DECLARATION

We **SURAJ R** bearing USN **4RA21CS097** student of 5th Sem B.E in **Computer Science and Engineering, Rajeev Institute of Technology, Hassan,** hereby declare that the work being presented in the dissertation entitled **"FACULTY FEEFBACK SYSTEM"** has been carried out by us under the supervision of guide **Mrs. Monika M.M,** Assistant Professor, Computer Science and Engineering, **Rajeev Institute of Technology, Hassan**, as partial fulfilment of requirement for the award of B.E Degree of **Bachelor of Engineering in Computer Science and Engineering at Visvesvaraya Technological University, Belagavi** is and authenticate record of my own carried out by us during academic year 2023-2024.

_____

SURAJ R

4RA21CS097

PLACE: HASSAN DATE:

……

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We would like to profoundly thank our **Management of RIT** & our President **Dr. RACHANA RAJEEV** for providing such a healthy environment.

We would like to express our sincere thanks to our principal **Dr. MAHESH P.K** Rajeev Institute of Technology for his encouragement.

We wish to express our gratitude to **Dr. Shrishail Math**, Head of the Department of CSE for providing a good working environment and for his constant support and encouragement

It gives us great pleasure to express our gratitude to **Mrs. Monika M.M**, Assistant Professor, Department of CSE for her expert guidance, initiative and encouragement that led us to complete this Mini project.

We would also like to thank all our staffs of Computer Science and Engineering department who have directly or indirectly helped us in the successful completion of this mini project and also, we would like to thank our parents.

SURAJ R
4RA21CS097

ABSTRACT

Faculty Feedback System is to provide feedback in an easy and quick manner to the college faculty. It is an interface between students and management for collecting feedback online. By using this technology, we can take feedback about the faculty by students fast and submit the same on time to head of departments as it is an online system. This project has two kinds of users Student, Administrator. The student can give feedback in online system provided by college staff. First of all, Administrator can prepare questions & add, update these questions to the online system. Those questions are viewed and answered by the students. These answers will be the feedback on that faculty individually. Each individual feedback is taken and consolidated as a report. This feedback report is used by the HODs to check the performance. The feedback report is the rating for each and every faculty given by students that can be used for further processing.

CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Introduction to database

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A **database-management system (DBMS)** is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyse data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Well-known DBMSs includes MySQL Microsoft SQL Server, Oracle, Sybase, SAP HANA, MySQL, SQLite and IBM DB2.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support; the most popular (database systems since the 1980s have all supported the relational model generally associated with the SQL language. Sometimes a DBMS is loosely referred to as a "database".

The technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational.

The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS,

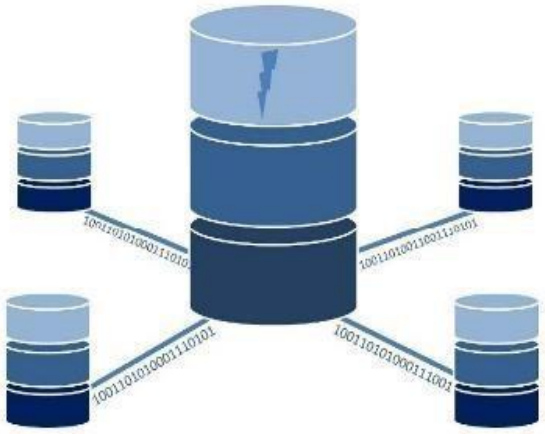

Fig 1.1: Sample figure of database

The relational model, first proposed in 1970 by Edgar F. Codd, departed from this tradition by insisting that applications should search for data by content, rather than by following links. The relational model employs set of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

Object databases were developed in the 1980s to overcome the inconvenience of object-relational impedance mismatch, which led to the coining of the term "post-relational" and also the development of hybrid object-relational databases.

The next generation of post-relational databases in the late 2000s became known as MySQL databases, introducing fast key-value stores and document-oriented databases. A competing "next generation" known as MYSQL databases attempted new implementations that retained the relational/SQL model while aiming to match the high performance of MySQL compared to commercially available relational DBMSs.

1.2 Introduction to Faculty Feedback System Management

This project is to develop web pages so that the traditional on paper feedback system can be done automated that is online.

