ABSTRACT

TEACHER'S ASSISTANT

Android is the ideal platform for developing such an application due to the wide variety of devices it supports. This Android app for teachers support basic functionalities such as adding student to each class/department, save notes, make schedules for classes etc. The First Activity Contains a Grid view that provides the connection to basic activities. The Attendance module is added to take attendance and store it in a database. Contact provides an easier way to save communicate and send notes to students. Notes can also be associated with a subject so that it will be popped up when the teacher takes attendance for that subject. The remaindar is basically used to schedule a particular event so that teachers won't be needing another application as reminder and also it provides to remain the project and assignment deadlines, association meatings etc. This project would be so helpful for the staffs to use time perfectly.

Features Available

- Add courses
- Take attendance and keep them class wise
- Edit Student/Attendance later
- Contact
- Marks can be stored
- Add new student. View each student's attendance separately
- Remainder/Deadlines.

SOFTWARE REQUIREMENTS:

- Front end : Android Studio
- Backend : Sqlite / firebase

TEACHER'S ASSISTANT

A project submitted to the Bharathidasan University
in partial fulfillment of the requirements
for the award of the Degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

V ADITHYA
Register Number: 175114206

Under the guidance of
Dr. R. JEMIMA PRIYADARSINI, MCA., M.Phil., Ph.D.,
Associate Professor

PG DEPARTMENT OF COMPUTER SCIENCE (S.F)
BISHOP HEBER COLLEGE (AUTONOMOUS)
(Nationally Reaccredited at the 'A' Grade by NAAC with the CGPA of 3.58 out of 4)
(Recognized by UGC as "College of Excellence")
(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI-620 017

APRIL – 2020

Date:

CERTIFICATE

This is to certify that the project work entitled **"Teacher's Assistant"** is a bonafide record work done by **V ADITHYA, Register Number: 175114206** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** during the period **2017 - 2020.**

Place: **Signature of the Guide**

PG DEPARTMENT OF COMPUTER SCIENCE (S.F)

BISHOP HEBER COLLEGE (AUTONOMOUS),

(Nationally Reaccredited at the 'A' Grade by NAAC with the CGPA of 3.58 out of 4)
(Recognized by UGC as "College of Excellence")
(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI - 620017

Date:

Course Title: Project **Course Code: U15CS6PJ**

CHAPTER 1

PROJECT DESCRIPTION

1.1 INDROTUCTION

Android is the ideal platform for developing such an application due to the wide variety of devices it supports. Now, every person in world living with independent mobile phones. So that teachers from Bishop Heber College, having notes and performing activities in their mobile phones. This Android app (Teacher's Assistant) for teachers of Bishop Heber College support basic functionalities such as adding student to each class/department, make schedules for classes etc.

The software is majorly built using Java and Android-XML. The IDE for Android development is provided by Google and JetBrains - Android Studio. Android API provided by Android Studio IDE through SDK tools has been used. SQLite has been used as Database and Firebase has been used for authentication, receiving feedback from users and tracking the application usage. Gradle build tool has been as the build automation software.

1.2 EXISTING SYSTEM

In existing system the teachers from Bishop Heber College does not have any application to assist them. So staffs do everything manually from noting down attendance in paper, staffs

also can easily forgot that what assignment/project which is given to the student etc. So this may be the serious disadvantage for the staffs.

DISADVANTAGES

- Loads of paper work.
- Student marks are noted in sheets.
- Hard to remember details.
- Takes lots of time.

1.3 PROPOSED SYSTEM

Android is the ideal platform for developing such an application due to the wide variety of devices it supports. This Android app for teachers support basic functionalities such as adding student to each class/department, setting remainder and deadlines, adding marks for each students etc. The first activity contains the name of the college and logo design which is a splash screen. The Attendance module is added to take attendance and store it in a sqlite database. Then the login and signup activity is authenticated by the Google Firebase and their details are stored in realtime firebase database. The storage and downloads are viewed in overview firebase. Then the functionalites can be used.

ADVANTAGES

- Take attendance and keep them class wise
- Add number of students for each courses created. View attendance by selecting date.
- Marks are noted for odd semester and even semester.
- Remainder and deadlines can be created for the future use.
- Two contact email address can be stored, one for head and another for student and easy to share information.
- Staffs can view and search the student details
- Staffs can backup their data which will store in database.
- Staffs can give feedback and rate the app.
- Safe to use because of google firebase authentication.

CHAPTER 2

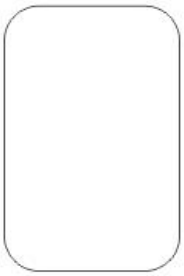
LOGICAL DEVELOPMENT

2.1. DFDs

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in.

Data flow Symbols:

Symbol	**Description**
	An **entity**. A source of data or a destination for data.

	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers. A context diagram gives an overview and it is the highest level in a data flow diagram, containing only one process representing the entire system. It should be split into major processes which give greater detail and each major process may further split to give more detail. Level 0 DFD must balance with the context diagram it describes. Input going into a process is different from outputs leaving the process. Data stores are first shown at this level.

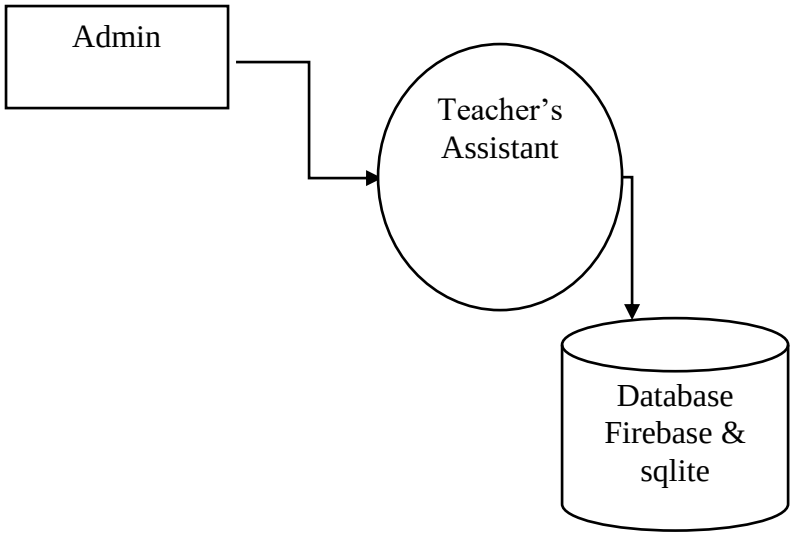
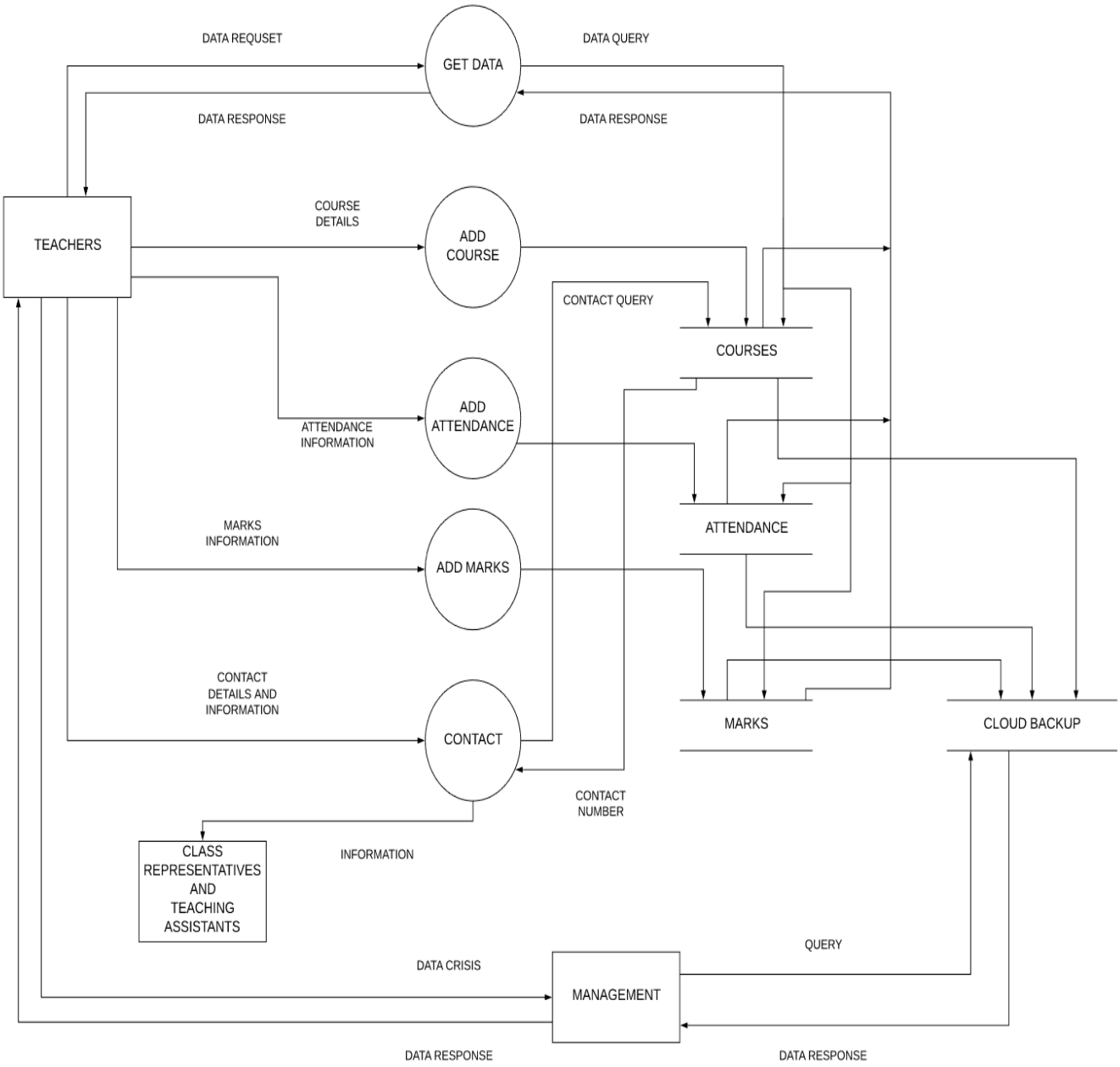


Fig 2.1.1 level 0-DFD

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. Level 1 - interaction between 2 different business applications. This is primarily used to explain the process to business and tech leads, QA leads. As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram.



2.2 ARCHITECTURAL DESIGN

Structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.

An architecture diagram is a graphical representation of a set of concepts that are part of architecture, including their principles, elements and components. Architecture diagram can help system designers and developers visualize the high-level, overall structure of their system or application, in order to ensure the system meets their users' needs. Using architecture diagram, you can also describe patterns that are used throughout the design. It's somewhat like a blueprint that you use as a guide, so that you and your colleagues can discuss, improve and follow.

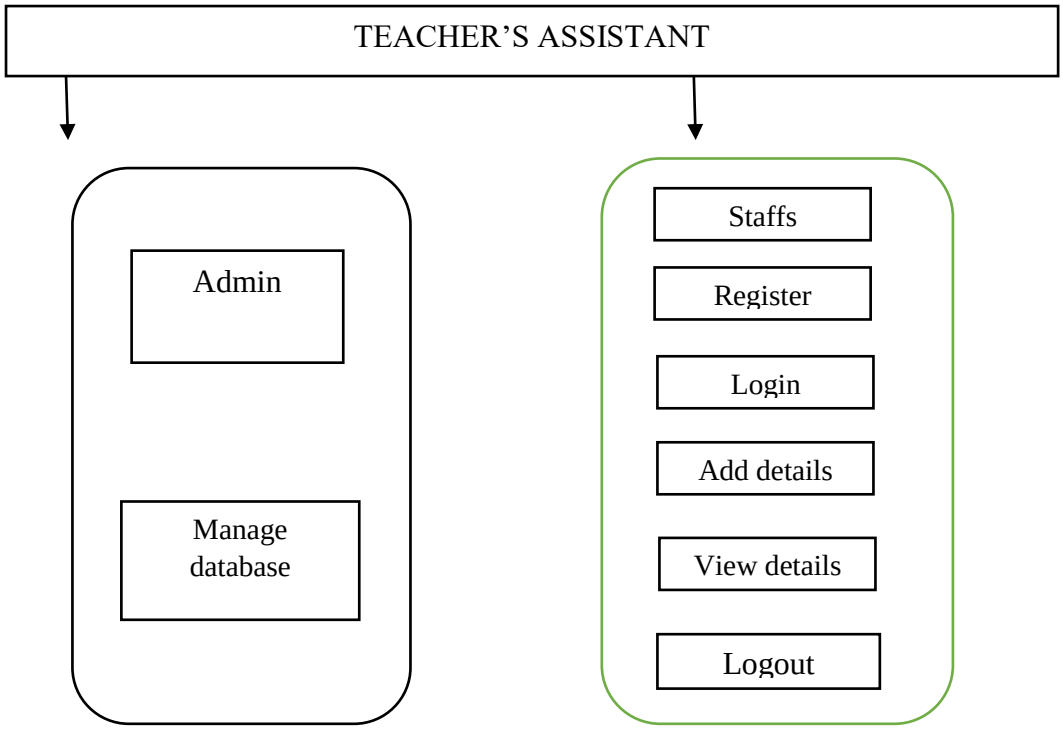


Fig 2.2.1 System Architecture

CHAPTER 3

DATABASE DESIGN

3.1 DATA DICTIONARY

Data Dictionaries are an integral component of analysis, since data flow diagram by him or she does not fully describe the subjects of the investigation. A data dictionary is a catalog of the element in as system. This element centers on data and the way are structured to meet user‘s requirements and needs. The major elements are dataflow, data stores and processes. Data dictionary stores details and description of these elements. It is developed during data analysis and assists analysis involved in determining the system.

3.2 TABLE DESIGN

3.2.1. Firebase

UserHelperClass:

FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINTS
name	String	Name of the staff	Not null
Username	String	Username	Primary Key
Email	String	Email	Not null
phoneNo	Integer	Phone Number	Not null
password	String	Password	Not null

3.2.2 Sqlite

Course.db:

Table Name: Courses

FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINTS
courseCode	String	Course Code	Primary Key
courseName	String	Course name	Not null

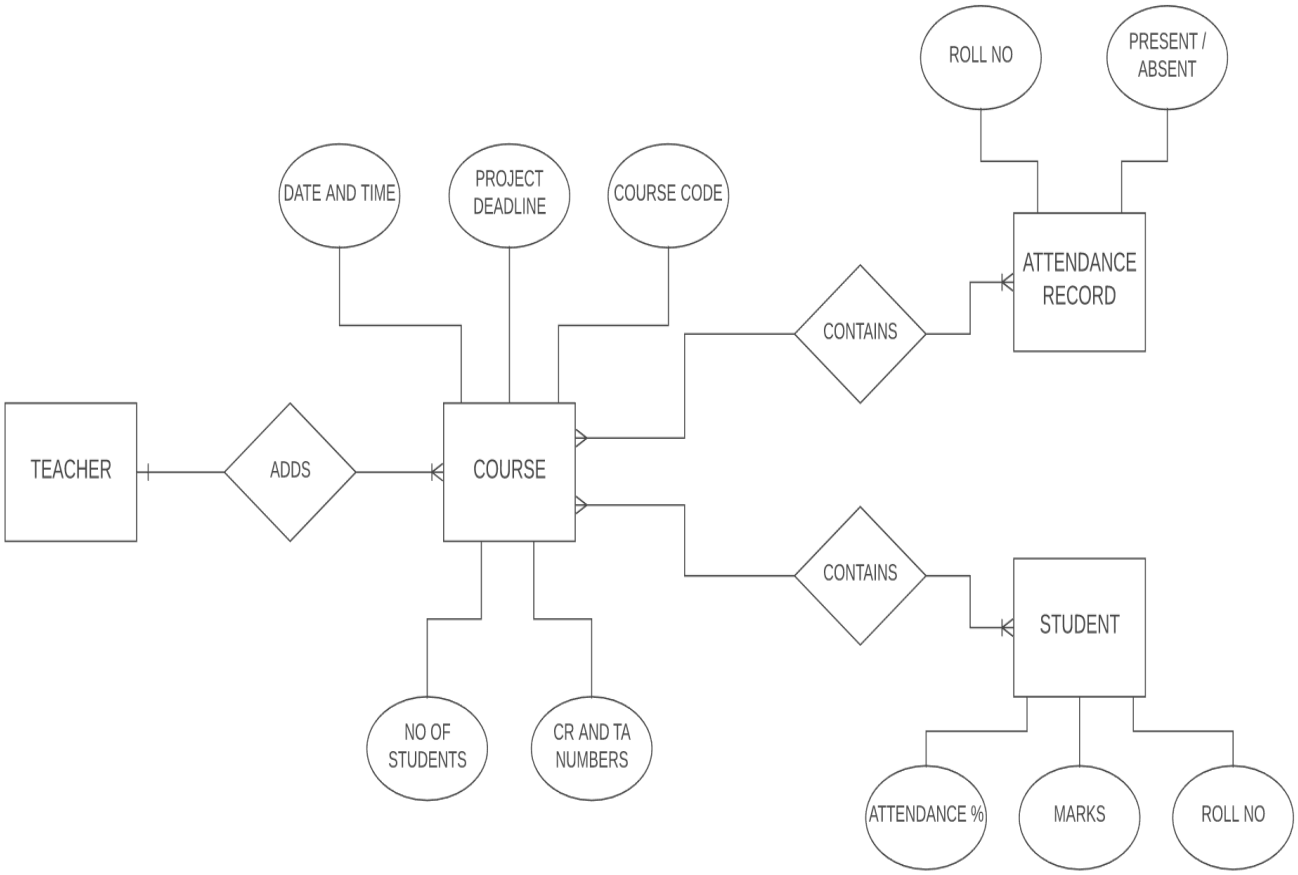
studentCount	Integer	No.of students	Not null
dayCount	Integer	No.of days	Not null
emailCr	String	Email contact	Not null
emailTa	String	Email contact	Not null

Table Name: Students

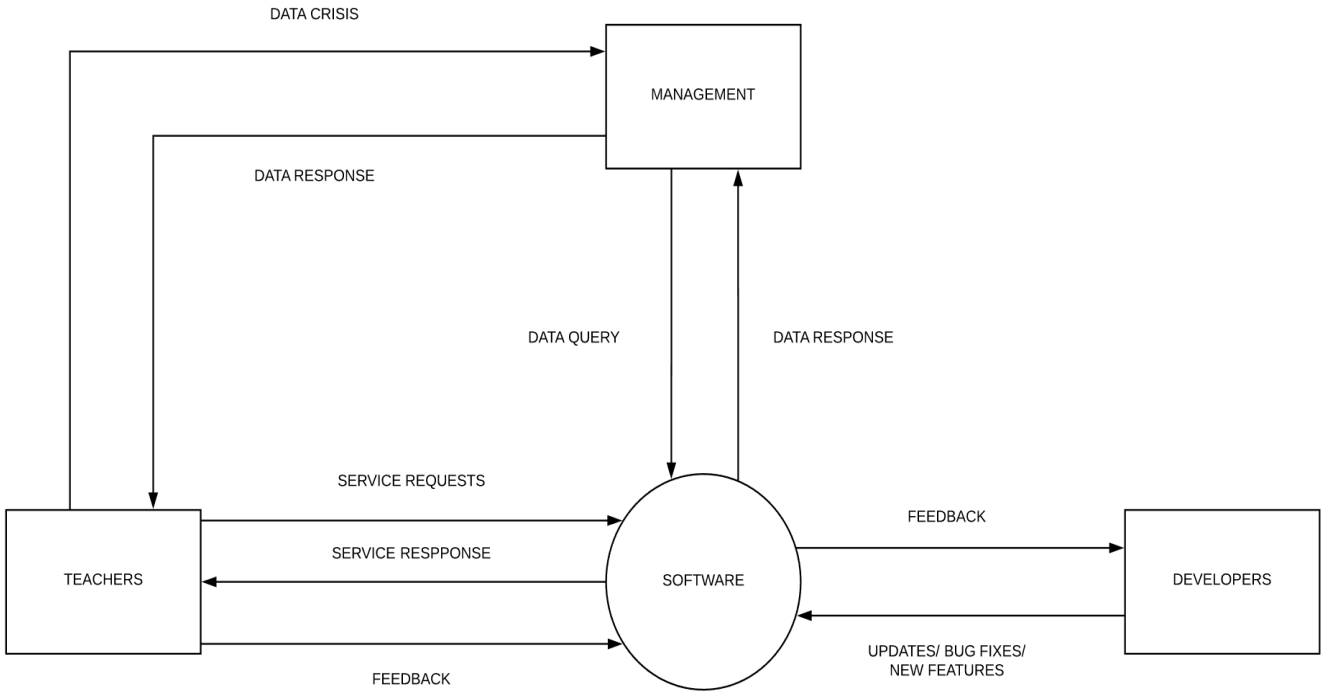
FIELD NAME	**DATA TYPE**	**DESCRIPTION**	**CONSTRAINTS**
id	Integer	**S**tudent ID	Not null
coursecode	String	Course Code	Not null
insem	Float	Even sem marks	Not null
endsem	Float	Odd sem marks	Not null
dates	String	Date	Not null

3.3 RELATIONSHIP DIAGRAM

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database.



