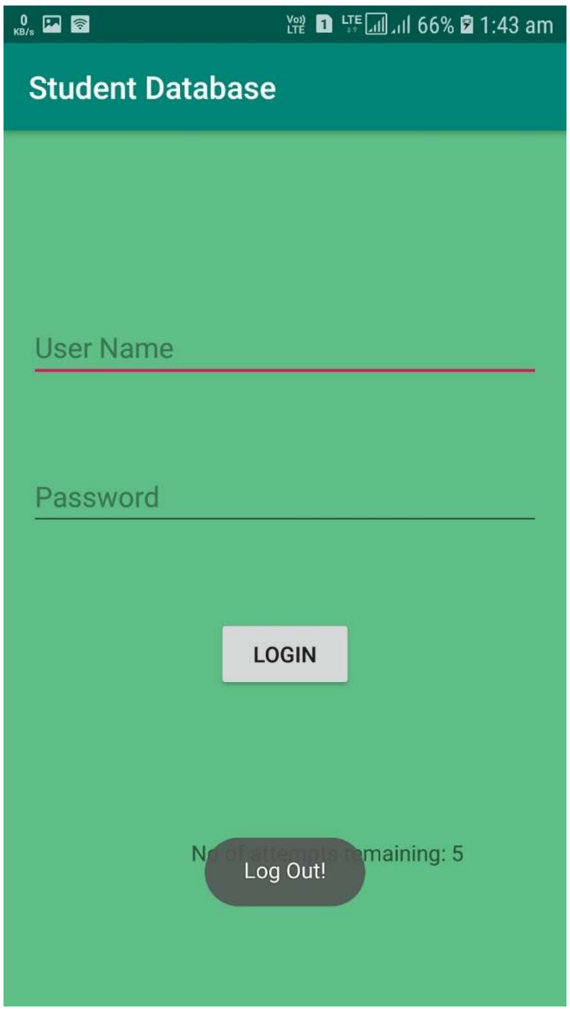


Deleted Successfully



All Data

ID:2
Name:Danesh
Email:xyz@gmail.com
Courses:ISEE, TOC, CN.



No. of attempts remaining: 5

Log Out!

USER MANUAL –

STARTUP SESSION:

This is used to start up the application to perform the desired operation.

MENUS:

It shows the menus which help the user to perform the desired operations.

Login Menu :

In this menu we there is a login section for a particular user.

Student database Menu:

This Menu shows the methods to Create, View, View all, Update, Delete data from the database.

EXIT SESSION:

This Option allows us to Exit the Application when our work is done.

8. CODING –

// MainActivity.java

```
package com.example.studentdatabase;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText Name;
    private EditText Password;
    private TextView Info;
    private Button Login;
    private int counter = 5;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Name = (EditText) findViewById(R.id.username2);
        Password = (EditText) findViewById(R.id.password);
        Info = (TextView) findViewById(R.id.textView);
        Login = (Button) findViewById(R.id.login);

        Info.setText("No of attempts remaining: 5");

        Login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String Name1 =Name.getText().toString();
                String Pwd = Password.getText().toString();
                if(Name1.equalsIgnoreCase("Admin") && Pwd.equals("nikh")){

                    Intent MainIntent = new
Intent(MainActivity.this,Main2Activity.class);

                    Name.setText("");
                    Password.setText("");

                    startActivity(MainIntent);
                    Toast.makeText(MainActivity.this,"Log
in!",Toast.LENGTH_SHORT).show();
                }else
                    {
                        Toast.makeText(MainActivity.this,"Enter correct
User Name and Password",Toast.LENGTH_LONG).show();
                        counter--;

                        Info.setText("No of attempts remaining:
```

```
"+String.valueOf(counter));

                        if(counter == 0) {
                            Login.setEnabled(false);
                        }
                }

                //validate(Name.getText().toString(),
Password.getText().toString());
            }
        });

    }

   /* private void validate(String userName, String userPass) {

        if((userName == "Nik") && (userPass == "1234")) {
            Intent intent = new Intent(MainActivity.this,
Main2Activity.class);
            startActivity(intent);
            Toast.makeText(MainActivity.this,"Log
in",Toast.LENGTH_SHORT).show();


        }
        /*else {
            counter--;

            Info.setText("No of attempts remaining:
"+String.valueOf(counter));

            if(counter == 0) {
                Login.setEnabled(false);
            }
        }*/
    //}
}
```

// main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="#47535E"
    tools:context=".Main2Activity">

    <EditText
        android:id="@+id/editText_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:hint="Id"
        android:textColor="#fff"
        android:textColorHint="#EAF0F1"
        android:inputType="number" />