1.2.1 OBJECTIVES

This project is about collecting feedback about the faculty from students effectively and in an easy way. FEEDBACK from customer/ client/ student/ user is the only way to develop the organization. In general organizations collect feedback from its clients to improve their services based on given feedback and suggestions. In case of educational institutions, it is mandatory to take feedback about the teaching staff so as to improve their standards of teaching. This feedback will help in internal improvement of standards and also external growth of the organization. This is because if we respond properly for the feedback and take some measures then the quality of education will be improved which will improve the placements, sponsorships, etc. So for an educational institution, taking feedback is an important thing.

1.2.2 MOTIVATION

The motivation for doing this project primarily is o To reduce the usage of papers for taking feedback from student individually. o To make feedback process effective. o To generate effective reports. o To reduce the time consumed for taking feedback.

1.2.3 PROBLEM DEFINITION

To find an effective and fast way to collect feedback about the faculty from the students of each class.

1.2.4 LIMITATIONS OF PROJECT

- All the students of same class must give feedback at same time.
- All the students must be logged in before any student completes giving the feedback.
- It has better look in latest browsers.

CHAPTER 2 FEASIBILITYSTUDY

After doing the project college Faculty Feedback system, study and analysing all the existing or required functionalities of the system, the next task is to do the feasible study for the project. All projects are feasible-given unlimited resources and infinite time.

Feasibility study included consideration of all the possible ways to provide a solution to the given problem. The proposed system should satisfy all the user requirements and should be feasible enough so that future changes can be easily done based on the future upcoming requirements.

2.1 Economical feasibility

This is very important aspect to be considered while developing project we decided the technology based on minimum cost factor.

All hardware and software cost as to be borne by the institution.

- Overall, we have estimated that the benefits the organisation is going to receive from the proposed system will surely overcome the initial cost and the later on running cost for the system.

2.2 Technical feasibility

This includes the study of function, performance and constraints that may affects the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, has described in the system requirement specification, and checked if everything was possible in different type of front end and backend platform.

2.3 Operational feasibility

No doubts the proposed system is fully user friendly and all inputs to be taken all selfexplanatory even to a layman. Asfar our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?

• Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

CHAPTER 3 SYSTEM SPECIFICATIONS

Software Requirement Specification is the starting point of the software developing activity. As system grew complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software is initiated by the client's needs.

3.1 SOFTWARE SPECIFICATIONS

- Apache Server
- My SQL
- Mozilla Firefox
- Internet Explorer 10
- Google Chrome

3.2 HARDWARE SPECIFICATIONS

- Intel Pentium 4 Processor
- 512 MB RAM
- 20 GB Hard Disk

3.3 Description of front end

3.3.1 Hypertext mark-up language

The Hyper Text Mark-up language (HTML) is a simple mark-up language used to create hypertext documents that are portable from one platform to another HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML version 4.0 HTML 4.01 aims to capture recommended practice as of early'96 and as such to be used as a replacement for HTML 3.2.

Why to use HTML

Website is a collection of pages, publications and documentation that reside on webserver. While these page publication and a document as a formatted in any single format you should use HTML for home page and all primary pages and the site. This will enable the millions of web users it considered first formatting any new material, you plan to publish the web HTML documents are platform independent, meaning that they don't conform to any standard it they are created properly you can more home page to any server platform or you can access them with any complaint www.browser.

• <HTML>…</HTML>

-All HTML files start and end with the tag pair.

• <HEAD> ...</HEAD>

- All HTML have a pair of --HEADI tags that indicate what the tile and other attributes of the page are going to be.

- <TITLE>...<TITLE>

-This tag indicates what the title of the HTML file is going to be on the BROWSER window title.

- <BODY>...</BODY>

-This tag pair is to logically separate the HTML file into the header and the body. Usually, the header contains information regarding Training & Placement System the html whereas the body contains information that the HTML file must actually contain.

- The HTML template must look like.

<! DOCTYPE HTML PUBLIC-THIS IS AN EXAMPLE>

<HTML><HEAD><TITLE> YOUR TITLE GOES

HERE</TITLE></HEAD></HTML>

•<P>...</P> ss- This tag pair used to indicate the paragraph. Any text that needs to be separated into a paragraph must be put in within a paragraph tag.