3.4. CONTEXT DIAGRAM



CHAPTER 4

PROGRAM DESIGN

4.1 MODULES

MODULES

1. Staff
2. Admin

MODULES DESCRIPTION

Staff Registration:

Staff will register with their details to register. They will be provided with their name, email, mobile number and so on.

Staff login:

Staff login module user will be provided with a username and password where they can login and do the process they want to do.

Add details:

The staff will add the details. They will add their details and student details to make the access and work easy.

View details:

The details added by the staff will be viewed by them for the later use. They will view in case of any doubts to be clarified regarding the details added.

Interact:

The Staff and the admin will interact if any doubts which are not clarified from the admin side. Admin will interact with the staff to clear their doubts

Admin:

Admin is a head of the Teacher's Assistant android app, they view and maintain all the data in the database. Admin can check the daily usage of the application and maintain the provided authentication.

CHAPTER 5

TESTING

5.1 TESTING

- **Testing**

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

5.1.1 Types of Testing:

- **System testing**

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

Testing measures that could be followed include:

- Applying functional tests to determine whether the test criteria have been met
- Applying qualitative assessments to determine whether the test criteria have been met.
- Conducting tests in “laboratory” conditions and conducting tests in a variety of “real life” conditions.

- Conducting tests over an extended period of time to ensure systems can perform consistently.
- Conducting "load tests", simulating as close as possible likely conditions while using or exceeding the amounts of data that can be expected to be handled in an actual situation.

Test measures for hardware may include:

- Applying "non-operating" tests to ensure that equipment can stand up to expected levels of physical handling.
- Testing "hard wired" code in hardware (firmware) to ensure its logical correctness and that appropriate standards are followed.

Tests for software components also include:

- Testing all programs to ensure its logical correctness and that appropriate design, development and implementation standards have been followed.
- Conducting "load tests", simulating as close as possible a variety of "real life" conditions using or exceeding the amounts of data that could be expected in an actual situation.
- Verifying that integrity of data is maintained throughout its required manipulation.

- **Unit testing**

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

- **Integration testing**

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product

is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

CHAPTER 6
CONCLUSION

Thus staffs manual works decreases using the developed web application. Thus the android application is developed by Android Studio with the front end and SQLite & Firebase in back end. Such that the data inserted can be easily stored and retrieved using the MySQL database. Staffs can retrieve the data from the database for their concern and doubts. The Future enhancement will focuses on increase the accuracy of plagiarism check, functionall requiremnts halted for user feedback shall be completed, focus on percentage calculation and some future enhancemnts. Also, it can be extended to add more details about the staff and the student and comunication will be developed between student and staffs. Since, it will be more convenient to the staffs and the students. This development made a boon for the staffs because their manual work decreases and it is easy to access.

CHAPTER 7

REFERENCES

1. https://www.javatpoint.com/android-tutorial
2. https://firebase.google.com/docs/database/?&gclid=CjwKCAiA98TxBRBtEiwAVRLqu6VmmlcLNPea2DXMGzV06-ZDOWHAcs0ey0ZtedCdP6hwusq06Zmr8xoCdp4QAvD_BwE
3. https://www.youtube.com/channel/UCnKhcV7frITmrYbIU5MrMZw

CHAPTER 8

APPENDIX

8.1 Source Code:

JAVA:

1. Adapters

a.AddAtttendanceAdapter

```
package com.adi.project.adapters;

import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CheckBox;
import android.widget.TextView;

import com.adi.project.R;
import com.adi.project.fragments.AddAttendanceFragment;
import com.adi.project.fragments.AddAttendanceFragment.StudentAttendance;

import java.util.List;

public class AddAttendanceAdapter extends
RecyclerView.Adapter<AddAttendanceAdapter.StudentAttendanceViewHolder> {

    private List<StudentAttendance> studentAttendancelist;

    public AddAttendanceAdapter(List<StudentAttendance> studentAttendancelist) {
        this.studentAttendancelist = studentAttendancelist;
    }

    @Override
    public StudentAttendanceViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_add_attendance,
parent, false);
        return new StudentAttendanceViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(StudentAttendanceViewHolder holder, int position)
{
        // getting item from array list
        AddAttendanceFragment.StudentAttendance studentAttendance =
studentAttendancelist.get(position);
```

```
        // set attributes
        holder.textView.setText(String.valueOf(studentAttendance.getId()));
        holder.checkBox.setChecked(studentAttendance.isChecked());
        holder.checkBox.setTag(studentAttendance);
    }

    @Override
    public int getItemCount() {
        return studentAttendancelist.size();
    }

    /**
     * Holds child views for one row.
     */
    static class StudentAttendanceViewHolder extends RecyclerView.ViewHolder {
        private CheckBox checkBox;
        private TextView textView;

        StudentAttendanceViewHolder(View itemView) {
            super(itemView);
            checkBox = itemView.findViewById(R.id.check);
            textView = itemView.findViewById(R.id.studentIdText);

            // If CheckBox is toggled, update the studentAttendance obj it is
tagged with.
            checkBox.setOnClickListener(new View.OnClickListener() {
                public void onClick(View v) {   // get the checkbox
                    CheckBox cb = (CheckBox) v;
                    // get the student Attendance obj associated with that
checkbox
                    AddAttendanceFragment.StudentAttendance studentAttendance =
(AddAttendanceFragment.StudentAttendance) cb.getTag();
                    // set the student selection if check box is selected
                    studentAttendance.setChecked(cb.isChecked());
                }
            });
        }
    }

}
```

b.CourseAdapter

```
package com.adi.project.adapters;


import android.content.Context;
import android.content.Intent;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.adi.project.CourseActivity;
```

```java
import com.adi.project.R;

import java.util.List;


public class CoursesAdapter extends
RecyclerView.Adapter<CoursesAdapter.CoursesViewHolder> {

    // TODO: Add colors to course codes
//    private int[] colors ;
//    private int positionColors;

    private List<String[]> courseCodeList;
    private Context mContext;

    public CoursesAdapter(Context context, List<String[]> courseCodeList) {
        mContext = context;
        this.courseCodeList = courseCodeList;
//      colors = new int[] {R.color.brown, R.color.pink, R.color.blue,
R.color.green, R.color.orange, R.color.purple};
//      positionColors = 0;
    }

    @Override
    public CoursesViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_course, parent,
false);
        return new CoursesViewHolder(view, mContext, courseCodeList);
    }

    @Override
    public void onBindViewHolder(CoursesViewHolder holder, int position) {
        // getting item from array list
        String[] courseInfo = courseCodeList.get(position);

        // set text of view
        holder.courseCodeText.setText(courseInfo[0]);
        holder.courseNameText.setText(courseInfo[1]);

    }

    @Override
    public int getItemCount() {
        return courseCodeList.size();
    }

    static class CoursesViewHolder extends RecyclerView.ViewHolder {

        TextView courseCodeText;
        TextView courseNameText;

        CoursesViewHolder(View itemView, final Context context, final
List<String[]> courseCodeList) {
            super(itemView);
            // get view from item_course to display course code and name
            courseCodeText = itemView.findViewById(R.id.courseCodeText);
            courseNameText = itemView.findViewById(R.id.courseNameText);
```

```
            // set an OnClickListener on the list item
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    // set intent to Course Activity
                    Intent courseIntent = new Intent(context,
CourseActivity.class);
                    // get course code of item clicked
                    String[] courseInfo =
courseCodeList.get(getAdapterPosition());

                    // send course with intent
                    courseIntent.putExtra("courseCode", courseInfo[0]);
                    courseIntent.putExtra("courseName", courseInfo[1]);
                    courseIntent.putExtra("emailCr", courseInfo[2]);
                    courseIntent.putExtra("emailTa", courseInfo[3]);
                    context.startActivity(courseIntent);
                }
            });
        }
    }
}
```

c.StudentDetailsAdapter

```
package com.adi.project.adapters;


import android.content.Context;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.adi.project.R;

import java.util.ArrayList;
import java.util.List;

public class StudentDetailsAdapter extends
RecyclerView.Adapter<StudentDetailsAdapter.StudentDetailsViewHolder> {
    private List<String[]> studentDetailsList;
    private Context context;

    public StudentDetailsAdapter(Context context, ArrayList<String[]>
studentDetailsList) {
        this.context = context;
        this.studentDetailsList = studentDetailsList;
    }

    @Override
    public StudentDetailsViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
```

```
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.item_student_details, parent, false);
        return new StudentDetailsViewHolder(view);
    }

    @Override
    public void onBindViewHolder(StudentDetailsViewHolder holder, int position) {
        // get item
        String[] studentDetails = studentDetailsList.get(position);
        // set view item values
        try {
            holder.studentId.setText(studentDetails[0]);
            holder.insem.setText(studentDetails[1]);
            holder.endsem.setText(studentDetails[2]);
            holder.attendance.setText(studentDetails[3]);
        } catch (Exception e) {
            Toast.makeText(context, "No entry found", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public int getItemCount() {
        return studentDetailsList.size();
    }

    static class StudentDetailsViewHolder extends RecyclerView.ViewHolder {

        TextView studentId;
        TextView insem;
        TextView endsem;
        TextView attendance;

        public StudentDetailsViewHolder(View itemView) {
            super(itemView);

            // get view elements
            studentId = itemView.findViewById(R.id.studentId);
            insem = itemView.findViewById(R.id.insem);
            endsem = itemView.findViewById(R.id.endsem);
            attendance = itemView.findViewById(R.id.attendance);
        }
    }
}
```

d.ViewAttendanceAdapter

```
package com.adi.project.adapters;

import android.app.Activity;
import android.content.Context;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.adi.project.R;

import java.util.ArrayList;
import java.util.List;


public class ViewAttendanceAdapter extends
RecyclerView.Adapter<ViewAttendanceAdapter.ViewAttendanceViewHolder> {
    private List<String[]> attendancesList;
    private Context context;

    public ViewAttendanceAdapter(Context context, ArrayList<String[]>
attendancesList) {
        this.attendancesList = attendancesList;
        this.context = context;
    }

    @Override
    public ViewAttendanceViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.item_attendance, parent, false);
        return new ViewAttendanceViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ViewAttendanceViewHolder holder, int position) {
        // get item
        String[] attendanceList = attendancesList.get(position);

        // set view item values
        try {
            // set text and attendance status
            holder.studentId.setText(attendanceList[0]);
            holder.attendanceStatus.setText(attendanceList[1]);
            if (attendanceList[1].equals("P")) {

holder.attendanceStatus.setTextColor(context.getResources().getColor(R.color.green
));
            } else {

holder.attendanceStatus.setTextColor(context.getResources().getColor(R.color.color
Primary));
            }
        } catch (Exception e) {
            Toast.makeText(context, "No entry found", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public int getItemCount() {
        return attendancesList.size();
    }

    static class ViewAttendanceViewHolder extends RecyclerView.ViewHolder {
```

```
        TextView studentId;
        TextView attendanceStatus;

        ViewAttendanceViewHolder(View itemView) {
            super(itemView);
            // get view elements
            studentId = itemView.findViewById(R.id.studentId);
            attendanceStatus = itemView.findViewById(R.id.attendanceStatus);
        }
    }
}
```

2.Fragments

a.AddAttendanceFragment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.os.Bundle;

import androidx.annotation.Nullable;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.adi.project.CourseActivity;
import com.adi.project.R;
import com.adi.project.adapters.AddAttendanceAdapter;
import com.adi.project.helpers.DatabaseHelper;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Locale;


public class AddAttendanceFragment extends Fragment {

    // view of fragment
    View view;
    private AddAttendanceAdapter mAddAttendanceAdapter;
    private ArrayList<StudentAttendance> mStudentAttendanceList;
    private DatabaseHelper mDatabaseHelper;
    private String mCourseCode;
```

```
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity) getActivity()).initToolbar("Add Attendance");
        ((CourseActivity) getActivity()).setViewHidden(true, R.color.white);

        fillStudentList();
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view = inflater.inflate(R.layout.fragment_add_attendance, container,
false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        initList();
    }


    /**
     * Initialize add attendance list view
     */
    private void initList() {
        // Create an adapter
        mAddAttendanceAdapter = new AddAttendanceAdapter(mStudentAttendanceList);
        // Get a reference to the recycler view
        RecyclerView addAttendanceList =
view.findViewById(R.id.add_attendance_recycler_view);
        // Set layout manager
        addAttendanceList.setLayoutManager(new LinearLayoutManager(getActivity(),
                LinearLayoutManager.VERTICAL, false));
        // Set
        addAttendanceList.addItemDecoration(new
DividerItemDecoration(getActivity(), DividerItemDecoration.VERTICAL));
        // set adapter
        addAttendanceList.setAdapter(mAddAttendanceAdapter);
    }

    /**
     * Helper to fill student id list array from database
     */
    private void fillStudentList() {
        // instantiate student list
        mStudentAttendanceList = new ArrayList<>();

        // get the course code
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
```

```
//        Log.d("Bundle Check", courseInfo.getString("courseCode"));

        // get student count
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        int studentCount = mDatabaseHelper.getStudentCount(mCourseCode);
//        Log.d("DB STUDNET COUNT",
String.valueOf(mDatabaseHelper.getStudentCount(courseInfo.getString("courseCode"))
));

        // fill student list
        for (int i = 1; i <= studentCount; ++i) {
            mStudentAttendanceList.add(new StudentAttendance(String.valueOf(i),
false));
        }
    }


    /**
     * Save attendance in database
     */
    private void saveAttendance() {
        // get db instance
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        // get current date
        String date = new SimpleDateFormat("dd/MM/yyyy",
Locale.getDefault()).format(new Date());
        // update attendance
        mDatabaseHelper.updateAttendance(mCourseCode, mStudentAttendanceList,
date);
        Toast.makeText(getActivity(), "Saved!", Toast.LENGTH_SHORT).show();
    }


    /**
     * Inflate menu items into views
     *
     * @param menu     menu xml
     * @param inflater inflater obj
     */
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu, menu);
        super.onCreateOptionsMenu(menu, inflater);
    }


    /**
     * Toolbar Item selection Handler
     *
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                getActivity().onBackPressed();
                return true;
            case R.id.save:
```

```
                saveAttendance();
                getActivity().onBackPressed();
                return true;
            }
            return super.onOptionsItemSelected(item);
        }


        @Override
        public void onDestroy() {
            super.onDestroy();
            // Reset Course Activity Toolbar
            ((CourseActivity) getActivity()).initToolbar("TITLE");
            ((CourseActivity) getActivity()).setViewHidden(false, R.color.background);
        }


        /**
         * Class template for Attendance of Student
         */
        public class StudentAttendance {

            private String id = "";
            private boolean checked = false;

            public StudentAttendance(String name, boolean checked) {
                this.id = name;
                this.checked = checked;
            }

            public String getId() {
                return id;
            }

            public boolean isChecked() {
                return checked;
            }

            public void setChecked(boolean checked) {
                this.checked = checked;
            }
        }


}
```

b.AddCourseFragment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.os.Bundle;
import androidx.annotation.Nullable;
import android.util.Log;
import android.util.Patterns;
```

```
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.Toast;

import com.adi.project.CoursesActivity;
import com.adi.project.R;
import com.adi.project.models.Course;


public class AddCourseFragment extends Fragment {

    // view of fragment
    View view;
    private EditText mCourseNameEditText;
    private EditText mCourseCodeEditText;
    private EditText mStudentCountEditText;
    private EditText mEmailCrEditText;
    private EditText mEmailTaEditText;
    private Course mCourse;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Courses Activity
        setHasOptionsMenu(true);
        ((CoursesActivity)getActivity()).setDrawerLocked(true);
        ((CoursesActivity)getActivity()).initToolbar("Course Details",
R.drawable.ic_arrow_back);
        ((CoursesActivity)getActivity()).setViewHidden(true, R.color.white);
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view =  inflater.inflate(R.layout.fragment_add_course, container, false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
    }


    /**
     * Create and save Course object
     */
    private void saveCourse() {
        // get value from input fields
        mCourseNameEditText = view.findViewById(R.id.courseNameEdit);
        mCourseCodeEditText = view.findViewById(R.id.courseCodeEdit);
        mStudentCountEditText = view.findViewById(R.id.studentCountEdit);
```

```
        mEmailCrEditText = view.findViewById(R.id.emailCrEdit);
        mEmailTaEditText = view.findViewById(R.id.emailTaEdit);

        if (mCourseNameEditText.getText().toString().equals("") ||
mStudentCountEditText.getText().toString().equals("") ||
mCourseCodeEditText.getText().toString().equals("") ||
mEmailCrEditText.getText().toString().equals("") ||
mEmailTaEditText.getText().toString().equals("")) {
            Toast.makeText(getActivity(), "Please fill all fields",
Toast.LENGTH_SHORT).show();
        }
        else {
            if
(Patterns.EMAIL_ADDRESS.matcher(mEmailCrEditText.getText()).matches() &&
Patterns.EMAIL_ADDRESS.matcher(mEmailTaEditText.getText()).matches()) {
                // create Course object
                mCourse = new Course(mCourseNameEditText.getText().toString(),
mCourseCodeEditText.getText().toString(),
Integer.parseInt(mStudentCountEditText.getText().toString()),
mEmailCrEditText.getText().toString(), mEmailTaEditText.getText().toString());
                // save object in Courses Activity
                ((CoursesActivity)getActivity()).insertNewCourse(mCourse);
                // return back
                getActivity().onBackPressed();
            }
            else {
                Toast.makeText(getActivity(), "Please fill a valid email address",
Toast.LENGTH_SHORT).show();
            }
        }
    }


    /**
     * Inflate menu items into views
     * @param menu menu xml
     * @param inflater inflater obj
     */
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu, menu);
        super.onCreateOptionsMenu(menu, inflater);
    }


    /**
     * Toolbar Item selection Handler
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                getActivity().onBackPressed();
                return true;
            case R.id.save:
                // save course
                saveCourse();
```

```
                return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onResume() {
        super.onResume();

        Log.d("RESUME ADD COURSE FRAG", "IN");
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d("DESTROY ADD COURSE FRAG", "IN");
        // Reset Courses Activity Toolbar
        ((CoursesActivity)getActivity()).setDrawerLocked(false);
        ((CoursesActivity)getActivity()).initToolbar("Courses",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setFabHidden(false);
        ((CoursesActivity)getActivity()).setViewHidden(false, R.color.background);
    }
}
```

c.AddMarksFragment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.graphics.Typeface;
import android.os.Bundle;
import androidx.annotation.Nullable;
import android.text.InputType;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import com.adi.project.CourseActivity;
import com.adi.project.R;
import com.adi.project.helpers.DatabaseHelper;

import java.util.ArrayList;
```

```
public class AddMarksFragment extends Fragment{

    // view of fragment
    View view;
    private ArrayList<AddMarksFragment.StudentMarks> mStudentMarksList;
    private DatabaseHelper mDatabaseHelper;
    private String mCourseCode;
    private int mStudentCount;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity)getActivity()).initToolbar("Add Marks");
        ((CourseActivity)getActivity()).setViewHidden(true, R.color.white);

        fillStudentList();
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view =  inflater.inflate(R.layout.fragment_add_marks, container, false);