    <EditText
        android:id="@+id/editText_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColorHint="#EAF0F1"
        android:textColor="#fff"
        android:layout_below="@id/editText_id"
        android:hint="Name"
        android:imeOptions="actionNext"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/editText_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColorHint="#EAF0F1"
        android:textColor="#fff"
        android:layout_below="@id/editText_name"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/editText_CC"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editText_email"
        android:hint="Enter Courses"
        android:textColor="#fff"
        android:textColorHint="#EAF0F1"
        android:layout_marginBottom="20dp"
        android:imeOptions="actionGo" />

    <Button
        android:id="@+id/button_add"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editText_CC"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="2dp"
        android:background="#3498DB"
        android:gravity="center"
        android:text="Add Data"
        android:textColor="#fff" />

    <Button
        android:id="@+id/button_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/button_add"
        android:layout_centerHorizontal="true"
        android:textColor="#fff"
        android:background="#10A881"
        android:layout_marginBottom="2dp"
        android:gravity="center"
        android:text="View" />

    <Button
        android:id="@+id/button_update"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/button_view"
        android:layout_centerHorizontal="true"
        android:textColor="#fff"
        android:background="#F3B431"
        android:layout_marginBottom="2dp"
        android:gravity="center"
        android:text="Update" />

    <Button
        android:id="@+id/button_delete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/button_update"
        android:textColor="#fff"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="2dp"
        android:background="#E44236"
        android:gravity="center"
        android:text="Delete" />

    <Button
        android:id="@+id/button_viewALL"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#fff"
        android:layout_marginBottom="2dp"
        android:layout_below="@id/button_delete"
        android:layout_centerHorizontal="true"
        android:background="#E74292"
        android:gravity="center"
        android:text="View All" />

    <Button
        android:id="@+id/logout"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:layout_below="@id/button_viewALL"
        android:layout_marginLeft="170dp"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:text="Log Out" />

</RelativeLayout>
```

// DatabaseHelper.java

```
package com.example.studentdatabase;

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

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {

    public final static String DATABASE_NAME ="MyStudent .db";
    public final static String TABLE_NAME ="myStudent_table";
    public final static String COL_1 ="ID";
    public final static String COL_2 ="NAME";
    public final static String COL_3 ="EMAIL";
    public final static String COL_4 ="COURSE_COUNT";

    public DatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS "
                + TABLE_NAME+
                "(ID BYTE(3) NOT NULL PRIMARY KEY," +
                " NAME TEXT NOT NULL UNIQUE," +
                " EMAIL TEXT NOT NULL UNIQUE," +
                " COURSE_COUNT TEXT NOT NULL)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(sqLiteDatabase);
    }

    public boolean insertData(String id, String name, String email, String
courseCount) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_1, id);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, email);
        contentValues.put(COL_4, courseCount);

        long result = db.insert(TABLE_NAME, null, contentValues);
        if (result ==-1){
            return false;
        } else {
            return true;
        }

    }

    public boolean updateData(String id, String name, String email, String
courseCount){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_1, id);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, email);
        contentValues.put(COL_4, courseCount);

        db.update(TABLE_NAME, contentValues,"ID=?", new String[]{id});
        return true;
    }

    public Cursor getData(String id){
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM "+TABLE_NAME+" WHERE ID='"+id+"'";
        Cursor cursor = db.rawQuery(query, null);

        return cursor;
    }

    public Integer deleteData(String id){
        SQLiteDatabase db = this.getWritableDatabase();

        return db.delete(TABLE_NAME,"ID=?", new String[]{id});
    }

    public Cursor getAllData(){
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM "+TABLE_NAME, null);
        return cursor;
    }

    /*public Cursor getUser() {
        SQLiteDatabase db =this.getReadableDatabase();
        Cursor res= db.rawQuery("SELECT * FROM "+TABLE_NAME +" ",null);
        return res;
    }*/
}
```

// Main2Activity.java

```
package com.example.studentdatabase;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Main2Activity extends AppCompatActivity {

    private Button LogOut;
    DatabaseHelper myDB;
    EditText editTextId, editName, editEmail, editCC;
    Button buttonAdd, buttonGetData, buttonUpdate, buttonDelete,
buttonViewAll, button_ExportToExcel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        myDB = new DatabaseHelper(this);

        editTextId = findViewById(R.id.editText_id);
        editName = findViewById(R.id.editText_name);
        editEmail = findViewById(R.id.editText_email);
        editCC = findViewById(R.id.editText_CC);


        buttonAdd = findViewById(R.id.button_add);
        buttonDelete = findViewById(R.id.button_delete);
        buttonUpdate = findViewById(R.id.button_update);
        buttonGetData = findViewById(R.id.button_view);
        buttonViewAll = findViewById(R.id.button_viewALL);