- <H1>Heading1</H1>

- <H2>Heading2</H2><H3>Heading3</H3>
  - This set of tags will show the Headings in smaller fonts as the heading increases.
- ALIGN

  -The align attribute can be used for headings as well. For <P>...</P> tags also, the ALIGN attribute can be used.
-
  - Used to insert a carriage return in the HTML file. The attribute to be used for this is the CLEAR attribute.
- <CENTER>...</CENTER>
  - To centre the entire block of text these tags are used.
- <A>...</A>
  - Anchor Tags. These tags are used linking namely hyperlinking.

3.3.2 Features of PHP

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The codes interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command- line interface capability and can be used in standalone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor to Microsoft's Active Server Pages (ASP) server- side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Webservers. Software that uses PHP includes Joomla, Word press, My BB, and Drupal. PHP was originally created by Ramus Leadoff in 1995. The main implementation of

PHP is now produced by The PHP Group and serves as the formal reference to the PHP language.

is free software released under the PHP License, which is incompatible with the GNU General Public

License (GPL) due to restrictions on the usage of the term

PHP While PHP originally stood for "Personal Home Page", it is now said to stand for "PHP: Hypertext Pre-processor", a recursive acronym

Description of back end

3.4 MySQL

MySQL is an object relational data base management system. It offers capabilities of both relational and object-oriented database system. In general object can be defined as reasonable software codes, which are location independent and perform a specific task on any application environment with little or no change.

Oracle products are based on a concept knows are client's server technology. This concept involves segregating the processing of the application between two systems. One performs all the activity related to database and the other performs the activity to interact with the application.

A client or a font end data base application also interact with the database by requesting the user and the database further it also checks four validations against the data entered by the user. The commonly used front-end tools of MySQL are MySQLBrowser.