        // get table layout
        TableLayout marksTable = view.findViewById(R.id.marksTable);
        for (int i = 0; i < mStudentCount; i++) {
            // create new row
            TableRow row = new TableRow(getActivity());
            TableRow.LayoutParams layoutParams = new
TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT);
            layoutParams.setMargins(0, 0, 8, 0);
            row.setLayoutParams(layoutParams);
            // edit text for insem marks
            EditText insemEditText = new EditText(getActivity());
            insemEditText.setTag("insem" + i);
            insemEditText.setWidth(16);
            insemEditText.setInputType(InputType.TYPE_CLASS_NUMBER |
InputType.TYPE_NUMBER_FLAG_DECIMAL);
            // edit text for endsem marks
            EditText endsemEditText = new EditText(getActivity());
            endsemEditText.setTag("endsem" + i);
            endsemEditText.setWidth(16);
            endsemEditText.setInputType(InputType.TYPE_CLASS_NUMBER |
InputType.TYPE_NUMBER_FLAG_DECIMAL);
            // id
            TextView idText = new TextView(getActivity());
            idText.setText(String.valueOf(i+1));
            idText.setGravity(Gravity.CENTER_HORIZONTAL);
            idText.setTypeface(null, Typeface.BOLD);
            idText.setTextSize(18);

            row.addView(idText);
            row.addView(insemEditText);
            row.addView(endsemEditText);
```

```
            marksTable.addView(row,i);
        }
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
    }


    /**
     * Helper to fill student list array from database
     */
    private void fillStudentList() {
        // instantiate student list
        mStudentMarksList = new ArrayList<>();

        // get the course code
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
//        Log.d("Bundle Check", courseInfo.getString("courseCode"));

        // get student count
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        mStudentCount = mDatabaseHelper.getStudentCount(mCourseCode);
//        Log.d("DB STUDENT COUNT",
String.valueOf(mDatabaseHelper.getStudentCount(courseInfo.getString("courseCode"))
));

        // fill student list
        for (int i = 0; i < mStudentCount; ++i) {
            mStudentMarksList.add(new
AddMarksFragment.StudentMarks(String.valueOf(i+1)));
        }
    }


    /**
     * Save marks in database
     */
    private void saveMarks() {
        // get db instance
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        // update attendance

        // accumulate entered marks
        for(int i = 0; i < mStudentCount; i++){
            // get marks fields
            EditText insemEditText = view.findViewWithTag("insem"+i);
            EditText endsemEditText = view.findViewWithTag("endsem"+i);
            if(insemEditText.getText().toString().equals("") ||
endsemEditText.getText().toString().equals("")) {
                Toast.makeText(getActivity(), "Please enter all fields",
Toast.LENGTH_SHORT).show();
                return;
            }
            // add to list
```

```
mStudentMarksList.get(i).setInsem(Float.valueOf(insemEditText.getText().toString()
));

mStudentMarksList.get(i).setEndsem(Float.valueOf(endsemEditText.getText().toString
()));

        }
        // save in database
        mDatabaseHelper.addMarks(mCourseCode, mStudentMarksList);
        Toast.makeText(getActivity(), "Saved!", Toast.LENGTH_SHORT).show();
        getActivity().onBackPressed();
    }


    /**
     * Inflate menu items into views
     * @param menu menu xml
     * @param inflater inflater obj
     */
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu, menu);
        super.onCreateOptionsMenu(menu, inflater);
    }


    /**
     * Toolbar Item selection Handler
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                getActivity().onBackPressed();
                return true;
            case R.id.save:
                saveMarks();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Course Activity Toolbar
        ((CourseActivity)getActivity()).initToolbar("TITLE");
        ((CourseActivity)getActivity()).setViewHidden(false, R.color.background);
    }


    /**
     * Class template for Attendance of Student
     */
    public class StudentMarks {
```

```
        private String id = "";
        private float insem;
        private float endsem;

        public StudentMarks(String name) {
            this.id = name;
            this.insem = 0;
            this.endsem = 0;
        }

        public String getId() {
            return id;
        }

        public void setId(String id) {
            this.id = id;
        }

        public float getInsem() {
            return insem;
        }

        public void setInsem(float insem) {
            this.insem = insem;
        }

        public float getEndsem() {
            return endsem;
        }

        public void setEndsem(float endsem) {
            this.endsem = endsem;
        }
    }
}
```

d.AddNotificationFragment

```
package com.adi.project.fragments;

import android.app.DatePickerDialog;
import android.app.Fragment;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import androidx.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.DatePicker;
```

```
import android.widget.TimePicker;

import com.adi.project.CourseActivity;
import com.adi.project.R;

import java.util.Calendar;

public class AddNotificationFragment extends Fragment {

    View view;
    private String mCourseCode;
    private Calendar mCalendar;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity)getActivity()).initToolbar("Add Notification");
        ((CourseActivity)getActivity()).setViewHidden(true, R.color.white);

        getCourseCode();
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view =  inflater.inflate(R.layout.fragment_add_notification, container,
false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        mCalendar = Calendar.getInstance();

        DatePickerDialog.OnDateSetListener date = new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
                // set date
                mCalendar.set(Calendar.YEAR, year);
                mCalendar.set(Calendar.MONTH, monthOfYear);
                mCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);

                // set time
                TimePickerDialog timePickerDialog = new
TimePickerDialog(getActivity(),
                        new TimePickerDialog.OnTimeSetListener() {

                            @Override
                            public void onTimeSet(TimePicker view, int hourOfDay,
                                                  int minute) {
```

```
                                mCalendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
                                mCalendar.set(Calendar.MINUTE, minute);
                                saveNotification();
                            }
                        }, mCalendar.get(Calendar.HOUR_OF_DAY),
mCalendar.get(Calendar.MINUTE), false);
                timePickerDialog.show();
            }
        };

        // display the date picker dialog
        new DatePickerDialog(getActivity(), date, mCalendar
                .get(Calendar.YEAR), mCalendar.get(Calendar.MONTH),
                mCalendar.get(Calendar.DAY_OF_MONTH)).show();
    }


    /**
     * Get course code
     */
    private void getCourseCode() {
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");

    }


    /**
     * Save notifications
     */
    private void saveNotification() {

        // send intent to google calendar
        // get time in millis
        long startTime = mCalendar.getTimeInMillis();

        // send intent
        Intent intent = new Intent(Intent.ACTION_EDIT);
        intent.setType("vnd.android.cursor.item/event");
        intent.putExtra("beginTime", startTime);
        intent.putExtra("endTime", startTime+60*60*1000);
        intent.putExtra("title", mCourseCode+": Project");

        // start activity
        startActivity(intent);

    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Course Activity Toolbar
        ((CourseActivity)getActivity()).initToolbar("TITLE");
        ((CourseActivity)getActivity()).setViewHidden(false, R.color.background);
    }


}
```

e.BackupFragment

```
package com.adi.project.fragments;

import android.app.Activity;
import android.app.Fragment;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Looper;
import androidx.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;


import com.adi.project.CoursesActivity;
import com.adi.project.R;
import com.adi.project.helpers.DatabaseHelper;
import com.adi.project.models.Course;
import com.adi.project.models.Student;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.auth.FirebaseAuth;

import java.lang.ref.WeakReference;
import java.util.ArrayList;



public class BackupFragment extends Fragment {

    // view of fragment
    View view;

    private static DatabaseHelper mDatabaseHelper;
    private static LinearLayout mLayoutProgress;
    private static ImageView doneImage;

    // Firebase instances
    private static FirebaseAuth mFirebaseAuth;
    private static FirebaseFirestore mFirestoreInstance;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Courses Activity
        setHasOptionsMenu(true);
        ((CoursesActivity)getActivity()).setFabHidden(true);
        ((CoursesActivity)getActivity()).initToolbar("Backup",
```

```
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setViewHidden(true, R.color.white);

        // get required instances
        mFirebaseAuth = FirebaseAuth.getInstance();
//        Log.d("FIREBASE USER", mFirebaseAuth.getCurrentUser().getUid());
        mFirestoreInstance = FirebaseFirestore.getInstance();
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view =  inflater.inflate(R.layout.fragment_backup, container, false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        // progress loader
        mLayoutProgress = view.findViewById(R.id.layoutProgress);
        // success indicator
        doneImage = view.findViewById(R.id.done);
        // listener on button
        Button backupButton = view.findViewById(R.id.backupButton);
        backupButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                doneImage.setVisibility(View.GONE);
                mLayoutProgress.setVisibility(View.VISIBLE);
                collectDataAndBackup();
            }
        });
    }


    /**
     * Collect Data from local database and Backup data to Firestore
     */
    private void collectDataAndBackup() {
        // run task on separate thread
        new CollectDataAndBackupTask(getActivity()).execute();
    }


    /**
     * Async Task class
     */
    private static class CollectDataAndBackupTask extends AsyncTask<Void, Void,
String> {

        private WeakReference<Activity> activityReference;

        // only retain a weak reference to the activity
        CollectDataAndBackupTask(Activity context) {
            activityReference = new WeakReference<>(context);
```

```
        }

        // task
        @Override
        protected String doInBackground(Void... params) {
            if (Looper.myLooper() == null) {
                Looper.prepare();
            }
            try {
                // get cursor for tables
                Cursor coursesCursor = mDatabaseHelper.getCourses();
                Log.d("LENGTH OF TABLE",
String.valueOf(coursesCursor.getCount()));
                Cursor studentsCursor = mDatabaseHelper.getStudentInfo();

                // check if course table is present
                if (coursesCursor.moveToFirst()) {
                    int id = 0;
                    // move through the course table
                    do {
                        // get row attributes for a course and instantiate a
object for these
                        int studentCount = coursesCursor.getInt(2);
                        String courseCode = coursesCursor.getString(0);
                        Course course = new Course(coursesCursor.getString(0),
coursesCursor.getString(1), studentCount, coursesCursor.getString(3),
coursesCursor.getString(4));
                        course.setmDayCount(coursesCursor.getInt(5));

                        // fill array list of Students for a course
                        ArrayList<Student> studentList = new ArrayList<>();
                        // check if student table exists
                        if (studentsCursor.moveToFirst()) {
                            // move through the students table
                            do {
                                // check for the course code as the current course
                                if
(studentsCursor.getString(0).equals(courseCode)) {
                                    // get all students for the course
                                    for (int i = 0; i < studentCount; ++i) {
                                        Log.d("CHECK COURSE DATES",
studentsCursor.getString(3));
                                        // instantiate Student obj from row
attributes and add in the list
                                        studentList.add(new Student(i + 1,
studentsCursor.getFloat(1), studentsCursor.getFloat(2),
studentsCursor.getString(3)));
                                        studentsCursor.moveToNext();
                                    }
                                    // add list to course
                                    course.setmStudentList(studentList);
                                    // store in Firestore under the email id of
the user and doc id as per course

mFirestoreInstance.collection(mFirebaseAuth.getCurrentUser().getEmail()).document(
String.valueOf(id)).set(course);
                                    id += 1;
                                    Log.d("DONE", "IN");
                                    break;
```

```
                                }
                            } while (studentsCursor.moveToNext());
                        }
                    } while (coursesCursor.moveToNext());
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

            return "task finished";
        }

        // after success
        @Override
        protected void onPostExecute(String result) {
            Toast.makeText(activityReference.get(), "Successful!",
Toast.LENGTH_SHORT).show();
            mLayoutProgress.setVisibility(View.GONE);
            doneImage.setVisibility(View.VISIBLE);
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Courses Activity Toolbar
        ((CoursesActivity)getActivity()).initToolbar("Courses",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setFabHidden(false);
        ((CoursesActivity)getActivity()).setViewHidden(false, R.color.background);
    }

}
```

f.DocumentSimilarityFraagment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.os.Bundle;
import androidx.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.adi.project.CoursesActivity;
import com.adi.project.R;

import java.util.Iterator;
import java.util.HashMap;
import java.util.Map;
```

```
public class DocumentSimilarityFragment extends Fragment {

    // view of fragment
    View view;


    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Courses Activity
        ((CoursesActivity)getActivity()).setFabHidden(true);
        ((CoursesActivity)getActivity()).initToolbar("Document Similarity",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setViewHidden(true, R.color.white);

    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        final View view =  inflater.inflate(R.layout.fragment_document_similarity,
container, false);

        Button button = view.findViewById(R.id.similarity_button);
        button.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {

                // Fetching the two strings
                EditText et1 = view.findViewById(R.id.doc1);
                EditText et2 = view.findViewById(R.id.doc2);

                String s1 = et1.getText().toString();
                String s2 = et2.getText().toString();

                // Calculating the result
                double result = cosine_algorithm(s1, s2);

                // Displaying the result
                TextView mytxt=(TextView)
view.findViewById(R.id.display_similarity);
                mytxt.setText("Result: "+String.valueOf(result)+"%");

            }
        });

        return view;

    }


    public double cosine_algorithm(String s1, String s2) {

        // Two hash maps created for cosine similarity algorithm
        HashMap<String, Integer> map1 = new HashMap<String, Integer>();
        HashMap<String, Integer> map2 = new HashMap<String, Integer>();
        String[] splitS1 = s1.trim().split("\\s+");
```

```
        String[] splitS2 = s2.trim().split("\\s+");

        // Splitting the strings into constituents
        for( int i = 0 ; i < splitS1.length ; ++i){
            if(map1.get(splitS1[i])==null){
                map1.put(splitS1[i], 1);
            }
            else{
                map1.put(splitS1[i], map1.get(splitS1[i])+1);
            }
        }

        for( int i = 0 ; i < splitS2.length ; ++i){
            if(map2.get(splitS2[i])==null){
                map2.put(splitS2[i], 1);
            }
            else{
                map2.put(splitS2[i], map2.get(splitS2[i])+1);
            }
        }

        int numerator=0,mod1=0,mod2=0;

        // Calculating the sum of roots for all the keys
        // Iterating on one of the hash maps and finding numerator
        Iterator it = map1.entrySet().iterator();
        while(it.hasNext()){
            Map.Entry pair = (Map.Entry)it.next();
            mod1=mod1+(int)pair.getValue()*(int)pair.getValue();
            if(map2.get(pair.getKey())!=null){
                numerator=numerator+(int)pair.getValue()*map2.get(pair.getKey());
            }
            it.remove();
        }


        it = map2.entrySet().iterator();
        while (it.hasNext()) {
            Map.Entry pair = (Map.Entry)it.next();
            mod2=mod2+(int)pair.getValue()*(int)pair.getValue();
        }

        // Calculating the final result
        double result=(double)numerator/(Math.sqrt(mod1)*Math.sqrt(mod2));
        result = result*(double)100;

        int upper = (int) Math.ceil(result);
        int lower = (int) Math.floor(result);

        // Ceiling or flooring in case of integral results
        double precision=1e-6;
        if(upper-result<precision){
            return upper;
        }
        if(result-lower<precision){
            return lower;
        }

        return result;
```

```
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Courses Activity Toolbar
        ((CoursesActivity)getActivity()).initToolbar("Courses",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setFabHidden(false);
        ((CoursesActivity)getActivity()).setViewHidden(false, R.color.background);
    }

}
```

g.HelpAndFeedbackFragment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.RatingBar;

import com.adi.project.CoursesActivity;
import com.adi.project.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;


public class HelpAndFeedbackFragment extends Fragment{

    // view of fragment
    View view;
```

```
    private int mViewCheck;

    private FirebaseFirestore mFirestoreInstance;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Courses Activity
        setHasOptionsMenu(true);
        ((CoursesActivity)getActivity()).setFabHidden(true);
        ((CoursesActivity)getActivity()).initToolbar("Help and Feedback",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setViewHidden(true, R.color.white);

        mFirestoreInstance = FirebaseFirestore.getInstance();
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {

        // INTERNET CONNECTION CHECK NOT REQUIRED
        view =  inflater.inflate(R.layout.fragment_help_and_feedback, container,
false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        final RatingBar ratingBar = view.findViewById(R.id.rating);
        ratingBar.setOnRatingBarChangeListener(new
RatingBar.OnRatingBarChangeListener() {
            @Override
            public void onRatingChanged(RatingBar ratingBar, float v, boolean b) {
                Log.d("RATING CHECK", "In");
            }
        });
    }


    /**
     * Send feedback gathering values from input fields
     */
    private void sendFeedback() {

        // get input values
        RatingBar ratingBar = view.findViewById(R.id.rating);
        EditText feedbackEdit = view.findViewById(R.id.feedbackEdit);
        EditText otherFeatureEdit = view.findViewById(R.id.otherFeatEdit);

        // store in map
        Map<String, Object> user = new HashMap<>();
        user.put("rating", String.valueOf(ratingBar.getRating()));
        user.put("feedback", feedbackEdit.getText().toString());
        user.put("suggestions",otherFeatureEdit.getText().toString());
```

```
        // Add a new document with a generated ID to Firebase Firestore
        mFirestoreInstance.collection("feedbacks")
                .add(user)
                .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                    @Override
                    public void onSuccess(DocumentReference documentReference) {
                        Log.d("FEEDBACK CHECK", "DocumentSnapshot added with ID: "
+ documentReference.getId());
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        Log.w("FEEDBACK CHECK", "Error adding document", e);
                    }
                });
    }


    /**
     * Inflate menu items into views
     * @param menu menu xml
     * @param inflater inflater obj
     */
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        inflater.inflate(R.menu.menu_send, menu);
        super.onCreateOptionsMenu(menu, inflater);
    }