        LogOut = findViewById(R.id.logout);
        //button_ExportToExcel = findViewById(R.id.button_ExportToExcel);

        AddData();
        getData();
        viewAll();
        updateData();
        deleteData();

        LogOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(true){

                    Intent MainIntent = new
Intent(Main2Activity.this,MainActivity.class);
```

```
                    startActivity(MainIntent);
                    finish();
                    Toast.makeText(Main2Activity.this,"Log
Out!",Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    public void AddData() {
        buttonAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String id = editTextId.getText().toString();
                String name= editName.getText().toString();
                String email = editEmail.getText().toString();
                String course = editCC.getText().toString();


                if(id.equals(String.valueOf(""))) {
                    editTextId.setError("Enter ID");
                    return;
                }
                if(name.equals(String.valueOf(""))) {
                    editName.setError("Enter Name");
                    return;
                }
                if(email.equals(String.valueOf(""))) {
                    editEmail.setError("Enter Email");
                    return;
                }
                if(course.equals(String.valueOf(""))) {
                    editCC.setError("Enter Courses");
                    return;
                }


                boolean isInserted=
myDB.insertData(editTextId.getText().toString(),editName.getText().toStrin
g(), editEmail.getText().toString(), editCC.getText().toString());
                if (isInserted == true) {
                    Toast.makeText(Main2Activity.this, "Data Inserted",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(Main2Activity.this, "Wrong Entries",
Toast.LENGTH_SHORT).show();
                }
                editTextId.setText("");
                editEmail.setText("");
                editName.setText("");
                editCC.setText("");
                closeKeyboard();

                //Toast.makeText(MainActivity.this, "test",
Toast.LENGTH_SHORT).show();
            }
        });
    }
    public void getData() {
        buttonGetData.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String id = editTextId.getText().toString();

                if(id.equals(String.valueOf(""))) {
                    editTextId.setError("Enter ID");
                    return;
                }
                Cursor cursor = myDB.getData(id);
                String data = null;

                if(cursor.moveToNext()) {
                    data = "ID: "+ cursor.getString(0) +"\n"+
                            "Name : "+ cursor.getString(1) +"\n"+
                            "Email: "+ cursor.getString(2) +"\n"+
                            "Courses: "+ cursor.getString(3) +"\n";
                }
                showMessage("Data: ",data);
                editTextId.setText("");
                editEmail.setText("");
                editName.setText("");
                editCC.setText("");
                closeKeyboard();
            }

        });
    }
    public void viewAll() {
        buttonViewAll.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Cursor cursor = myDB.getAllData();

                //Small Test
                if (cursor.getCount() == 0) {
                    showMessage("Error","No Entries found in Database");
                    return;
                }

                StringBuffer buffer = new StringBuffer();

                while (cursor.moveToNext()) {
                    buffer.append("ID:"+cursor.getString(0)+"\n");
                    buffer.append("Name:"+cursor.getString(1)+"\n");
                    buffer.append("Email:"+cursor.getString(2)+"\n");
                    buffer.append("Courses:"+cursor.getString(3)+"\n\n");
                }
                showMessage("All Data", buffer.toString());
                editTextId.setText("");
                editEmail.setText("");
                editName.setText("");
                editCC.setText("");
            }
        });
    }
    public void updateData() {
        buttonUpdate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String id = editTextId.getText().toString();
                if(id.equals(String.valueOf(""))) {
                    editTextId.setError("Enter ID");
                    return;
                }

                boolean isUpadate =
myDB.updateData(editTextId.getText().toString(),
                        editName.getText().toString(),
                        editEmail.getText().toString(),
                        editCC.getText().toString());

                if (isUpadate == true) {
                    Toast.makeText(Main2Activity.this, "Data Updated
Successfully", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(Main2Activity.this, "Oops!! Data is Not
Updated", Toast.LENGTH_SHORT).show();
```

```
                }
                editTextId.setText("");
                editEmail.setText("");
                editName.setText("");
                editCC.setText("");
                closeKeyboard();
            }
        });
    }
    public void deleteData() {
        buttonDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String id = editTextId.getText().toString();
                if(id.equals(String.valueOf(""))) {
                    editTextId.setError("Enter ID");
                    return;
                }
                Integer deletedRow =
myDB.deleteData(editTextId.getText().toString());

                if (deletedRow > 0) {
                    Toast.makeText(Main2Activity.this, "Deleted
Successfully", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(Main2Activity.this, "Oops!! Delete
Unsuccessful", Toast.LENGTH_SHORT).show();
                }
                editTextId.setText("");
                editEmail.setText("");
                editName.setText("");
                editCC.setText("");
                closeKeyboard();
            }
        });
    }

    private void showMessage(String title, String message){
        AlertDialog.Builder builder= new AlertDialog.Builder(this);
        builder.create();
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }

    public void closeKeyboard() {
        View view = this.getCurrentFocus();
        if (view !=null) {
            InputMethodManager imm =
(InputMethodManager)getSystemService(Context.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
        }
    }