CHAPTER 4 SYSTEM DESIGN

4.1 FLOW CHART

4.1.1 Flow Chart for Student Side

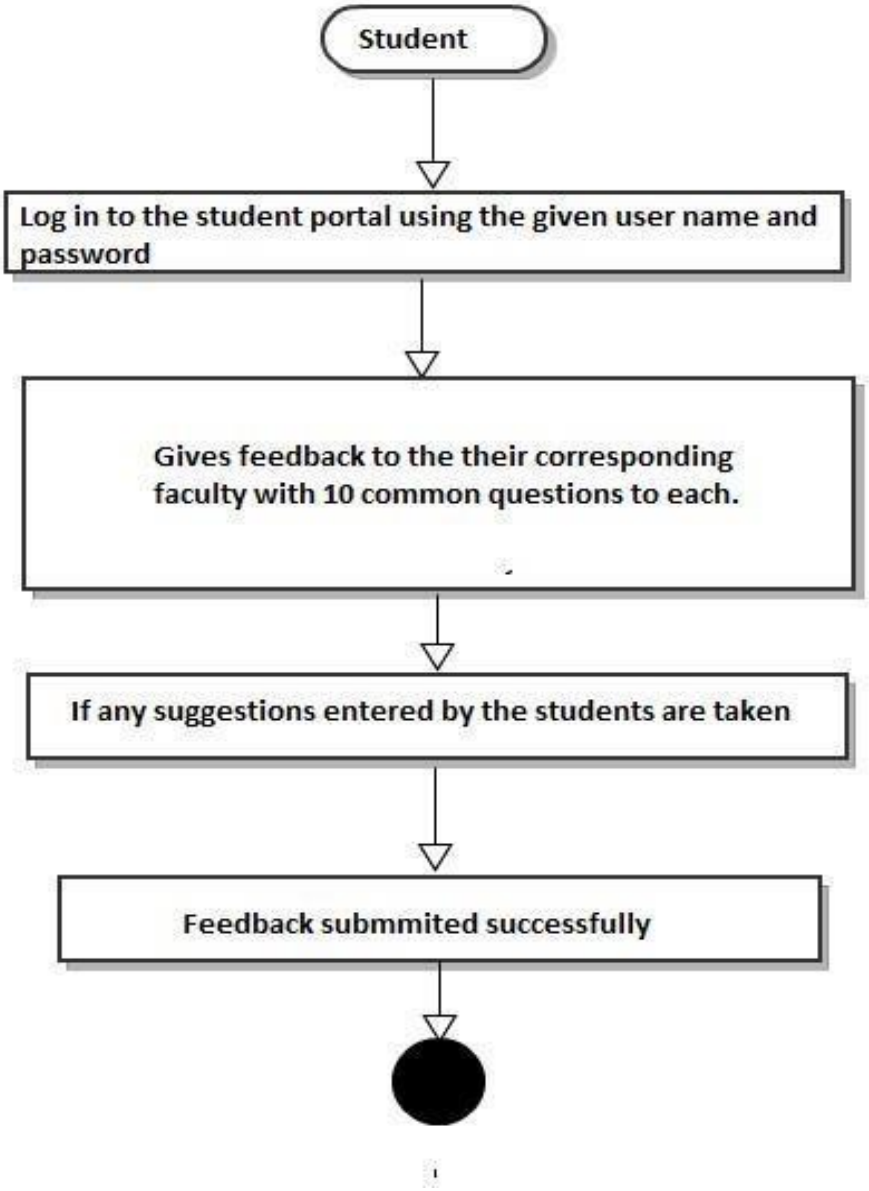

Fig 4.1.1: Flow chart for Student side

4
.1.2 Flow Chart for Administrator Side

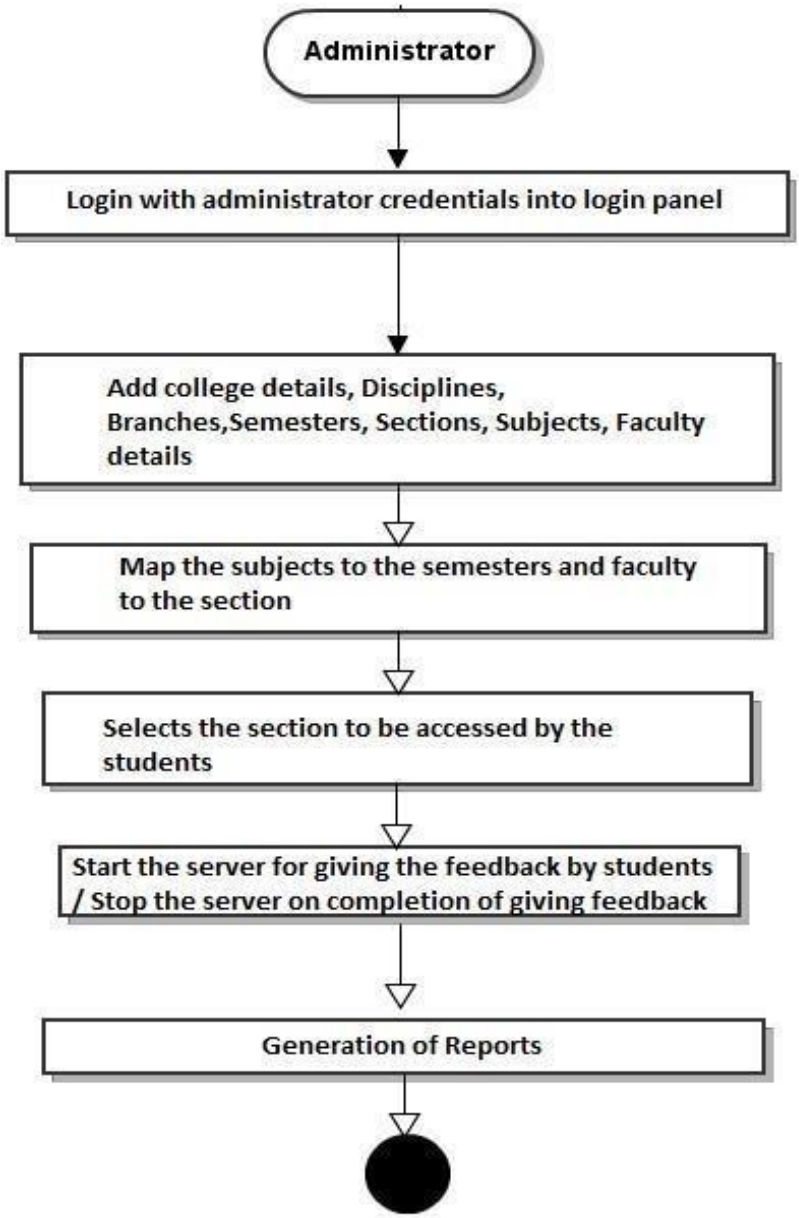

Fig 4.1.2: Flow chart for Administrator side

4

.2 ER DIAGRAM

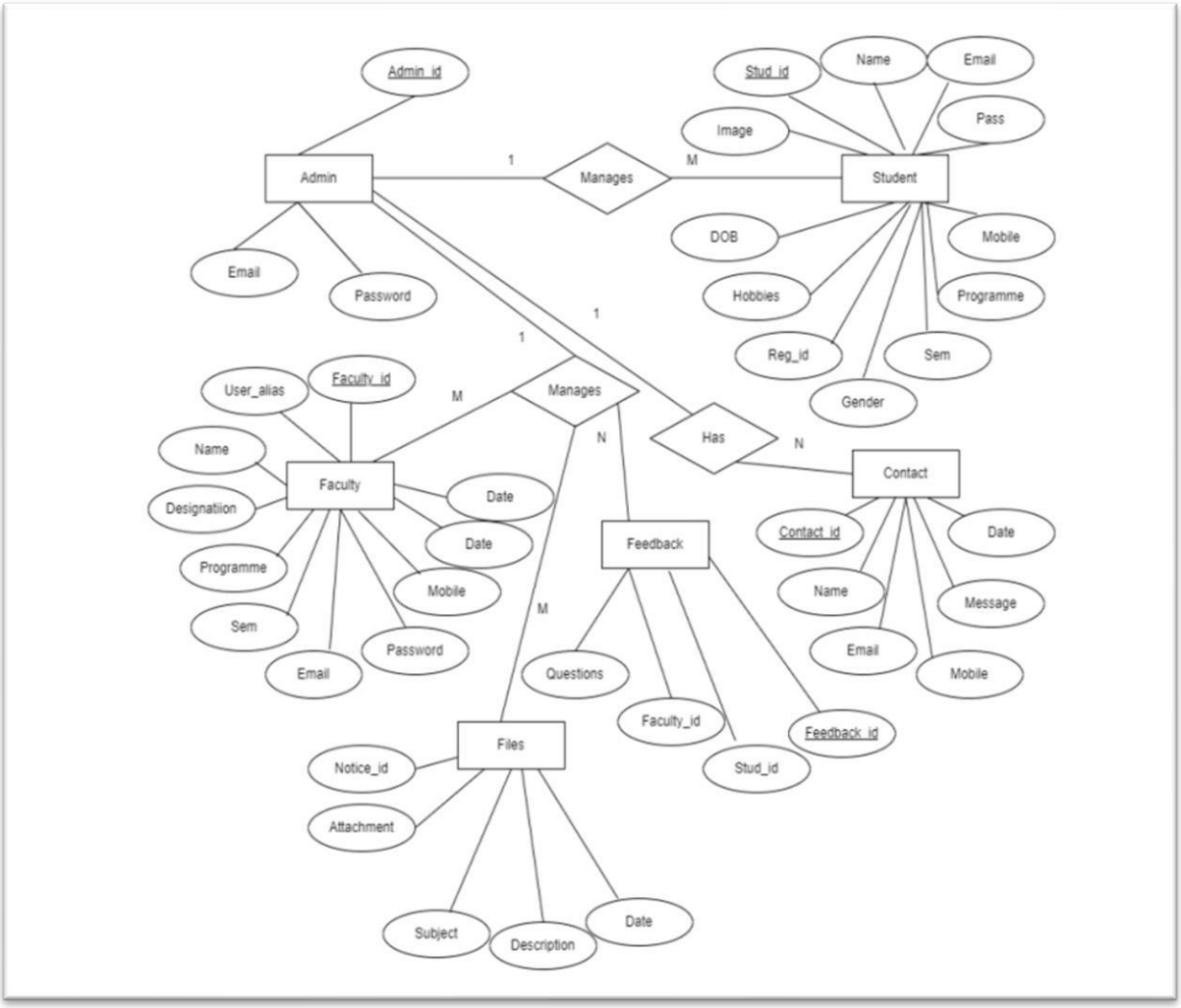

Fig 4.2: Entity relationship diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a company of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of these symbols, that make it unique. Entities may be characterized not may be relationships, but also by additional properties (attributes), which include identifiers called “primary keys”. Diagrams created to represent attributes as well as entities and relationships may be called entity-attributes-relationship diagrams, rather than entity-relationships models.