    /**
     * Toolbar Item selection Handler
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.send:
                sendFeedback();
                getActivity().onBackPressed();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Courses Activity Toolbar
        ((CoursesActivity)getActivity()).initToolbar("Courses",
R.drawable.ic_menu);
        ((CoursesActivity)getActivity()).setFabHidden(false);
        ((CoursesActivity)getActivity()).setViewHidden(false, R.color.background);
    }
}
```

h.SearchByStudentIDFragment

```
package com.adi.project.fragments;

import android.app.Fragment;
import android.database.Cursor;
import android.os.Bundle;
import androidx.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.adi.project.CourseActivity;
import com.adi.project.R;
import com.adi.project.helpers.DatabaseHelper;

import java.util.ArrayList;



public class SearchByStudentIdFragment extends Fragment{
    // view of fragment
    View view;
    private DatabaseHelper mDatabaseHelper;
    private String mCourseCode;
    private ArrayList<String[]> mStudentDetailsList;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity)getActivity()).initToolbar("Search");
        ((CourseActivity)getActivity()).setViewHidden(true, R.color.white);
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view =  inflater.inflate(R.layout.fragment_search_student, container,
false);

        mStudentDetailsList = new ArrayList<>();
        // get course code
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
        Log.d("COURSE CODE", mCourseCode);
```

```
        // get student count for this course
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        int studentCount = mDatabaseHelper.getStudentCount(mCourseCode);
        // get no of days attendance has been taken for this course
        int dayCount = mDatabaseHelper.getDayCount(mCourseCode);

        // get student information
        Cursor c = mDatabaseHelper.getStudentInfo();
        if (c.moveToFirst()) {
            do {
                // check for the course code
                if (c.getString(0).equals(mCourseCode)) {
                    // found course entry
                    break;
                }
            } while (c.moveToNext());
        }

        // fill mStudentsList with the information
        for (int i = 0; i < studentCount; i++) {
            // no of days attended
            String dates = c.getString(3);
            int dateCount = dates.length() - dates.replace(",", "").length();;
            Log.d("DATE COUNT", String.valueOf(dateCount));
            Log.d("DAY COUNT", String.valueOf(dayCount));
            // add info
            mStudentDetailsList.add(new String[]{String.valueOf(i + 1),
String.valueOf(c.getFloat(1)), String.valueOf(c.getFloat(2)), dayCount !=
0?String.valueOf((float)((dateCount * 100) / dayCount)):"0"});
            if(!c.moveToNext()){break;}
        }
        c.close();


        Button searchButton = view.findViewById(R.id.searchButton);

        searchButton.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {


                // TODO Auto-generated method stub
                EditText searchForId = view.findViewById(R.id.searchForId);
                String studentIdObtained = searchForId.getText().toString();

                int i, flag=0;

                for( i = 0; i < mStudentDetailsList.size(); ++i){

                    if(mStudentDetailsList.get(i)[0].equals(studentIdObtained)){

                        flag=1;

                        TextView studentId = view.findViewById(R.id.studentId);
                        TextView insemMarks = view.findViewById(R.id.insemMarks);
                        TextView endsemMarks =
view.findViewById(R.id.endsemMarks);
                        TextView attendancePercent =
view.findViewById(R.id.attendancePercent);
```

```
                        studentId.setText(mStudentDetailsList.get(i)[0]);
                        insemMarks.setText(mStudentDetailsList.get(i)[1]);
                        endsemMarks.setText(mStudentDetailsList.get(i)[2]);
                        attendancePercent.setText(mStudentDetailsList.get(i)[3]);

                        break;
                    }
                }

                if(flag==0){
                    Toast.makeText(getActivity(),"Invalid
Input!",Toast.LENGTH_SHORT).show();
                }
            }
        });
        // return inflated view
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

    }

    /**
     * Toolbar Item selection Handler
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                getActivity().onBackPressed();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Course Activity Toolbar
        ((CourseActivity)getActivity()).initToolbar("TITLE");
        ((CourseActivity)getActivity()).setViewHidden(false, R.color.background);
    }
}
```

i.StudentDetailsFragment

```java
package com.adi.project.fragments;

import android.app.Fragment;
import android.database.Cursor;
import android.os.Bundle;
import androidx.annotation.Nullable;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;


import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;

import com.adi.project.CourseActivity;
import com.adi.project.R;
import com.adi.project.adapters.StudentDetailsAdapter;
import com.adi.project.helpers.DatabaseHelper;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class StudentDetailsFragment extends Fragment {


    // view of fragment
    View view;
    private DatabaseHelper mDatabaseHelper;
    private String mCourseCode;
    private ArrayList<String[]> mStudentDetailsList;
    StudentDetailsAdapter mStudentDetailsAdapter;


    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity) getActivity()).initToolbar("Student Details");
        ((CourseActivity) getActivity()).setViewHidden(true, R.color.white);
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view = inflater.inflate(R.layout.fragment_student_details, container,
false);

        mStudentDetailsList = new ArrayList<>();
        // get course code
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
```

```
        Log.d("COURSE CODE", mCourseCode);

        // get student count for this course
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        int studentCount = mDatabaseHelper.getStudentCount(mCourseCode);
        // get no of days attendance has been taken for this course
        int dayCount = mDatabaseHelper.getDayCount(mCourseCode);

        // get student information
        Cursor c = mDatabaseHelper.getStudentInfo();
        if (c.moveToFirst()) {
            do {
                // check for the course code
                if (c.getString(0).equals(mCourseCode)) {
                    // found course entry
                    break;
                }
            } while (c.moveToNext());
        }

        // fill mStudentsList with the information
        for (int i = 0; i < studentCount; i++) {
            // no of days attended
            String dates = c.getString(3);
            int dateCount = dates.length() - dates.replace(",", "").length();
            ;
            Log.d("DATE COUNT", String.valueOf(dateCount));
            Log.d("DAY COUNT", String.valueOf(dayCount));
            // add info
            mStudentDetailsList.add(new String[]{String.valueOf(i + 1),
String.valueOf(c.getFloat(1)), String.valueOf(c.getFloat(2)), dayCount != 0 ?
String.valueOf((float) ((dateCount * 100) / dayCount)) : "0"});
            if (!c.moveToNext()) {
                break;
            }
        }
        c.close();

        // return inflated view
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        mStudentDetailsAdapter = new StudentDetailsAdapter(getActivity(),
mStudentDetailsList);
        RecyclerView studentDetailsList =
view.findViewById(R.id.list_student_details);
        studentDetailsList.setAdapter(mStudentDetailsAdapter);
        studentDetailsList.setLayoutManager(new LinearLayoutManager(getActivity(),
                LinearLayoutManager.VERTICAL, false));
        studentDetailsList.addItemDecoration(new
DividerItemDecoration(getActivity(),
                DividerItemDecoration.VERTICAL));
    }
```

```java
/**
 * Sort mStudentDetailsList by marks
 */
private void sortByMarks() {
    Collections.sort(mStudentDetailsList, new Comparator<String[]>() {
        @Override
        public int compare(String[] s1, String[] s2) {
            return Float.valueOf(s1[2]).compareTo(Float.valueOf(s2[2]));
        }
    });
    mStudentDetailsAdapter.notifyDataSetChanged();
}


/**
 * Sort mStudentDetailsList by attendance %
 */
private void sortByAttendance() {
    Collections.sort(mStudentDetailsList, new Comparator<String[]>() {
        @Override
        public int compare(String[] s1, String[] s2) {
            return s2[3].compareTo(s1[3]);
        }
    });
    mStudentDetailsAdapter.notifyDataSetChanged();
}


/**
 * Inflate menu items into views
 *
 * @param menu     menu xml
 * @param inflater inflater obj
 */
@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.menu_overflow, menu);
    super.onCreateOptionsMenu(menu, inflater);
}


/**
 * Toolbar Item selection Handler
 *
 * @param item in toolbar
 * @return true: to hold and exit, false: to fall through
 */
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            getActivity().onBackPressed();
            return true;
        case R.id.sort_marks:
            sortByMarks();
            return true;
        case R.id.sort_attendance:
            sortByAttendance();
```

```
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Course Activity Toolbar
        ((CourseActivity) getActivity()).initToolbar("TITLE");
        ((CourseActivity) getActivity()).setViewHidden(false, R.color.background);
    }
}
```

j.ViewAttendanceFragment

```
package com.adi.project.fragments;

import android.app.DatePickerDialog;
import android.app.Fragment;
import android.database.Cursor;
import android.os.Bundle;
import androidx.annotation.Nullable;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.DatePicker;

import com.adi.project.CourseActivity;
import com.adi.project.R;
import com.adi.project.adapters.ViewAttendanceAdapter;
import com.adi.project.helpers.DatabaseHelper;

import java.util.ArrayList;
import java.util.Calendar;


public class ViewAttendanceFragment extends Fragment {

    // view of fragment
    View view;
    // course code
    private String mCourseCode;
    private DatabaseHelper mDatabaseHelper;
    private ArrayList<String[]> mAttendanceList;

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        // configure Toolbar in Course Activity
        setHasOptionsMenu(true);
        ((CourseActivity) getActivity()).initToolbar("Attendance");
        ((CourseActivity) getActivity()).setViewHidden(true, R.color.white);
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        view = inflater.inflate(R.layout.fragment_view_attendance, container,
false);
        return view;
    }


    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        // get calendar instance
        final Calendar myCalendar = Calendar.getInstance();

        // TODO: Keep the dialog persistent and make a call to backPressed of
Parent when cancel is pressed
        // set date picker dialog listener for selection of date
        DatePickerDialog.OnDateSetListener date = new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
                // set date
                myCalendar.set(Calendar.YEAR, year);
                myCalendar.set(Calendar.MONTH, monthOfYear);
                myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
                Log.d("DATE PICKER", String.valueOf(year));
                Log.d("DATE PICKER", String.valueOf(monthOfYear));
                Log.d("DATE PICKER", String.valueOf(dayOfMonth));

                String month = "";
                if (monthOfYear < 10) {
                    month = "0" + String.valueOf(monthOfYear + 1);
                }

                displayAttendanceForTheDay(String.valueOf(dayOfMonth) + '/' +
month + '/' + String.valueOf(year));
            }
        };

        // display the date picker dialog
        new DatePickerDialog(getActivity(), date, myCalendar
                .get(Calendar.YEAR), myCalendar.get(Calendar.MONTH),
                myCalendar.get(Calendar.DAY_OF_MONTH)).show();
    }


    /**
     * Display Attendance for the day
     *
```

```
     * @param date date picked
     */
    private void displayAttendanceForTheDay(String date) {
        Log.d("SELECTED DATE", date);
        // get course code
        Bundle courseInfo = getActivity().getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
        Log.d("COURSE CODE", mCourseCode);

        // initialize attendance list
        mAttendanceList = new ArrayList<>();

        // get student count
        mDatabaseHelper = DatabaseHelper.getInstance(getActivity());
        int studentCount = mDatabaseHelper.getStudentCount(mCourseCode);
        Log.d("STUDENT COUNT", String.valueOf(studentCount));

        // get attendance dates and course code from database
        Cursor c = mDatabaseHelper.getStudentAttendance();
        if (c.moveToFirst()) {
            do {
                // check for the course code
                if (c.getString(0).equals(mCourseCode)) {
                    // found course entry
                    break;
                }
            } while (c.moveToNext());
        }

        // fil the attendance list
        Log.d("CHECK ATT", "IN");
        for (int i = 0; i < studentCount; i++) {
            Log.d("ATTENDANCE DATES", c.getString(1));
            boolean isFound = c.getString(1).contains(date);
            mAttendanceList.add(new String[]{String.valueOf(i + 1), isFound ? "P"
: "A"});
            if (!c.moveToNext()) {
                break;
            }
        }
        c.close();

        // set adapter and init recycler view
        ViewAttendanceAdapter viewAttendanceAdapter = new
ViewAttendanceAdapter(getActivity(), mAttendanceList);
        RecyclerView attendanceRecyclerView =
view.findViewById(R.id.view_attendance_list);
        attendanceRecyclerView.setAdapter(viewAttendanceAdapter);
        attendanceRecyclerView.setLayoutManager(new
LinearLayoutManager(getActivity(),
                LinearLayoutManager.VERTICAL, false));
        attendanceRecyclerView.addItemDecoration(new
DividerItemDecoration(getActivity(),
                DividerItemDecoration.VERTICAL));
    }


    /**
     * Toolbar Item selection Handler
```

```
     *
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                getActivity().onBackPressed();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        // Reset Course Activity Toolbar
        ((CourseActivity) getActivity()).initToolbar("TITLE");
        ((CourseActivity) getActivity()).setViewHidden(false, R.color.background);
    }
}
```

3.Models

a.CourseActivity

```
package com.adi.project;

import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.ContentUris;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.CalendarContract;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.widget.Toolbar;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;

import androidx.appcompat.app.AppCompatActivity;

import com.adi.project.fragments.AddAttendanceFragment;
import com.adi.project.fragments.AddMarksFragment;
import com.adi.project.fragments.AddNotificationFragment;
import com.adi.project.fragments.SearchByStudentIdFragment;
import com.adi.project.fragments.StudentDetailsFragment;
import com.adi.project.fragments.ViewAttendanceFragment;
import com.adi.project.interfaces.CourseViewInterface;
```

```
public class CourseActivity extends AppCompatActivity implements
CourseViewInterface {

    private String mCourseCode;
    private String mCrEmail;
    private String mTaEmail;
    private String mCourseName;

    private static final String EMAIL_BODY = "\n\nSent from: Course Assistant";

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

        // Intent bundle for getting extra info
        Bundle courseInfo = getIntent().getExtras();
        mCourseCode = courseInfo.getString("courseCode");
        mCrEmail = courseInfo.getString("emailCr");
        mTaEmail = courseInfo.getString("emailTa");
        mCourseName = courseInfo.getString("courseName");

        // init methods
        initToolbar(mCourseCode);
        initButtons();
    }


    /**
     * Set toolbar details
     * @param title title for toolbar
     */
    @Override
    public void initToolbar(String title) {
        // Add toolbar support
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar actionbar = getSupportActionBar();
        // add home button in toolbar
        actionbar.setDisplayHomeAsUpEnabled(true);
        actionbar.setDisplayShowHomeEnabled(true);
        // set title
        actionbar.setTitle(title);
    }


    /**
     * Hide activity view
     * @param enabled true: hide, false: show
     */
    @Override
    public void setViewHidden(boolean enabled, int color) {
        LinearLayout courseView = findViewById(R.id.course_view);
        LinearLayout l = findViewById(R.id.layout_course);
        l.setBackgroundColor(getResources().getColor(color));
        if (enabled) {
            courseView.setVisibility(View.GONE);
        }
```

```
        else {
            courseView.setVisibility(View.VISIBLE);
            initToolbar(mCourseCode);
        }
    }


    /**
     * initialize buttons in the cards of view
     */
    private void initButtons() {
        // TODO: Remove redundant FragmentManager objects in each listener
        // add attendance button
        Button addAttendance = findViewById(R.id.addAttendance);
        addAttendance.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new fragment
                AddAttendanceFragment addAttendanceFragment = new
AddAttendanceFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();
                transaction.add(R.id.fragContainer, addAttendanceFragment, "Add
Attendance Fragment");
                // add this fragment to stack
                transaction.addToBackStack("Add Attendance Fragment");
                // commit this transaction
                transaction.commit();
            }
        });

        // view attendance button
        Button viewAttendance = findViewById(R.id.viewAttendance);
        viewAttendance.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new fragment
                ViewAttendanceFragment viewAttendanceFragment = new
ViewAttendanceFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();
                transaction.add(R.id.fragContainer, viewAttendanceFragment, "View
Attendance Fragment");
                // add this fragment to stack
                transaction.addToBackStack("View Attendance Fragment");
                // commit this transaction
                transaction.commit();
            }
        });


        // view student details button
        Button viewStudentDetails = findViewById(R.id.viewStudentDetails);
        viewStudentDetails.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {
                // Create a new fragment
                StudentDetailsFragment studentDetailsFragment = new
StudentDetailsFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();
                transaction.add(R.id.fragContainer, studentDetailsFragment,
"Student Details Fragment");
                // add this fragment to stack
                transaction.addToBackStack("Student Details Fragment");
                transaction.commit();
            }
        });

        // search for a specific student's details
        Button searchByStudentId = findViewById(R.id.searchByStudentId);
        searchByStudentId.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new fragment
                SearchByStudentIdFragment searchByStudentIdFragment = new
SearchByStudentIdFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();
                transaction.add(R.id.fragContainer, searchByStudentIdFragment,
"Search By Student ID Fragment");
                // add this fragment to stack
                transaction.addToBackStack("Search By Student ID Fragment");
                transaction.commit();
            }
        });

        // add marks
        Button addMarks = findViewById(R.id.addMarks);
        addMarks.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new fragment
                AddMarksFragment addMarksFragment = new AddMarksFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();

                transaction.add(R.id.fragContainer, addMarksFragment, "Add Marks
Fragment");
                // add this fragment to stack
                transaction.addToBackStack("Add Marks Fragment");
                // commit this transaction
                transaction.commit();
            }
        });

        // add notification
        Button addNotification = findViewById(R.id.addNotification);
```

```
        addNotification.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Create a new fragment
                AddNotificationFragment addNotificationFragment = new
AddNotificationFragment();
                // get transaction manager
                FragmentManager manager = getFragmentManager();
                // start transaction
                FragmentTransaction transaction = manager.beginTransaction();

                transaction.add(R.id.fragContainer, addNotificationFragment, "Add
Notification Fragment");
                // add this fragment to stack
                transaction.addToBackStack("Add Notification Fragment");
                // commit this transaction
                transaction.commit();
            }
        });

        // view notification
        Button viewNotification = findViewById(R.id.viewNotification);
        viewNotification.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                long startMillis = System.currentTimeMillis();
                Uri.Builder builder = CalendarContract.CONTENT_URI.buildUpon();
                builder.appendPath("time");
                ContentUris.appendId(builder, startMillis);
                Intent intent;
                intent = new Intent(Intent.ACTION_VIEW).setData(builder.build());
                startActivity(intent);
            }
        });
    }


    // methods to contact CR and TA
    // TODO: Limit send options to only email apps
    public void contactMailCr(View v) {
        Intent email = new Intent(android.content.Intent.ACTION_SEND);

        email.setType("text/plain");
        email.putExtra(android.content.Intent.EXTRA_EMAIL, new
String[]{mCrEmail});
        email.putExtra(android.content.Intent.EXTRA_SUBJECT, mCourseCode + ": " +
mCourseName);
        email.putExtra(android.content.Intent.EXTRA_TEXT, EMAIL_BODY);

        startActivity(Intent.createChooser(email, "Send mail"));
    }
    public void contactMailTa(View v) {
        Intent email = new Intent(android.content.Intent.ACTION_SEND);
        email.setType("plain/text");
        email.putExtra(android.content.Intent.EXTRA_EMAIL, new
String[]{mTaEmail});
        email.putExtra(android.content.Intent.EXTRA_SUBJECT, mCourseCode + ": " +
mCourseName);
        email.putExtra(android.content.Intent.EXTRA_TEXT, EMAIL_BODY);
```