}
```

//activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="16dp"
android:background="#47535E"
tools:context=".Main2Activity">

<EditText
    android:id="@+id/editText_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:hint="Id"
    android:textColor="#fff"
    android:textColorHint="#EAF0F1"
    android:inputType="number" />

<EditText
    android:id="@+id/editText_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColorHint="#EAF0F1"
    android:textColor="#fff"
    android:layout_below="@id/editText_id"
    android:hint="Name"
    android:imeOptions="actionNext"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/editText_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColorHint="#EAF0F1"
    android:textColor="#fff"
    android:layout_below="@id/editText_name"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/editText_CC"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText_email"
    android:hint="Enter Courses"
    android:textColor="#fff"
    android:textColorHint="#EAF0F1"
    android:layout_marginBottom="20dp"
    android:imeOptions="actionGo" />

<Button
    android:id="@+id/button_add"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText_CC"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="2dp"
    android:background="#3498DB"
    android:gravity="center"
    android:text="Add Data"
    android:textColor="#fff" />

<Button
    android:id="@+id/button_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/button_add"
    android:layout_centerHorizontal="true"
    android:textColor="#fff"
    android:background="#10A881"
```

```
            android:layout_marginBottom="2dp"
            android:gravity="center"
            android:text="View" />

        <Button
            android:id="@+id/button_update"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/button_view"
            android:layout_centerHorizontal="true"
            android:textColor="#fff"
            android:background="#F3B431"
            android:layout_marginBottom="2dp"
            android:gravity="center"
            android:text="Update" />

        <Button
            android:id="@+id/button_delete"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/button_update"
            android:textColor="#fff"
            android:layout_centerHorizontal="true"
            android:layout_marginBottom="2dp"
            android:background="#E44236"
            android:gravity="center"
            android:text="Delete" />

        <Button
            android:id="@+id/button_viewALL"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#fff"
            android:layout_marginBottom="2dp"
            android:layout_below="@id/button_delete"
            android:layout_centerHorizontal="true"
            android:background="#E74292"
            android:gravity="center"
            android:text="View All" />

        <Button
            android:id="@+id/logout"
            android:layout_width="200dp"
            android:layout_height="40dp"
            android:layout_below="@id/button_viewALL"
            android:layout_marginLeft="170dp"
            android:layout_marginTop="10dp"
            android:gravity="center"
            android:text="Log Out" />

</RelativeLayout>
```

// AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.studentdatabase">

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".Main2Activity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

9. TESTING –

Software testing is used to test the system for finding the bugs, errors in the system and checking all components of the system are working properly or not. It is also used to fix errors and bugs.

Types of testing that we have used in the project are as follows:

9.1 GUI TESTING:

GUI testing is also known as a UI or User Interface testing. By using this type of testing, we have tested that user interface of the system. In this testing, we have checked that fonts used are visible to the user. Font color is visible and is matched with

9.2 MANUAL TESTING:

Manual testing includes testing software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and test the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user-acceptance testing.

9.3 PERFORMANCE TESTING:

It is mostly used to identify any bottlenecks or performance issues rather than finding bugs in software. There are different causes that contribute to lowering the performance of the software:

- Database transaction processing
- Data rendering

We have tested the software using all of the above options.

9.4 REGRESSION TESTING:

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation.

We have tested that no change in the new module affects the existing module in any way.

10. CONCLUSION AND FUTURE SCOPE –

10.1 Conclusion –

The Student Database is similar to the application that stores the data on a daily basis. This application is Because of the provided cause of problems. However, after extensive testing, we were able to produce a working Student Database with good results. Thanks to the SQLite database which works with java efficiently. From this, we can view all the data at a time.

10.2 Future Scope –

There is a lot of future work that can be done with this project. The user experience could be improved upon -

- Features such as downloading a database in an Excel file can be done.
- We can connect our database to Firebase so that we can manipulate this database in real-time (by the use of the internet).

10.3: Application:

- Student Database can be used in a library management system.
- Student Database can be used in a college management system.
- Student Database can be used in an attendance monitoring system.

10.4. References -

1. https://www.learncodeonline.in/ -via Hitesh Choudhary
2. https://www.geeksforgeeks.org/