4

.3 Schema Diagram

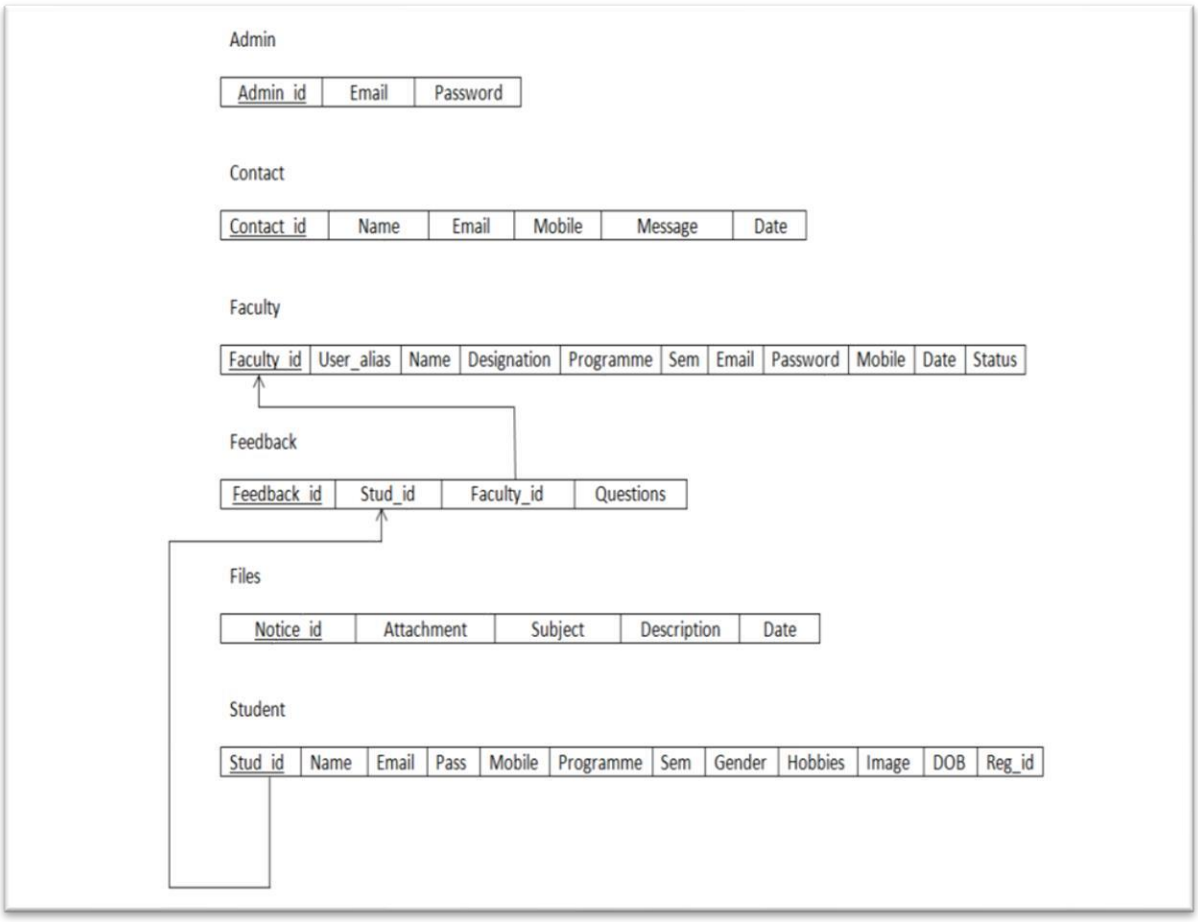

Fig 4.1: Schema diagram

4.3 Table description Table

Name: admin

Primary Key: admin_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	admin_id	int(11)			No	*None*		AUTO_INCREMENT
2	user	varchar(50)	latin1_swedish_ci		No	*None*		
3	pass	varchar(50)	latin1_swedish_ci		No	*None*		

4

Table 4.3.1: admin table description

Table Name: contact

Primary Key: contact_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	name	varchar(50)	latin1_swedish_ci		No	None		
3	email	varchar(100)	latin1_swedish_ci		No	None		
4	mobile	bigint(20)			No	None		
5	message	text	latin1_swedish_ci		No	None		
6	Date	datetime			No	None		