```
        startActivity(Intent.createChooser(email, "Send mail"));
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                onBackPressed();
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    @Override
    public void onBackPressed() {

        if(getFragmentManager().getBackStackEntryCount() > 0) {
            getFragmentManager().popBackStack();
        }
        else {
            super.onBackPressed();
            finish();
        }
    }
}
```

b.CoursesActivity

```
package com.adi.project;

import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.appcompat.app.ActionBar;
import androidx.recyclerview.widget.RecyclerView;
import androidx.appcompat.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Toast;

import com.adi.project.adapters.CoursesAdapter;
import com.adi.project.fragments.AddCourseFragment;
```

```java
import com.adi.project.fragments.BackupFragment;
import com.adi.project.fragments.DocumentSimilarityFragment;
import com.adi.project.fragments.HelpAndFeedbackFragment;
import com.adi.project.helpers.DatabaseHelper;
import com.adi.project.interfaces.CoursesViewInterface;
import com.adi.project.models.Course;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.navigation.NavigationView;

import java.util.ArrayList;

public class CoursesActivity extends AppCompatActivity implements
CoursesViewInterface {

    private DrawerLayout mDrawerLayout;
    private FloatingActionButton mAddCourseFab;
    private Class mCurrentFragmentClass;
    private DatabaseHelper mDbHelper;
//    private RecyclerView mRecyclerView;

    private ArrayList<String[]> mCourseCodeList;
    private CoursesAdapter mCoursesAdapter;
    private RecyclerView mRecyclerView;

    private static final String EMAIL_BODY = "\n\nSent from: BHC Staff";

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

        // Drawer Layout instantiated
        mDrawerLayout = findViewById(R.id.drawer_layout);

        // display courses
        displayCourses();

        // initialize views
        initToolbar("Courses", R.drawable.ic_menu);
        initSideNav();
        initAddCourseFab();

    }


    /**
     * Display courses from db
     */
    private void displayCourses() {
        // course code list
        mCourseCodeList = new ArrayList<>();

        // get course column from db
        mDbHelper = DatabaseHelper.getInstance(this);
        Cursor c = mDbHelper.getCourseInfo();

        // iterate through column elements
        if (c.moveToFirst()) {
            do {
```

```
                // add to list
                mCourseCodeList.add(new String[]{c.getString(0), c.getString(1),
c.getString(2), c.getString(3)});
            } while (c.moveToNext());
        }
        c.close();

        // set custom adapter and add to recycler view
        mCoursesAdapter = new CoursesAdapter(this, mCourseCodeList);
        mRecyclerView = findViewById(R.id.courses_recycler_view);
        mRecyclerView.setAdapter(mCoursesAdapter);

        // set the layout manager to the recycler view
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this,
                LinearLayoutManager.VERTICAL, false));

//        // listener for click events on list view
 //      mRecyclerView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
 //          @Override
  //          public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
  //            // set intent to Course Activity
  //            Intent courseIntent = new Intent(CoursesActivity.this,
CourseActivity.class);
   //           // get course code of item clicked
  //            String[] courseInfo = mCourseCodeList.get(position);

//                // send course with intent
  //            courseIntent.putExtra("courseCode", courseInfo[0]);
  //          courseIntent.putExtra("courseName", courseInfo[1]);
   //            courseIntent.putExtra("emailCr", courseInfo[2]);
  //            courseIntent.putExtra("emailTa", courseInfo[3]);
  //            startActivity(courseIntent);
   // }
   //    });
    }


    /**
     * Insert new course into database and list
     *
     * @param course passed from fragment
     */
    public void insertNewCourse(Course course) {
        // add course code and name to the list
        mCourseCodeList.add(new String[]{course.getmCourseCode(),
course.getmCourseName(), course.getmEmailCr(), course.getmEmailTa()});
//        for (String member : mCourseCodeList){
//            Log.i("Member name: ", member);
//        }
        // notify adapter that list has changed
        mCoursesAdapter.notifyDataSetChanged();
        // insert course in database
        mDbHelper.insertCourse(course);

        Toast.makeText(this, "Saved!", Toast.LENGTH_SHORT).show();
    }
```

```
    /**
     * Toolbar Item selection Handler
     *
     * @param item in toolbar
     * @return true: to hold and exit, false: to fall through
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // get current fragment in activity
        Fragment currentFragment = getFragmentManager().findFragmentByTag("Add New
Course Fragment");
        // if yes then call onOptionsItemSelected of fragment first
        if (currentFragment != null &&
currentFragment.onOptionsItemSelected(item)) {
            // onOptionsItemSelected of current fragment will return true is item
is home
            return true;
        }

        // check for item selection in Toolbar
        switch (item.getItemId()) {
            case android.R.id.home:
                mDrawerLayout.openDrawer(GravityCompat.START);
                return true;
        }
        return super.onOptionsItemSelected(item);
    }


    /**
     * Set listener to Add course Floating Action Button
     * and display Add New Course Fragment
     */
    private void initAddCourseFab() {

        mAddCourseFab = findViewById(R.id.addCourseFab);
        mAddCourseFab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // if there are no previous fragments in back stack
                if (getFragmentManager().getBackStackEntryCount() == 0) {
                    // Create a new fragment
                    AddCourseFragment mAddCourseFragment = new
AddCourseFragment();
                    // get transaction manager
                    FragmentManager manager = getFragmentManager();
                    // start transaction
                    FragmentTransaction transaction = manager.beginTransaction();
                    transaction.add(R.id.fragContainer, mAddCourseFragment, "Add
New Course Fragment");
                    // add this fragment to stack
                    transaction.addToBackStack("Add New Course Fragment");
                    // commit this transaction
                    transaction.commit();

                    // set current fragment class as Add Course Fragment
                    mCurrentFragmentClass = AddCourseFragment.class;
```

```
                }
            }
        });
    }


    /**
     * Configure Toolbar params
     *
     * @param title   for toolbar
     * @param ic_home icon to be displayed for home
     */
    public void initToolbar(String title, int ic_home) {
        // Add toolbar support
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar actionbar = getSupportActionBar();
        // add home button in toolbar
        actionbar.setDisplayHomeAsUpEnabled(true);
        actionbar.setDisplayShowHomeEnabled(true);
        actionbar.setHomeAsUpIndicator(ic_home);
        // set title
        actionbar.setTitle(title);
        Log.d("Check init toolbar", title);
    }


    /**
     * Set listener for Side Nav item selection
     */
    private void initSideNav() {
        // set listener
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(
                new NavigationView.OnNavigationItemSelectedListener() {
                    @Override
                    public boolean onNavigationItemSelected(MenuItem menuItem) {
                        selectDrawerItem(menuItem);
                        return true;

                    }
                });
    }


    /**
     * Handler for nav drawer selection
     *
     * @param menuItem item clicked in nav drawer
     */
    public void selectDrawerItem(MenuItem menuItem) {
        removeAllFragments();

        mCurrentFragmentClass = null;
        // Create a new fragment and specify the fragment to show based on nav
item clicked
        Fragment fragment = null;

        // select
```

```
        switch (menuItem.getItemId()) {
            case R.id.home:
                // On selection, highlight it and quit the drawer
                menuItem.setChecked(true);
                setFabHidden(false);
                mDrawerLayout.closeDrawers();
                return;
            case R.id.similarity_check:
                mCurrentFragmentClass = DocumentSimilarityFragment.class;
                break;
            case R.id.help_feedback:
                mCurrentFragmentClass = HelpAndFeedbackFragment.class;
                break;
            case R.id.contact_us:
                // select home in nav drawer
                menuItem.setChecked(false);
                NavigationView navigation = findViewById(R.id.nav_view);
                Menu drawer_menu = navigation.getMenu();
                MenuItem m = drawer_menu.findItem(R.id.home);
                m.setChecked(true);
                mDrawerLayout.closeDrawers();
                // set intent
                contactUs();
                return;
            case R.id.request_backup:
                mCurrentFragmentClass = BackupFragment.class;
                break;
            default:
                return;
        }

        // make fragment object from class
        try {
            fragment = (Fragment) mCurrentFragmentClass.newInstance();
        } catch (Exception e) {
            e.printStackTrace();
        }

        // Insert the fragment by replacing any existing fragment
        FragmentManager fragmentManager = getFragmentManager();
        FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.fragContainer, fragment);
        transaction.addToBackStack(null);
        transaction.commit();

        // Highlight the selected item has been done by NavigationView
        menuItem.setChecked(true);
        // Set action bar title
//        setTitle(menuItem.getTitle());
        // Close the navigation drawer
        mDrawerLayout.closeDrawers();
    }


    /**
     * contact us handler
     */
    private void contactUs() {
        Intent email = new Intent(android.content.Intent.ACTION_SEND);
```

```
        email.setType("text/plain");
        email.putExtra(android.content.Intent.EXTRA_EMAIL, new
String[]{"teachersassistantadmin@gmail.com"});
        email.putExtra(android.content.Intent.EXTRA_SUBJECT, "Query");
        email.putExtra(android.content.Intent.EXTRA_TEXT, EMAIL_BODY);

        startActivity(Intent.createChooser(email, "Send mail"));
    }


    /**
     * Lock Side Nav
     *
     * @param enabled true: lock, false: unlock
     */
    @Override
    public void setDrawerLocked(boolean enabled) {
        // check if Drawer exists
        if (mDrawerLayout != null) {
            if (enabled) {

mDrawerLayout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
            } else {
                mDrawerLayout.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED);
            }
        }
    }


    /**
     * Hide and show Add Course FAB
     *
     * @param enable true: hide, false: show
     */
    @Override
    public void setFabHidden(boolean enable) {
        if (enable) {
            mAddCourseFab.hide();
        } else {
            mAddCourseFab.show();
        }
    }


    /**
     * Hide view of Courses Activity and set background color
     *
     * @param enabled true: hide, false: show
     * @param color   set background color
     */
    @Override
    public void setViewHidden(boolean enabled, int color) {
        RecyclerView l = findViewById(R.id.courses_recycler_view);
        mDrawerLayout.setBackgroundColor(getResources().getColor(color));
        if (enabled) {
            l.setVisibility(View.GONE);
        } else {
            l.setVisibility(View.VISIBLE);
```

```
        }
    }


    /**
     * Remove all fragments from back stack
     */
    private void removeAllFragments() {
        // remove all fragments
        FragmentManager fragmentManager = getFragmentManager();
        for (int i = 0; i < fragmentManager.getBackStackEntryCount(); ++i) {
            fragmentManager.popBackStack();
        }
    }


    /**
     * Decide navigation on press of back button
     */
    @Override
    public void onBackPressed() {

        // if fragments are present
        if (mCurrentFragmentClass != null) {
            Log.d("Check back", "fRAG");
            // remove all fragments in stack
            removeAllFragments();
            // no current fragment
            mCurrentFragmentClass = null;
            // select home item in nav drawer and check it
            NavigationView navigation = findViewById(R.id.nav_view);
            Menu drawer_menu = navigation.getMenu();
            MenuItem menuItem = drawer_menu.findItem(R.id.home);
            if (!menuItem.isChecked()) {
                menuItem.setChecked(true);
            }
        } else {
            Log.d("Check back", "SUPER");
            super.onBackPressed();
        }
    }
}
```

c.Login

```
package com.adi.project;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
```

```
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;

import com.adi.project.fragments.AddCourseFragment;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;

import java.util.Objects;

public class Login extends AppCompatActivity {

    Button callSignup, login_btn;
    ImageView image;
    TextView logoText, sloganText;
    TextInputLayout username, password;
    ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //This Line will hide the status bar from the screen
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_login);


        callSignup = findViewById(R.id.signup_screen);
        login_btn = findViewById(R.id.reg_btn);

        username = findViewById(R.id.user_name);
        password = findViewById(R.id.pass_word);


        callSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Login.this, Signup.class);
                startActivity(intent);

            }

        });

        login_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (!validateUsername() | !validatePassword()) {
                    return;
                } else {
                    isUser();
                }

            }
```

```
        });
    }

    private Boolean validateUsername() {
        String val = username.getEditText().getText().toString();

        if (val.isEmpty()) {
            username.setError("Field cannot be empty");
            return false;
        } else {
            username.setError(null);
            username.setErrorEnabled(false);
            return true;
        }
    }

    private Boolean validatePassword() {
        String val = password.getEditText().getText().toString();

        if (val.isEmpty()) {
            password.setError("Field cannot be empty");
            return false;
        } else {
            password.setError(null);
            password.setErrorEnabled(false);
            return true;
        }
    }



    private void isUser() {
        final String userEnteredUsername =
username.getEditText().getText().toString().trim();
        final String userEnteredPassword =
password.getEditText().getText().toString().trim();
        DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("users");
        Query checkUser =
reference.orderByChild("username").equalTo(userEnteredUsername);
        checkUser.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (dataSnapshot.exists()) {
                    username.setError(null);
                    username.setErrorEnabled(false);
                    String passwordFromDB =
dataSnapshot.child(userEnteredUsername).child("password").getValue(String.class);
                    if (passwordFromDB.equals(userEnteredPassword)) {
                        username.setError(null);
                        username.setErrorEnabled(false);
                        String nameFromDB =
dataSnapshot.child(userEnteredUsername).child("name").getValue(String.class);
                        String usernameFromDB =
dataSnapshot.child(userEnteredUsername).child("username").getValue(String.class);
                        String phoneNoFromDB =
dataSnapshot.child(userEnteredUsername).child("phoneNo").getValue(String.class);
                        String emailFromDB =
```

```
dataSnapshot.child(userEnteredUsername).child("email").getValue(String.class);
                        Intent intent = new Intent(getApplicationContext(),
CoursesActivity.class);
                        intent.putExtra("name", nameFromDB);
                        intent.putExtra("username", usernameFromDB);
                        intent.putExtra("email", emailFromDB);
                        intent.putExtra("phoneNo", phoneNoFromDB);
                        intent.putExtra("password", passwordFromDB);
                        startActivity(intent);
                    } else {
                        password.setError("Wrong Password");
                        password.requestFocus();
                    }
                } else {
                    username.setError("No such User exist");
                    username.requestFocus();
                }
            }


            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }


}
```

d.MainActivity

```
package com.adi.project;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ActivityOptions;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Pair;
import android.view.View;
```

```
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private static int SPLASH_SCREEN = 4000;

    //variables
    Animation topAnim, bottomAnim;
    ImageView image, image1;
    TextView logo, slogan, text2, text3 ;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.Layo
utParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_main);

        //Animations
        topAnim = AnimationUtils.loadAnimation(this, R.anim.top_animation);
        bottomAnim = AnimationUtils.loadAnimation(this,R.anim.bottom_animation);

        //Hooks
        image = findViewById(R.id.imageView);
        image1 = findViewById(R.id.imageView2);
        logo = findViewById(R.id.textView);
        slogan = findViewById(R.id.textView2);
        text2 = findViewById(R.id.textView3);
        text3 = findViewById(R.id.textView4);



        logo.setAnimation(topAnim);
        slogan.setAnimation(topAnim);
        image.setAnimation(topAnim);
        image1.setAnimation(bottomAnim);
        text2.setAnimation(bottomAnim);
        text3.setAnimation(bottomAnim);




        new Handler() .postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(MainActivity.this, Login.class);

                Pair[] pairs = new Pair[2];
                pairs[0] = new Pair<View,String>(image, "logo_image");
                pairs[1] = new Pair<View,String>(logo, "logo_text");

                ActivityOptions options =
ActivityOptions.makeSceneTransitionAnimation(MainActivity.this,pairs);
```

```
                startActivity(intent,options.toBundle());


            }
        },SPLASH_SCREEN);


    }
}
```

e.Signup

```
package com.adi.project;

import androidx.appcompat.app.AppCompatActivity;

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

import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class Signup extends AppCompatActivity {

    //Variables
    TextInputLayout regName, regUsername, regEmail, regPhoneNo, regPassword;
    Button  regBtn, callLogin;

    FirebaseDatabase rootNode;
    DatabaseReference reference;

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


        regName = findViewById(R.id.reg_name);
        regUsername = findViewById(R.id.reg_username);
        regEmail = findViewById(R.id.reg_email);
        regPhoneNo = findViewById(R.id.reg_phoneNo);
        regPassword = findViewById(R.id.reg_password);
        regBtn = findViewById(R.id.reg_btn);
        callLogin = findViewById(R.id.login_screen);

        callLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent1 = new Intent(Signup.this,Login.class);
                startActivity(intent1);
```

```
            }
        });

        regBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                rootNode = FirebaseDatabase.getInstance();
                reference = rootNode.getReference("users");

                if(!validateName() | !validatePassword() | !validatePhoneNo() |
!validateEmail() | !validateUsername()){
                    return;
                }

                //Get all the values
                String name = regName.getEditText().getText().toString();
                String username = regUsername.getEditText().getText().toString();
                String email = regEmail.getEditText().getText().toString();
                String phoneNo = regPhoneNo.getEditText().getText().toString();
                String password = regPassword.getEditText().getText().toString();
                UserHelperClass helperClass = new UserHelperClass(name, username,
email, phoneNo, password);
                reference.child(username).setValue(helperClass);
            }

        });
    }

    private Boolean validateName() {
        String val = regName.getEditText().getText().toString();

        if (val.isEmpty()) {
            regName.setError("Field cannot be empty");
            return false;
        } else {
            regName.setError(null);
            regName.setErrorEnabled(false);
            return true;
        }
    }

    private Boolean validateUsername() {
        String val = regUsername.getEditText().getText().toString();
        String noWhiteSpace = "\\A\\w{4,20}\\z";

        if (val.isEmpty()) {
            regUsername.setError("Field cannot be empty");
            return false;
        } else if (val.length() >= 15) {
            regUsername.setError("Username too long");
            return false;
        } else if (!val.matches(noWhiteSpace)) {
            regUsername.setError("White Spaces are not allowed");
            return false;
        } else {
            regUsername.setError(null);
            regUsername.setErrorEnabled(false);
```

```
            return true;
        }
    }

    private Boolean validateEmail() {
        String val = regEmail.getEditText().getText().toString();
        String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+";

        if (val.isEmpty()) {
            regEmail.setError("Field cannot be empty");
            return false;
        } else if (!val.matches(emailPattern)) {
            regEmail.setError("Invalid email address");
            return false;
        } else {
            regEmail.setError(null);
            regEmail.setErrorEnabled(false);
            return true;
        }
    }

    private Boolean validatePhoneNo() {
        String val = regPhoneNo.getEditText().getText().toString();

        if (val.isEmpty()) {
            regPhoneNo.setError("Field cannot be empty");
            return false;
        } else {
            regPhoneNo.setError(null);
            regPhoneNo.setErrorEnabled(false);
            return true;
        }
    }

    private Boolean validatePassword() {
        String val = regPassword.getEditText().getText().toString();
        String passwordVal = "^" +
                //"(?=.*[0-9])" +         //at least 1 digit
                //"(?=.*[a-z])" +         //at least 1 lower case letter
                //"(?=.*[A-Z])" +         //at least 1 upper case letter
                "(?=.*[a-zA-Z])" +      //any letter
                "(?=.*[@#$%^&+=])" +    //at least 1 special character
                "(?=\\S+$)" +           //no white spaces
                ".{4,}" +               //at least 4 characters
                "$";

        if (val.isEmpty()) {
            regPassword.setError("Field cannot be empty");
            return false;
        } else if (!val.matches(passwordVal)) {
            regPassword.setError("Password must contain atleast 4 characters and
one special character");
            return false;
        } else {
            regPassword.setError(null);
            regPassword.setErrorEnabled(false);
            return true;
        }
```

```
    }
}
```

4.Database

a.userhelperclass

```
package com.adi.project;

public class UserHelperClass {
    String name, username, email, phoneNo, password;

    public UserHelperClass() {
    }
    public UserHelperClass(String name, String username, String email, String
phoneNo, String password) {
        this.name = name;
        this.username = username;
        this.email = email;
        this.phoneNo = phoneNo;
        this.password = password;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhoneNo() {
        return phoneNo;
    }

    public void setPhoneNo(String phoneNo) {
```

```
        this.phoneNo = phoneNo;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

b.Databasehelper

```
package com.adi.project.helpers;

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

import com.adi.project.fragments.AddAttendanceFragment;
import com.adi.project.fragments.AddMarksFragment;
import com.adi.project.models.Course;

import java.util.ArrayList;


public class DatabaseHelper extends SQLiteOpenHelper {

    // instance of database helper
    private static DatabaseHelper sInstance;

    // identifiers
    public static final String DATABASE_NAME = "Course.db";
    public static final String COURSE_TABLE_NAME = "courses";
    public static final String STUDENT_TABLE_NAME = "students";

    public static final String COURSE_COLUMN_COURSE_CODE = "courseCode";
    public static final String COURSE_COLUMN_COURSE_NAME = "courseName";
    public static final String COURSE_COLUMN_STUDENT_COUNT = "studentCount";
    public static final String COURSE_COLUMN_DAY_COUNT = "dayCount";
    public static final String COURSE_COLUMN_CR_EMAIL = "emailCr";
    public static final String COURSE_COLUMN_TA_EMAIL = "emailTa";

    public static final String STUDENT_COLUMN_COURSE_CODE = "courseCode";
    public static final String COURSE_COLUMN_ID = "id";
    public static final String COURSE_COLUMN_INSEMESTER = "insem";
    public static final String COURSE_COLUMN_ENDSEMESTER = "endsem";
    public static final String COURSE_COLUMN_DATES = "dates";
```

```
    public static synchronized DatabaseHelper getInstance(Context context) {

        // Use the application context, which will ensure that you
        // don't accidentally leak an Activity's context.
        if (sInstance == null) {
            sInstance = new DatabaseHelper(context.getApplicationContext());
        }
        return sInstance;
    }


    /**
     * Constructor should be private to prevent direct instantiation.
     * make call to static method "getInstance()" instead.
     */
    private DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, 1);
    }


    @Override
    public void onCreate(SQLiteDatabase db) {
        // create two tables
        db.execSQL(
                "create table courses " +
                        "(courseCode STRING primary key, courseName STRING,
studentCount INTEGER, dayCount INTEGER, emailCr STRING, emailTa STRING)"
        );

        db.execSQL(
                "create table students " +
                        "(id INTEGER, courseCode STRING, insem FLOAT, endsem
FLOAT, dates STRING)"
        );
    }


    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }

    public void insertCourse(Course course) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("courseCode", course.getmCourseCode());
        contentValues.put("courseName", course.getmCourseName());
        contentValues.put("studentCount", course.getmStudentCount());
        contentValues.put("dayCount", course.getmDayCount());
        contentValues.put("emailCr", course.getmEmailCr());
        contentValues.put("emailTa", course.getmEmailTa());
        db.insert("courses", null, contentValues);

        insertStudentForCourse(course.getmCourseCode(),
course.getmStudentCount());
    }


    /**
```

```
     * Insert Students for a course in students table
     *
     * @param courseCode   course code for that course
     * @param studentCount student count for that course
     */
    private void insertStudentForCourse(String courseCode, int studentCount) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        for (int i = 0; i < studentCount; ++i) {
            contentValues.put("courseCode", courseCode);
            contentValues.put("id", i + 1);
            contentValues.put("dates", "");
            contentValues.put("insem", 0);
            contentValues.put("endsem", 0);
            db.insert("students", null, contentValues);
        }

    }


    /**
     * Get student count for a course
     *
     * @param courseCode course code for that course
     * @return student count for the course
     */
    public int getStudentCount(String courseCode) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor c = db.rawQuery("select courseCode,studentCount from courses",
null);
        int studentCount = 0;
        // iterate through column elements
        if (c.moveToFirst()) {
            do {
                // check for the course code passed
                if (c.getString(0).equals(courseCode)) {
                    // found course entry
                    studentCount = c.getInt(1);
                    break;
                }
            } while (c.moveToNext());
        }
        c.close();
        return studentCount;
    }


    /**
     * Get day count for a course
     * @param courseCode of the course
     * @return daycount
     */
    public int getDayCount(String courseCode) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor c = db.rawQuery("select courseCode,dayCount from courses", null);
        int dayCount = 0;
        // iterate through column elements
        if (c.moveToFirst()) {
            do {
```

```
                // check for the course code passed
                if (c.getString(0).equals(courseCode)) {
                    // found course entry
                    dayCount = c.getInt(1);
                    break;
                }
            } while (c.moveToNext());
        }
        c.close();
        return dayCount;
    }


    /**
     * Increment day count for a course
     * @param courseCode of the course
     */
    private void incrementDayCount(String courseCode) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor c = db.rawQuery("select courseCode,dayCount from courses", null);

        if (c.moveToFirst()) {
            do {
                if (c.getString(0).equals(courseCode)) {
                    int newDayCount = c.getInt(1);
                    // add new day count to dayCount in table
                    ContentValues values = new ContentValues();
                    values.put("dayCount", newDayCount + 1);
                    db.update("courses", values, "courseCode= '" + courseCode
+"'", null);
                }
            } while(c.moveToNext());
        }
//        c = db.rawQuery("select courseCode,dayCount from courses", null);
//        // move though rows in tables selected above
//        if (c.moveToFirst()) {
//            do {
//                Log.d("CODE", c.getString(0));
//                Log.d("DAYCOUNT", String.valueOf(c.getInt(1)));
//            } while (c.moveToNext());
//        }
//        c.close();

    }


    /**
     * Update attendance field in dates
     * @param courseCode            course for the attendance
     * @param studentAttendanceList attendance list
     * @param date                  date of taking attendance
     */
    public void updateAttendance(String courseCode,
ArrayList<AddAttendanceFragment.StudentAttendance> studentAttendanceList, String
date) {
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor c = db.rawQuery("select courseCode,id,dates from students", null);
        // iterator for student attendance list
```

```
        int it = 0;

        // move though rows in tables selected above
        if (c.moveToFirst()) {
            do {
                if (it == studentAttendanceList.size()) {
                    break;
                }
                // check for the course code
                if (c.getString(0).equals(courseCode)) {
                    // check if student is present or absent
                    if (studentAttendanceList.get(it).isChecked()) {
                        String dates = c.getString(2);
                        dates += "," + date;
                        // add date to dates in table
                        ContentValues values = new ContentValues();
                        values.put("dates", dates);
                        db.update("students", values, "courseCode= '" + courseCode
+ "' AND id= " + (it + 1) + "", null);
                    }
                    ++it;
                }
            } while (c.moveToNext());
        }
        c.close();

        c = db.rawQuery("select courseCode,id,dates from students", null);
        // move though rows in tables selected above
        if (c.moveToFirst()) {
            do {
                Log.d("CODE", c.getString(0));
                Log.d("ID", c.getString(1));
                Log.d("DATES", c.getString(2));

            } while (c.moveToNext());
        }
        c.close();

        incrementDayCount(courseCode);
    }


    /**
     * Get course code, insem, endsem, dates column from students table
     * @return Cursor for the above
     */
    public Cursor getStudentInfo() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery( "select courseCode,insem,endsem,dates from students",
null );
    }


    /**
     * Get course code column from course table
     * @return course code column
     */
    public Cursor getCourseInfo() {
        SQLiteDatabase db = this.getReadableDatabase();
```

```
        return db.rawQuery( "select courseCode,courseName,emailCr,emailTa from
courses", null );
    }


    /**
     * Get course code and dates from students table
     * @return cursor to both columns
     */
    public Cursor getStudentAttendance() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery( "select courseCode,dates from students", null );
    }


    /**
     * Update insem, endsem marks for students in a course
     * @param courseCode of the course
     * @param studentMarksList student marks list
     */
    public void addMarks(String courseCode,
ArrayList<AddMarksFragment.StudentMarks> studentMarksList) {
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor c = db.rawQuery("select courseCode,id,insem,endsem from students",
null);
        // iterator for student marks list
        int it = 0;

        // move though rows in tables selected above
        if (c.moveToFirst()) {
            do {
                if (it == studentMarksList.size()) {
                    break;
                }
                // check for the course code
                if (c.getString(0).equals(courseCode)) {
                    ContentValues values = new ContentValues();
//                    Log.d("Check
insem",String.valueOf(studentMarksList.get(it).getInsem()));
                    // add fields
                    values.put("insem", studentMarksList.get(it).getInsem());
                    values.put("endsem", studentMarksList.get(it).getEndsem());
                    db.update("students", values, "courseCode= '" + courseCode +
"' AND id= " + (it + 1) + "", null);
                    ++it;
                }
            } while (c.moveToNext());
        }
        c.close();
    }


    /**
     * Get the course table columns
     * @return Cursor for the above
     */
    public Cursor getCourses() {
        SQLiteDatabase db = this.getReadableDatabase();
```

```
        return db.rawQuery("select
courseCode,courseName,studentCount,emailCr,emailTa,dayCount  from courses", null);
    }
}
```

LAYOUTS:

a.activity_course.xml

```
<?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:layout_width="match_parent"
    android:id="@+id/layout_course"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CourseActivity">

    <!--Add toolbar-->
    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/colorPrimary"
```

```
            app:titleTextColor="@android:color/white"
            android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
    </com.google.android.material.appbar.AppBarLayout>

    <!-- Fragment space -->
    <FrameLayout
        android:id="@+id/fragContainer"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </FrameLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:id="@+id/course_view">
        <!--Cards-->
        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="8dp">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:padding="16dp">
                    <TextView
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginBottom="8dp"
                        android:text="Attendance"
                        android:textColor="#000"
                        android:textSize="24sp" />
                    <TextView
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:text="Record and display attendance"
                        android:textColor="#555"
                        android:textSize="14sp"/>
                </LinearLayout>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content">
                    <Button
                        android:id="@+id/addAttendance"
                        android:layout_width="wrap_content"
```

```
                    android:layout_height="wrap_content"
                    android:text="ADD"
                    android:textSize="16sp"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle"
                    />
                <Button
                    android:id="@+id/viewAttendance"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="VIEW"
                    android:textSize="16sp"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle" />
            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:padding="16dp">
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="8dp"
                    android:text="Student Details"
                    android:textColor="#000"
                    android:textSize="24sp" />
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Display student marks and attendance"
                    android:textColor="#555"
                    android:textSize="14sp"/>
            </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <Button
                    android:id="@+id/viewStudentDetails"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="View"
                    android:textSize="16sp"
                    android:textColor="#F79C0D"
```

```
                    style="?android:attr/borderlessButtonStyle" />

                <Button
                    android:id="@+id/searchByStudentId"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Search"
                    android:textSize="16sp"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle" />

            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:padding="16dp">
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="8dp"
                    android:text="Contact"
                    android:textColor="#000"
                    android:textSize="24sp" />
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Frequently used contacts"
                    android:textColor="#555"
                    android:textSize="14sp"/>
            </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <Button
                    android:onClick="contactMailCr"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="CR"
                    android:textSize="16sp"
                    android:textAllCaps="false"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle" />
                <Button
```

```
                    android:onClick="contactMailTa"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="TA"
                    android:textSize="16sp"
                    android:textAllCaps="false"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle" />
            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:padding="16dp">
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="8dp"
                    android:text="Marks"
                    android:textColor="#000"
                    android:textSize="24sp" />
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Record insem and endsem marks"
                    android:textColor="#555"
                    android:textSize="14sp"/>
            </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <Button
                    android:id="@+id/addMarks"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="ADD"
                    android:textSize="16sp"
                    android:textColor="#F79C0D"
                    style="?android:attr/borderlessButtonStyle"
                    />

            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="16dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="8dp"
                android:text="Deadlines"
                android:textColor="#000"
                android:textSize="24sp" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Add notification for project submission"
                android:textColor="#555"
                android:textSize="14sp"/>
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <Button
                android:id="@+id/addNotification"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="ADD"
                android:textSize="16sp"
                android:textColor="#F79C0D"
                style="?android:attr/borderlessButtonStyle"
                />

            <Button
                android:id="@+id/viewNotification"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="VIEW"
                android:textSize="16sp"
                android:textColor="#F79C0D"
                style="?android:attr/borderlessButtonStyle"
                />

        </LinearLayout>
```

```
            </LinearLayout>

        </androidx.cardview.widget.CardView>

    </LinearLayout>

    </ScrollView>

</LinearLayout>
```

b.activity_courses.xml

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#EEEEEE"
    android:fitsSystemWindows="true">

    <!--Main body-->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <!--Add toolbar-->
        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="@color/colorPrimary"
                android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
                app:titleTextColor="@android:color/white" />
        </com.google.android.material.appbar.AppBarLayout>


        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/courses_recycler_view"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_marginTop="64dp"/>


        <!--FAB-->
        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/addCourseFab"
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|end"
            android:layout_margin="16dp"
            android:clickable="true"
            android:focusable="true"
            android:src="@drawable/ic_add"
            android:tint="@android:color/white"
            app:backgroundTint="@color/colorPrimary"
            app:rippleColor="@color/cardview_light_background" />
    </FrameLayout>


    <!-- Fragment -->
    <FrameLayout
        android:id="@+id/fragContainer"
        android:layout_width="fill_parent"
        android:layout_height="0dp"></FrameLayout>


    <!--Drawer part-->
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header"
        app:menu="@menu/drawer_view" />
    <!--drawer_view has the contents and nav_header has the header-->

</androidx.drawerlayout.widget.DrawerLayout>
```

c.activity_login.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#fff"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".Login">

    <ImageView
        android:id="@+id/logo_image"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:src="@drawable/logo"
        android:transitionName="logo_image" />


    <TextView
        android:id="@+id/logo_name"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/bungee"
        android:text="Hello there, Welcome Back"
        android:textColor="#F03A3A"
        android:textSize="40sp"
        android:transitionName="logo_text" />

    <TextView
        android:id="@+id/slogan_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign In to continue"
        android:textSize="18sp" />


    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="20dp"
        android:orientation="vertical">

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/user_name"
            style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Username">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/pass_word"
            style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            app:passwordToggleEnabled="true">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

                android:inputType="textPassword" />

        </com.google.android.material.textfield.TextInputLayout>

        <Button
            android:layout_width="200dp"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_margin="5dp"
            android:background="#00000000"
            android:elevation="0dp"
```

```
            android:text="Forget Password?" />

        <Button
            android:id="@+id/reg_btn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="5dp"
            android:background="#000"
            android:text="GO"
            android:textColor="#fff" />

        <Button
            android:id="@+id/signup_screen"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_margin="5dp"
            android:background="#00000000"
            android:elevation="0dp"
            android:text="New User? SIGN UP"
            android:textColor="#000" />

    </LinearLayout>

</LinearLayout>
```

d.activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
    tools:context="MainActivity">
    android:background="#ECE8E8">


    <ImageView
        android:id="@+id/imageView"
        android:layout_width="102dp"
        android:layout_height="102dp"
        android:layout_marginStart="10dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="10dp"
        android:src="@drawable/logo"
        android:textAlignment="center"
        android:transitionName="logo_image"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintVertical_bias="0.0" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="400dp"
        android:layout_height="458dp"
        android:layout_marginStart="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginBottom="70dp"
        android:src="@drawable/logo2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.6"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.558" />


    <TextView
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="9dp"
        android:background="#ff0000"
        android:fontFamily="@font/bungee"
        android:text="BISHOP HEBER COLLEGE"
        android:textAlignment="center"
        android:textColor="#F3EBEB"
        android:textSize="18sp"
        android:transitionName="logo_text"
        app:layout_constraintBottom_toTopOf="@+id/imageView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="5dp"
        android:fontFamily="@font/aladin"
        android:text="Teacher's Assistant"
        android:textAlignment="center"
        android:textColor="#ff0000"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+id/imageView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="18dp"
        android:layout_marginBottom="16dp"
```

```
        android:text='Recognized by UGC as "college of Excellence"'
        android:textAlignment="center"
        android:textColor="#00008b"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView2"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="1dp"
        android:text="Nationally Re-Acccredited at the 'A'  by NAAC with  a CGPA
of 3.58 out of 4"
        android:textAlignment="center"
        android:textColor="#00008b"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

e.activity_signup.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".Signup">

    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/logo" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="-10dp"
        android:fontFamily="@font/bungee"
        android:text="Welcome"
        android:textColor="#F03A3A"
        android:textSize="40sp" />

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

```
        android:layout_marginTop="-10dp"
        android:text="SignUp to Start with TA"
        android:textColor="#F03A3A"
        android:textSize="18sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="20dp"
        android:orientation="vertical">

        <com.google.android.material.textfield.TextInputLayout

            android:id="@+id/reg_name"
            style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Full Name">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="text" />

        </com.google.android.material.textfield.TextInputLayout>


        <com.google.android.material.textfield.TextInputLayout

            android:id="@+id/reg_username"
            style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="UserName"
            app:counterMaxLength="15">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="text" />

        </com.google.android.material.textfield.TextInputLayout>


        <com.google.android.material.textfield.TextInputLayout

            android:id="@+id/reg_email"
            style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="textEmailAddress" />

        </com.google.android.material.textfield.TextInputLayout>
```

```
        <com.google.android.material.textfield.TextInputLayout

            android:id="@+id/reg_phoneNo"
            style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Phone Number">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="number" />

        </com.google.android.material.textfield.TextInputLayout>


        <com.google.android.material.textfield.TextInputLayout

            android:id="@+id/reg_password"
            style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            app:passwordToggleEnabled="true">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="textPassword" />

        </com.google.android.material.textfield.TextInputLayout>

    </LinearLayout>


    <Button
        android:id="@+id/reg_btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#000"
        android:text="Go"
        android:textColor="#fff" />

    <Button
        android:id="@+id/login_screen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#00000000"
        android:text="Already have an account? LogIn" />


</LinearLayout>
```

f.fragment_add_attendance.xml

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

    <!-- Attendance list -->
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/add_attendance_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

g.fragment_add_course.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment_add_course"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginStart="16dp"
    android:layout_marginTop="64dp"
    android:background="#ffffff"
    tools:context="com.adi.project.fragments.AddCourseFragment">

    <!-- Course Code -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_course" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/courseNameEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="Course"
                android:inputType="text" />
```

```
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>

    <!-- Course Code -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_code" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/courseCodeEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="Course Code"
                android:inputType="textCapCharacters" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>

    <!-- Student Count -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_count" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/studentCountEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="Student Count"
                android:inputType="numberDecimal" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>
```

```
    <!-- Email CR -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_email" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/emailCrEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="CR"
                android:inputType="textEmailAddress" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>