Table: 4.3.2: contact table description

Table Name: faculty

Primary key: faculty_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	user_alias	varchar(30)	latin1_swedish_ci		No	None		
3	Name	varchar(30)	latin1_swedish_ci		No	None		
4	designation	varchar(100)	latin1_swedish_ci		No	None		
5	programme	varchar(50)	latin1_swedish_ci		No	None		
6	semester	varchar(10)	latin1_swedish_ci		No	None		
7	email	varchar(255)	latin1_swedish_ci		No	None		
8	password	varchar(75)	latin1_swedish_ci		No	None		
9	mobile	bigint(20)			No	None		
10	date	datetime			No	None		
11	status	int(11)			No	None		

Table: 4.3.3: faculty table description

Table Name: feedback

Primary key: feedback_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	student_id	varchar(50)	latin1_swedish_ci		No	None		
3	faculty_id	varchar(50)	latin1_swedish_ci		No	None		
4	ques1	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
5	ques2	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
6	ques3	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
7	ques4	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
8	ques5	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
9	ques6	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
10	ques7	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
11	ques8	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
12	ques9	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
13	ques10	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
14	ques11	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
15	ques12	enum('5', '4', '3', '2', '1')	latin1_swedish_ci		No	None		
16	ques13	text	latin1_swedish_ci		No	None		
17	ques14	text	latin1_swedish_ci		No	None		
18	date	date			No	None		

Table: 4.3.3: feedback table description

Table Name: files

Primary key: notice_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	notice_id	int(11)			No	None		AUTO_INCREMENT
2	attachment	varchar(255)	latin1_swedish_ci		No	None		
3	subject	varchar(100)	latin1_swedish_ci		No	None		
4	Description	text	latin1_swedish_ci		No	None		
5	Date	datetime			No	None		

Table: 4.3.3 files table description

Table Name: user

Primary key: user_id

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	name	char(40)	latin1_swedish_ci		No	None		
3	email	varchar(40)	latin1_swedish_ci		No	None		
4	pass	varchar(40)	latin1_swedish_ci		No	None		
5	mobile	bigint(11)			No	None		
6	programme	varchar(20)	latin1_swedish_ci		No	None		
7	semester	varchar(10)	latin1_swedish_ci		No	None		
8	gender	varchar(40)	latin1_swedish_ci		No	None		
9	hobbies	varchar(40)	latin1_swedish_ci		No	None		
10	image	varchar(50)	latin1_swedish_ci		No	None		
11	dob	date			No	None		
12	regid	varchar(20)	latin1_swedish_ci		No	None		

Table: 4.3.3: user table description

CHAPTER 5 DESIGN

5.1 DESIGN

The architecture of this feedback consists of 8 main modules. The divisions and their connections between them is shown in fig 1. In this first the discipline and its duration are taken and branches are added to corresponding discipline and each branch is divided into number of years given during creation of discipline and to branches, faculty and subjects are added. Under the year module section and semesters are added to it. To the section, subject and faculty are mapped and make ready for the accessing of a particular section.

After accessing the reports are generated and saved locally.