    <!-- Email TA -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_email" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/emailTaEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="TA"
                android:inputType="textEmailAddress" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>

</LinearLayout>
```

h.fragment_add_marks.xml

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

    <!-- Table Headings -->
    <LinearLayout
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:layout_weight="1"
            android:textSize="18sp"
            android:text="Student ID"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Insemester"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Endsemester"/>
    </LinearLayout>

    <!-- Scroll View -->
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
            android:layout_margin="16dp">
            <!-- Table for id, insem, endsem -->
            <TableLayout
                android:padding="8dp"
                android:id="@+id/marksTable"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:stretchColumns="0,1,2">
            </TableLayout>
        </LinearLayout>
    </ScrollView>

</LinearLayout>
```

i.fragment_add_notification.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">

</LinearLayout>
```

j.fragment_backup.xml

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

    <!-- Post loader -->
    <LinearLayout
        android:id="@+id/layoutProgress"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:visibility="gone" >

        <ProgressBar
            android:id="@+id/pbHeaderProgress"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >
        </ProgressBar>
    </LinearLayout>


    <ImageView
```

```
        android:id="@+id/done"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_done_all"
        android:layout_margin="32dp"
        android:visibility="gone"
        />

    <!-- Request Backup -->
    <Button
        android:id="@+id/backupButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Backup all data"
        android:background="@color/colorPrimary"
        android:textColor="#ffffff"
        android:layout_gravity="center"
        android:gravity="center"
        android:padding="8dp"
        />

</LinearLayout>
```

k.fragment_document_similarity.cml

```
<?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">

    <TextView
        android:layout_marginTop="72dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="16dp"
        android:textSize="16dp"
        android:textStyle="bold"
        android:text="Document 1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <!--First string-->
    <EditText
        android:id="@+id/doc1"
        android:hint="Enter first document"
        android:paddingTop="8dp"
        android:paddingBottom="8dp"
        android:layout_margin="16dp"
        android:inputType="textMultiLine"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:textSize="16dp"
```

```
        android:textStyle="bold"
        android:layout_margin="16dp"
        android:text="Document 2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <!--Second string-->
    <EditText
        android:id="@+id/doc2"
        android:hint="Enter second document"
        android:paddingTop="8dp"
        android:paddingBottom="8dp"
        android:layout_margin="16dp"
        android:inputType="textMultiLine"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/similarity_button"
        android:padding="8dp"
        android:layout_margin="16dp"
        android:text="SIMILARITY"
        android:background="@color/colorPrimary"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@android:color/white"/>

    <!--Display result-->
    <TextView
        android:id="@+id/display_similarity"
        android:textSize="20dp"
        android:padding="8dp"
        android:textStyle="bold"
        android:hint="Result"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

l.fragment_help_and_feedback.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment_help_and_feedback"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ffffff"
    android:layout_marginTop="64dp"
    tools:context="com.adi.project.fragments.HelpAndFeedbackFragment">

    <!-- rating bar with default style -->
    <RatingBar
```

```
        android:id="@+id/rating"
        android:layout_gravity="center_horizontal"
        android:layout_width="wrap_content"
        android:layout_height="56dp"
        android:theme="@style/RatingBar"
        style="android:style/Widget.Material.RatingBar.Small"
        android:numStars="5"
        android:rating="5"
        android:stepSize="1"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="32dp"/>

    <!-- Feedback -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:layout_marginStart="16dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_thumbs_up_down" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/feedbackEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginStart="16dp"
                android:hint="Write your feedback"
                android:inputType="textMultiLine"
                android:lines="5"
                android:maxLines="10"
                android:gravity="top|left"
                />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>

    <!-- Suggestions -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginStart="16dp"
        android:orientation="horizontal">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/ic_new_feat"/>
```

```
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/otherFeatEdit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginEnd="32dp"
                android:layout_marginTop="8dp"
                android:layout_marginStart="16dp"
                android:hint="Any other features needed"
                android:inputType="text" />
        </com.google.android.material.textfield.TextInputLayout>

    </LinearLayout>

</LinearLayout>
```

m.fragment_search_Student.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">

    <LinearLayout
        android:layout_margin="16dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/searchForId"
            android:layout_width="80dp"
            android:inputType="number"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/searchButton"
            android:padding="8dp"
            android:layout_margin="16dp"
            android:text="Search"
            android:background="@color/colorPrimary"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="clip_vertical"
            android:textColor="@android:color/white"/>

    </LinearLayout>

    <!-- View Headings -->
    <LinearLayout
        android:orientation="horizontal"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:layout_weight="1"
            android:textSize="18sp"
            android:text="ID"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Insem"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Endsem"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="%"/>
    </LinearLayout>


        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="16dp">

            <TextView
                android:textAlignment="center"
                android:text=""
                android:id="@+id/studentId"
                android:layout_weight="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
```

```
                android:textAlignment="center"
                android:text=""
                android:id="@+id/insemMarks"
                android:layout_weight="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                android:textAlignment="center"
                android:text=""
                android:id="@+id/endsemMarks"
                android:layout_weight="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                android:textAlignment="center"
                android:text=""
                android:id="@+id/attendancePercent"
                android:layout_weight="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

        </LinearLayout>


</LinearLayout>
```

n.fragment_Student_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">

    <!-- View Headings -->
    <LinearLayout
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:layout_weight="1"
            android:textSize="18sp"
            android:text="ID"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Insem"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="Endsem"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:textStyle="bold"
            android:layout_weight="1"
            android:layout_gravity="center_vertical"
            android:textAlignment="center"
            android:textSize="18sp"
            android:text="%"/>
    </LinearLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/list_student_details"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </androidx.recyclerview.widget.RecyclerView>

</LinearLayout>
```

o.fragment_view_attendance.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">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/view_attendance_list"/>

</LinearLayout>
```

p.item_add_attendance.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="horizontal"
    android:gravity="center"
    android:padding="16dp">

    <!-- Student Id -->
    <TextView
        android:id="@+id/studentIdText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:layout_weight="12"
        android:layout_gravity="center_vertical"
        android:textColor="@android:color/black"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp" />

    <!-- Check box -->
    <CheckBox
        android:id="@+id/check"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="end"
        android:layout_marginEnd="16dp"
        style="@style/GreenCheck"
        />

</LinearLayout>
```

q.item_attendance

```
<?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">

    <!-- Student Id -->
    <TextView
        android:id="@+id/studentId"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginEnd="16dp"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:textColor="@android:color/black"
        android:textSize="24sp" />
```

```
    <!-- Attendance Status -->
    <TextView
        android:id="@+id/attendanceStatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="16dp"
        android:layout_marginEnd="16dp"
        android:layout_gravity="center_vertical"
        style="@style/GreenCheck"
        />

</LinearLayout>
```

r.item_course.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Card elevation elevation and margin attribute without frame layout -->
<FrameLayout 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="wrap_content">

    <!-- Card -->
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FEFEFE"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:elevation="2dp"
        app:cardCornerRadius="4dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <!-- Course Name -->
            <TextView
                android:id="@+id/courseNameText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="24sp"
                android:layout_marginTop="22dp"
                android:layout_marginStart="16dp"
                android:layout_marginEnd="16dp"
                android:textColor="@android:color/black"
                />
```

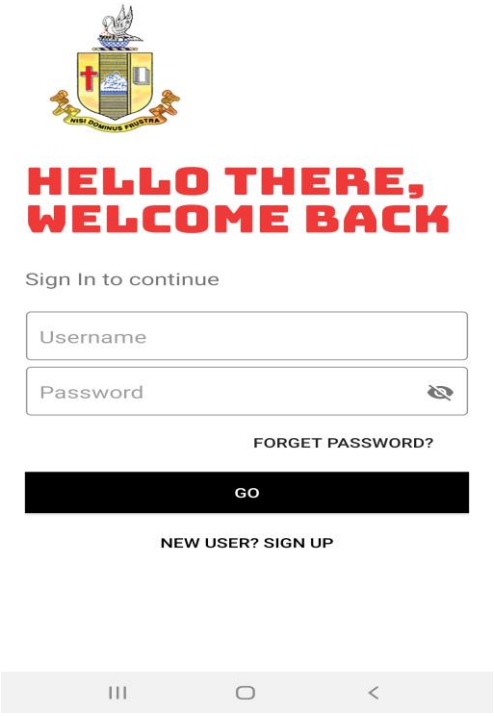
```
            <!-- Course Code -->
            <TextView
                android:id="@+id/courseCodeText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="18sp"
                android:layout_marginTop="12dp"
                android:layout_marginStart="16dp"
                android:layout_marginEnd="16dp"
                android:layout_marginBottom="16dp"
                android:textStyle="bold"
                />

        </LinearLayout>

    </androidx.cardview.widget.CardView>

</FrameLayout>
```

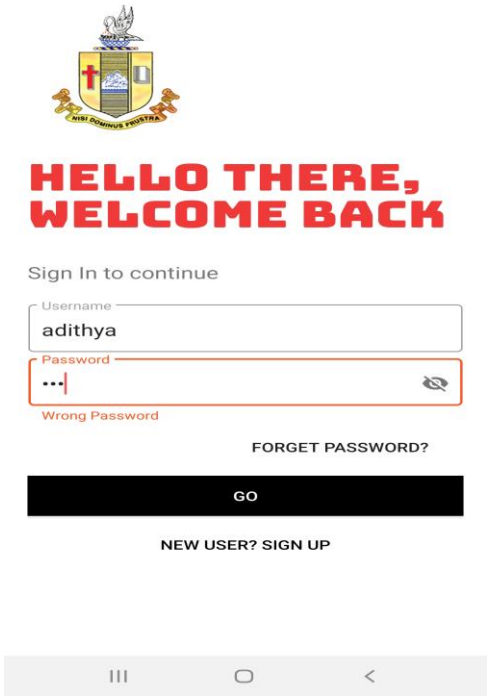
s.iteam_Student_details.xml

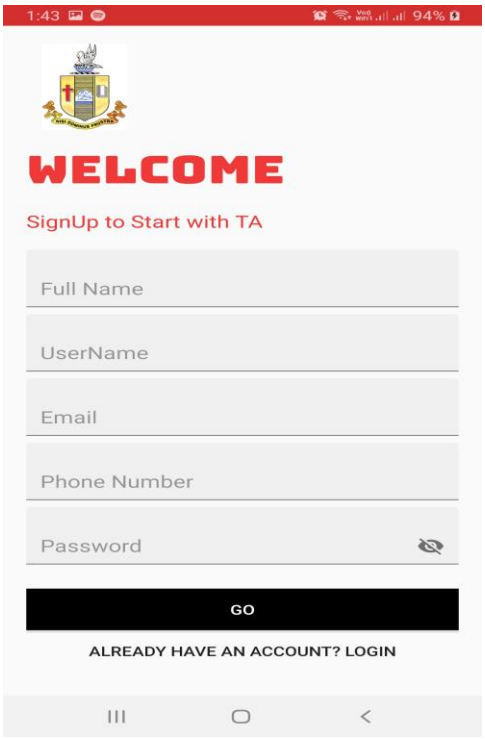
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <!-- ID -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:layout_weight="1"
        android:layout_margin="16dp"
        android:textAlignment="center"
        android:id="@+id/studentId"/>

    <!-- Insem -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:textSize="18sp"
        android:textAlignment="center"
        android:layout_weight="1"
        android:id="@+id/insem"/>

    <!-- Endsem -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:layout_weight="1"
```

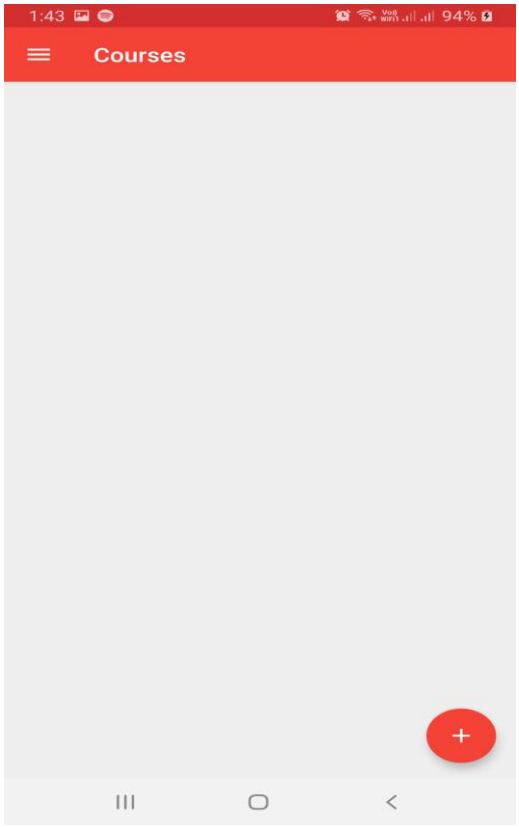
```
        android:textAlignment="center"
        android:textSize="18sp"
        android:id="@+id/endsem"/>

    <!-- Attendance % -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:layout_weight="1"
        android:textAlignment="center"
        android:textSize="18sp"
        android:id="@+id/attendance"/>

</LinearLayout>
```

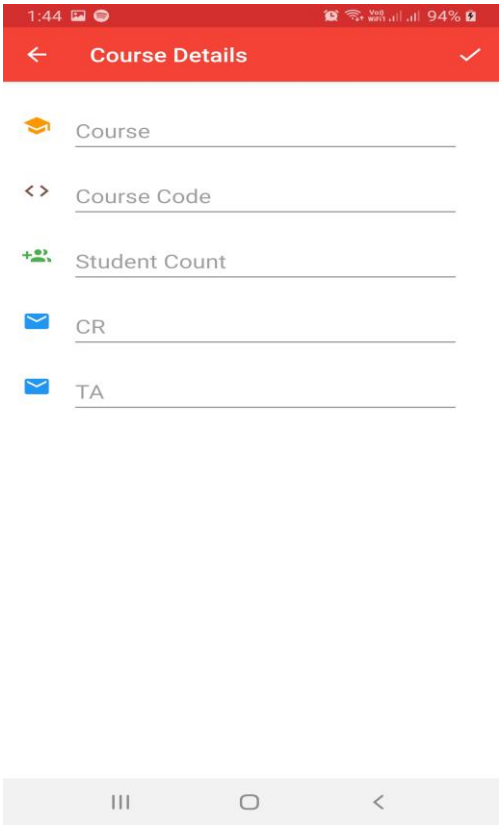
SCREENSHOTS:

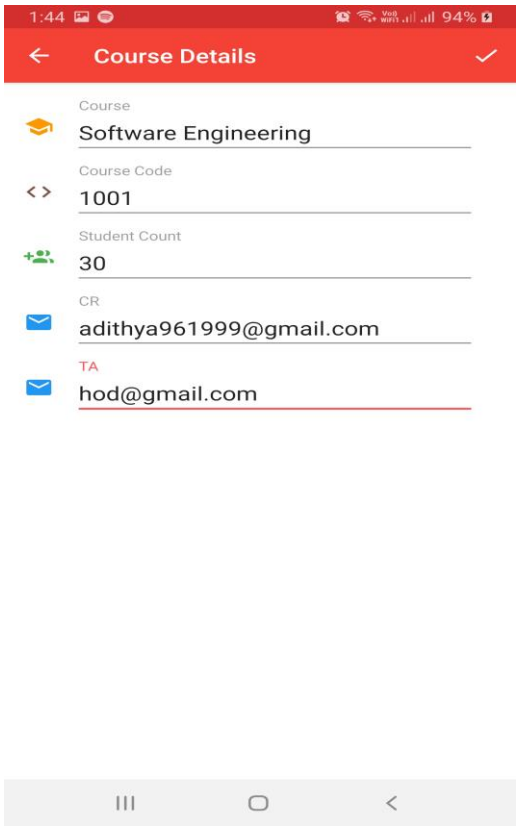


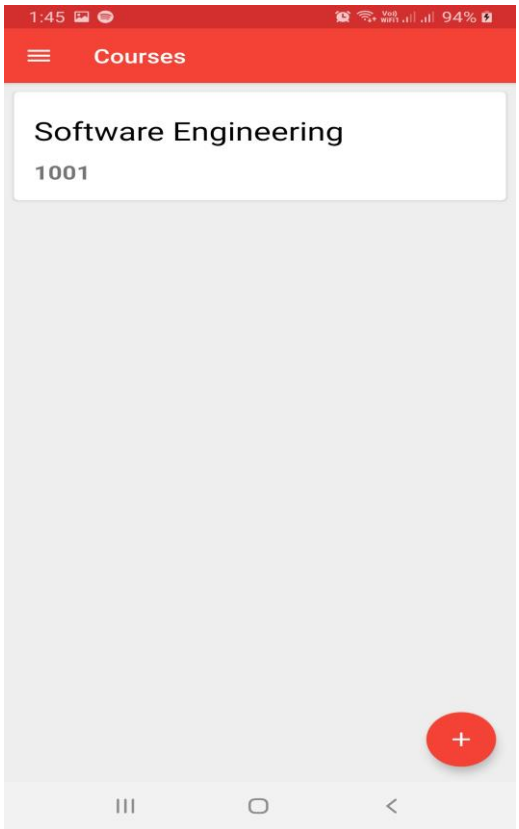


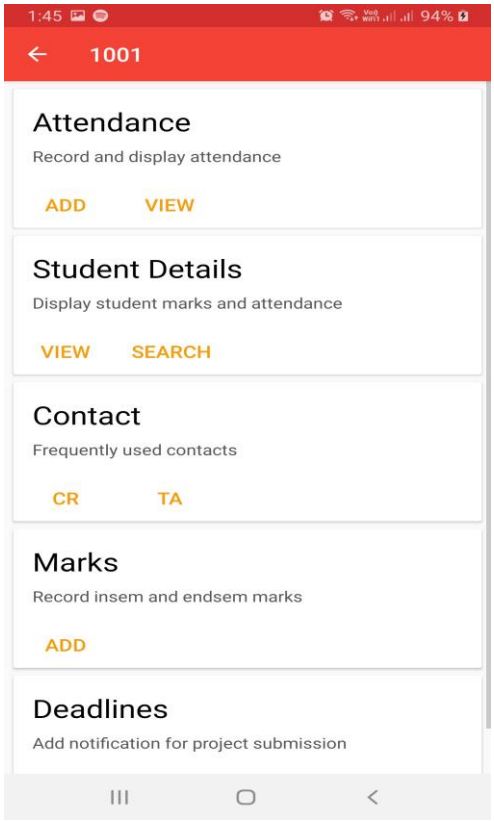


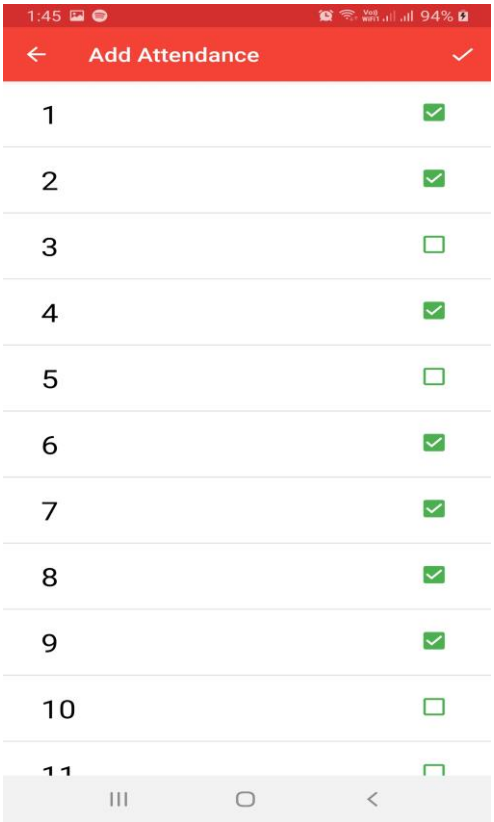

1:43
94%
Courses

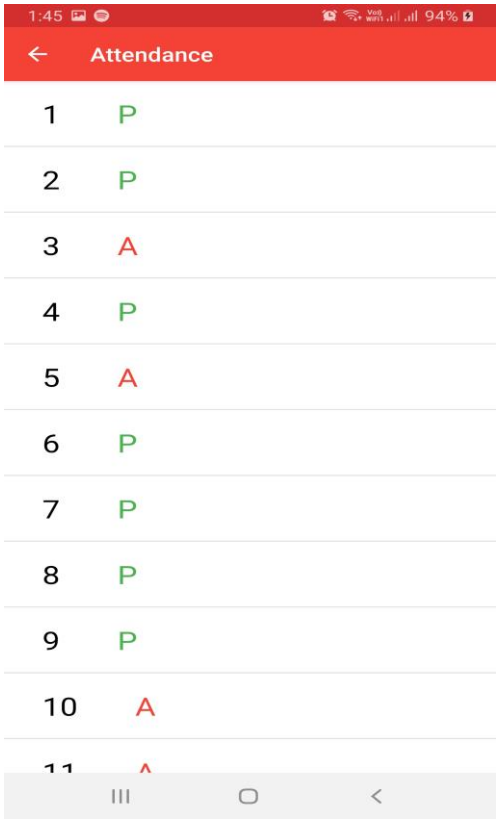

1:44
94%
Course Details
Course
Course Code
Student Count
CR
TA

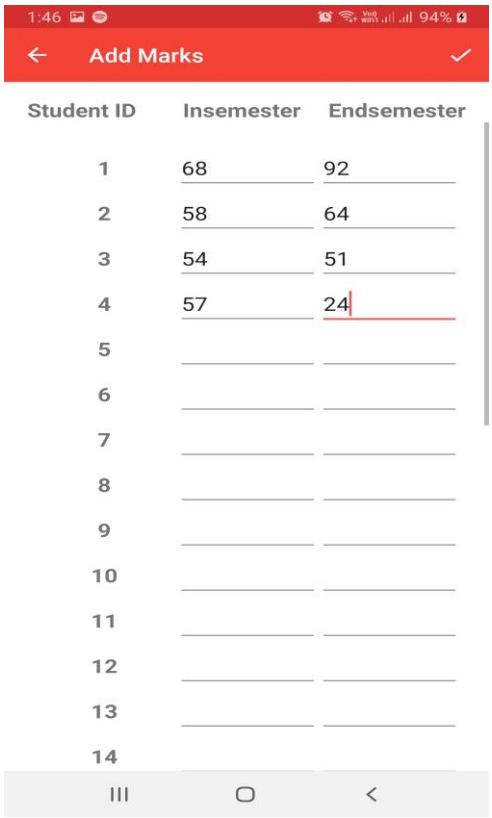

1:44
94%
Course Details
Course
Software Engineering
Course Code
1001
Student Count
30
CR
adithya961999@gmail.com
TA
hod@gmail.com

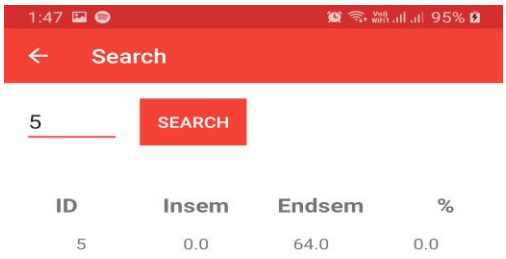

1:45
94%
Courses
Software Engineering
1001

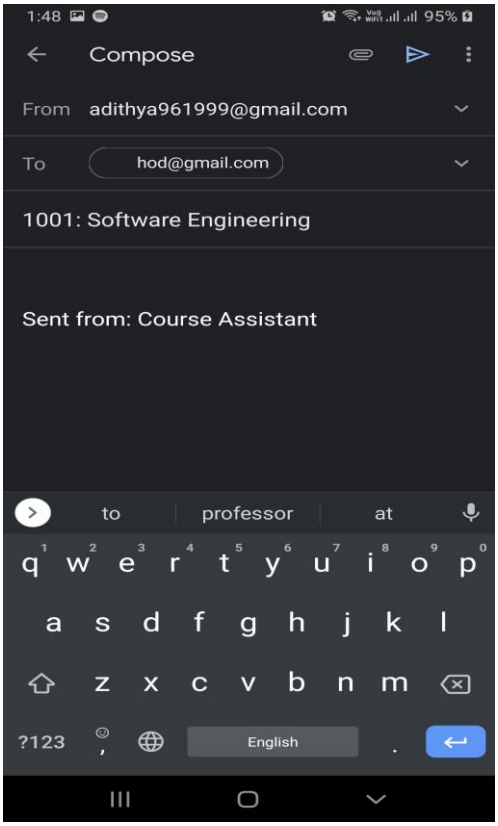

1:45
94%
1001
Attendance
Record and display attendance
ADD
VIEW
Student Details
Display student marks and attendance
VIEW
SEARCH
Contact
Frequently used contacts
CR
TA
Marks
Record insem and endsem marks
ADD
Deadlines
Add notification for project submission

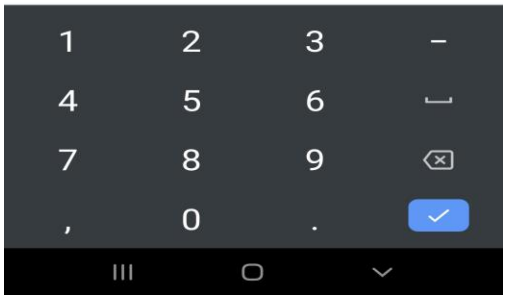

1:45
94%
Add Attendance
1
2
3
4
5
6
7
8
9
10

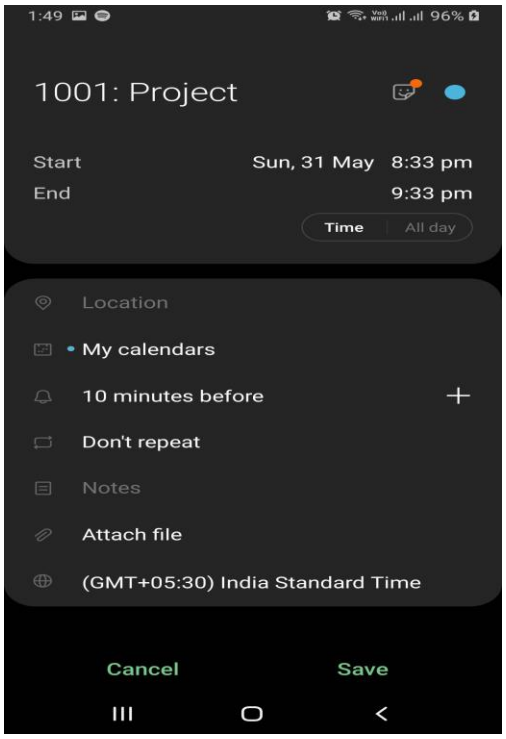

1:45
94%
Attendance
1 P
2 P
3 A
4 P
5 A
6 P
7 P
8 P
9 P
10 A

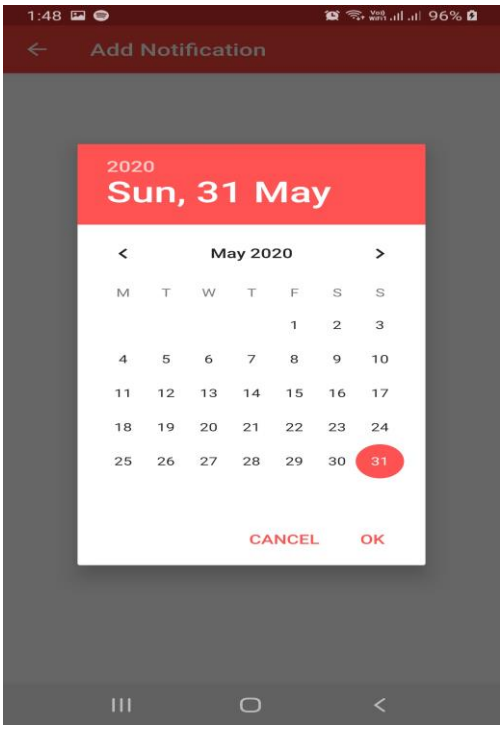

1:46
94%
Add Marks
Student ID
Insemester
Endsemester
1 68 92
2 58 64
3 54 51
4 57 24
5
6
7
8
9
10
11
12
13
14

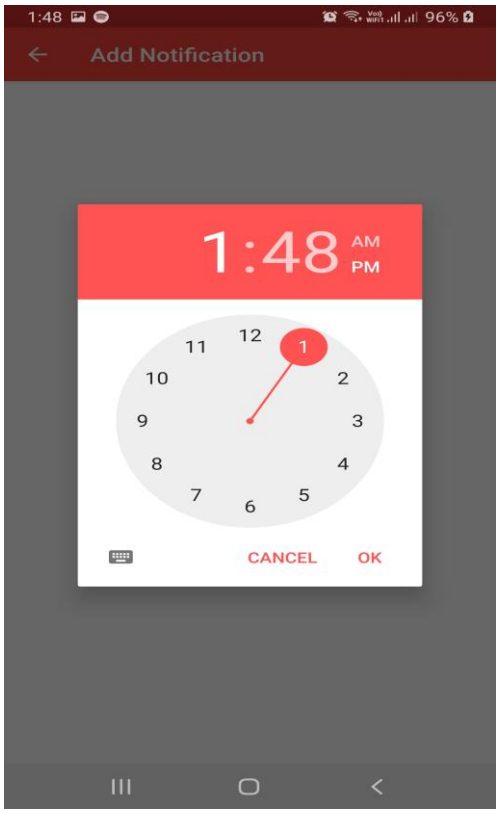

1:47
95%
Search
5
SEARCH
ID
Insem
Endsem
%
5
0.0
64.0
0.0


1:48
95%
Compose
From
adithya961999@gmail.com
To
hod@gmail.com
1001: Software Engineering
Sent from: Course Assistant
to
professor
at
English

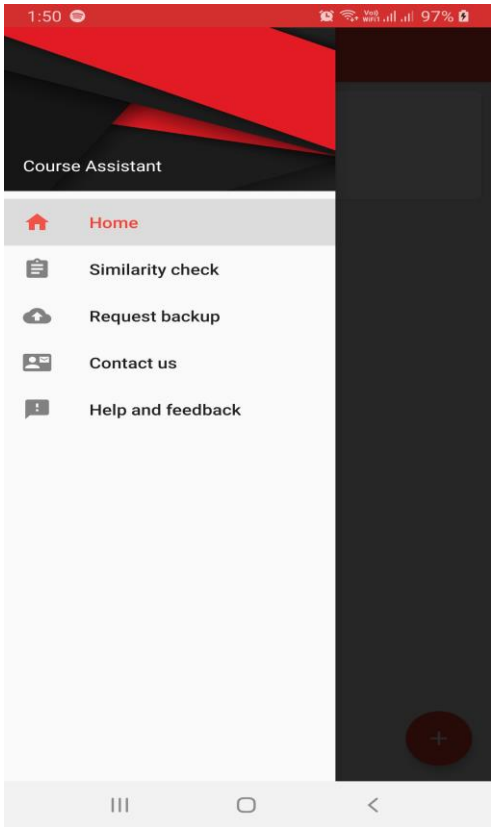

1 2 3 –
4 5 6
7 8 9
, 0 .

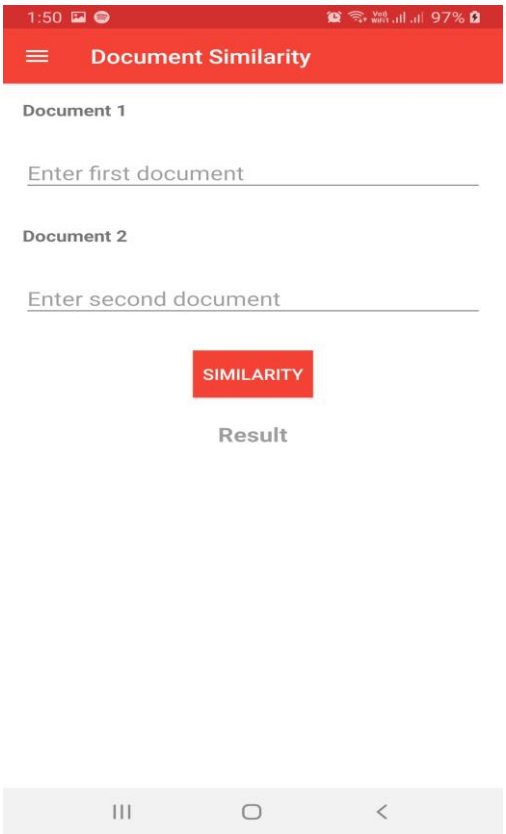

1:49
96%
1001: Project
Start
Sun, 31 May 8:33 pm
End
9:33 pm
Time
All day
Location
My calendars
10 minutes before
Don't repeat
Notes
Attach file
(GMT+05:30) India Standard Time
Cancel
Save

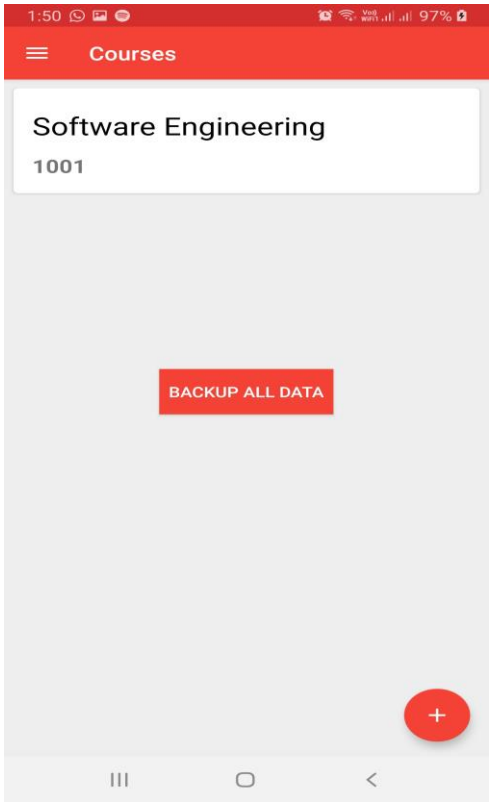

1:48
96%
Add Notification
2020
Sun, 31 May
May 2020
M T W T F S S
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
CANCEL
OK

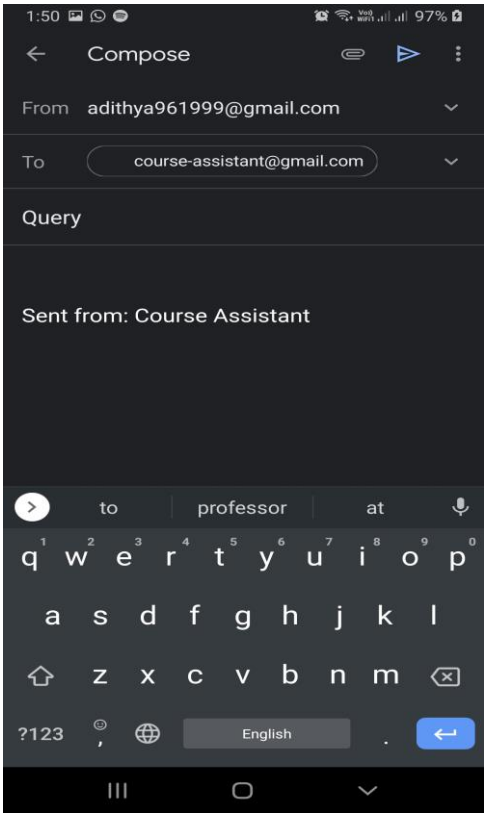

1:48
Add Notification
1:48 AM PM
CANCEL
OK

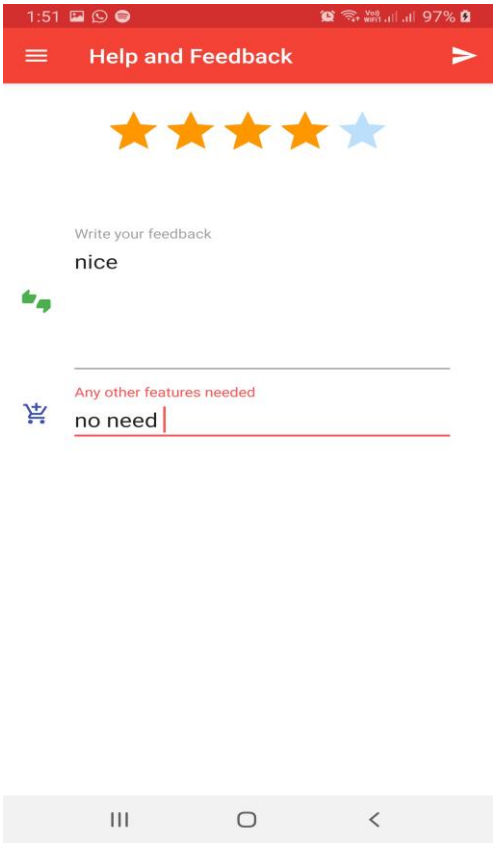

1:49
MAY 2020
M T W T F S S
Assignment
1001: Project


1:50
Course Assistant
Home
Similarity check
Request backup
Contact us
Help and feedback


1:50
Document Similarity
Document 1
Enter first document
Document 2
Enter second document
SIMILARITY
Result


1:50
97%
Courses
Software Engineering
1001
BACKUP ALL DATA


1:50
97%
Compose
From
adithya961999@gmail.com
To
course-assistant@gmail.com
Query
Sent from: Course Assistant
to
professor
at
English


1:51
97%
Help and Feedback
Write your feedback
nice
Any other features needed
no need

ACKNOWLEDGEMENT

I would like to express my gratitude to all those who gave me the possibility to complete this project.

I express my sincere gratitude to **Dr. D. PAUL DHAYABARAN, M.Sc., M.Phil., PGDCA., Ph.D., Principal,** Bishop Heber College (Autonomous), Trichy, for permitting me to do this project.

I wish to place on record my gratitude to **Dr. G. SOBERS SMILES DAVID, MCA., M.Phil., Associate Professor** & **Head**, Department of Computer Science, Bishop Heber College (Autonomous), Trichy, for his support during this project work.

I express my deep sense of gratitude to **Dr. R. JEMIMA PRIYADARSINI, MCA., M.Phil., Ph.D., Associate Professor,** Department of Computer Science, Bishop Heber College (Autonomous), Trichy, who has offered invaluable guidance and constant encouragement throughout this work.

I thank my friends for their help and innovative ideas during my project work. I also thank my parents for extending help in all aspects during the course of this project.

V.ADITHYA