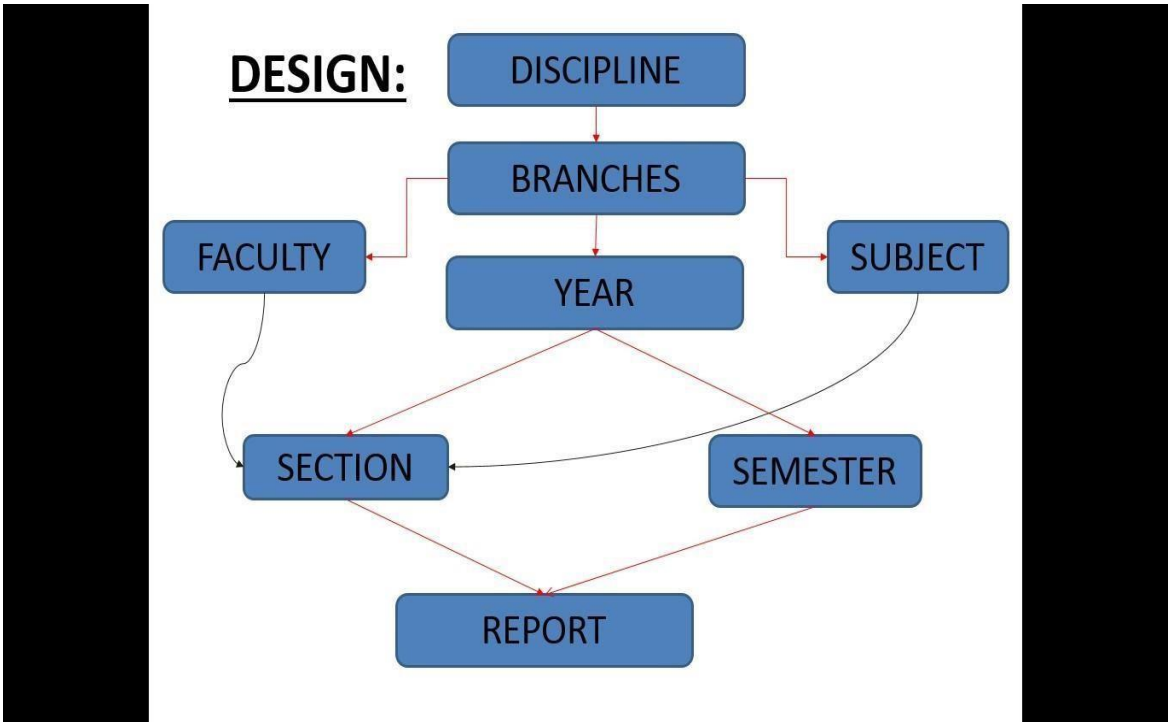

Fig 5.1: Design of the Project

5.2 EXAMPLE

The below given design is an instance of the project design for a particular section. In this example for section A 1st year 1st semester branch IT under discipline B. Tech the faculty are mapped to corresponding subjects and reports are generated after the students have given feedback.

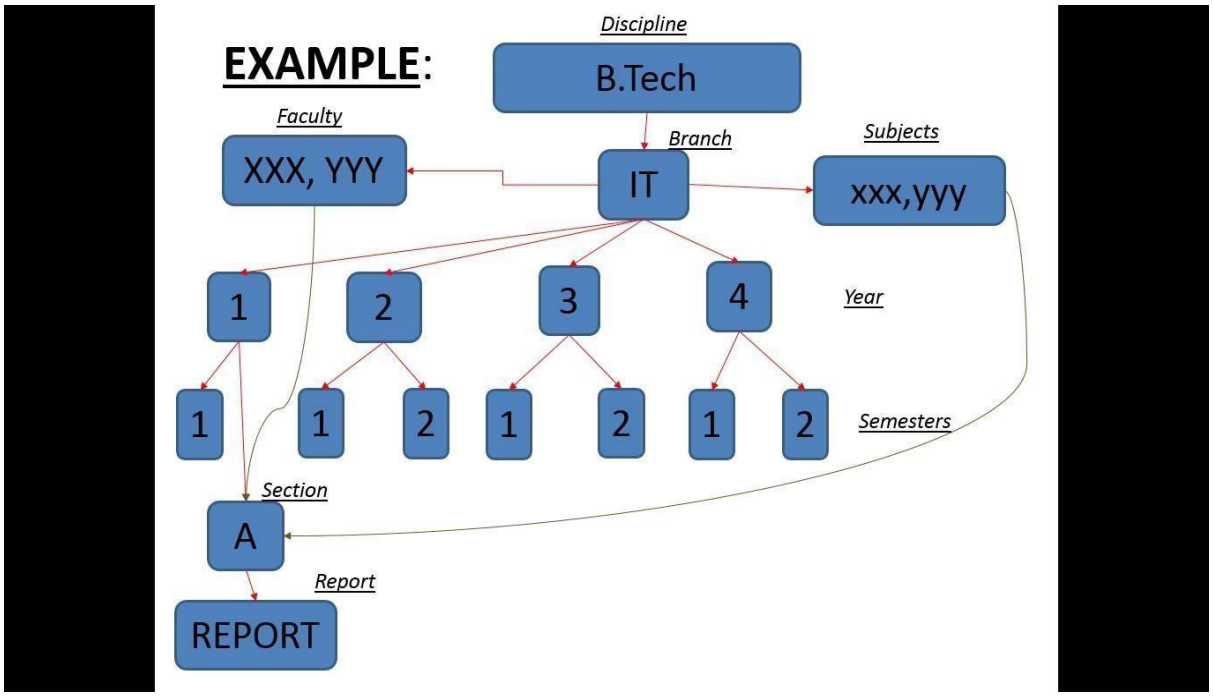

Fig 5.2: Instance of the design

SNAPSHOTS

6.1 HOME PAGE

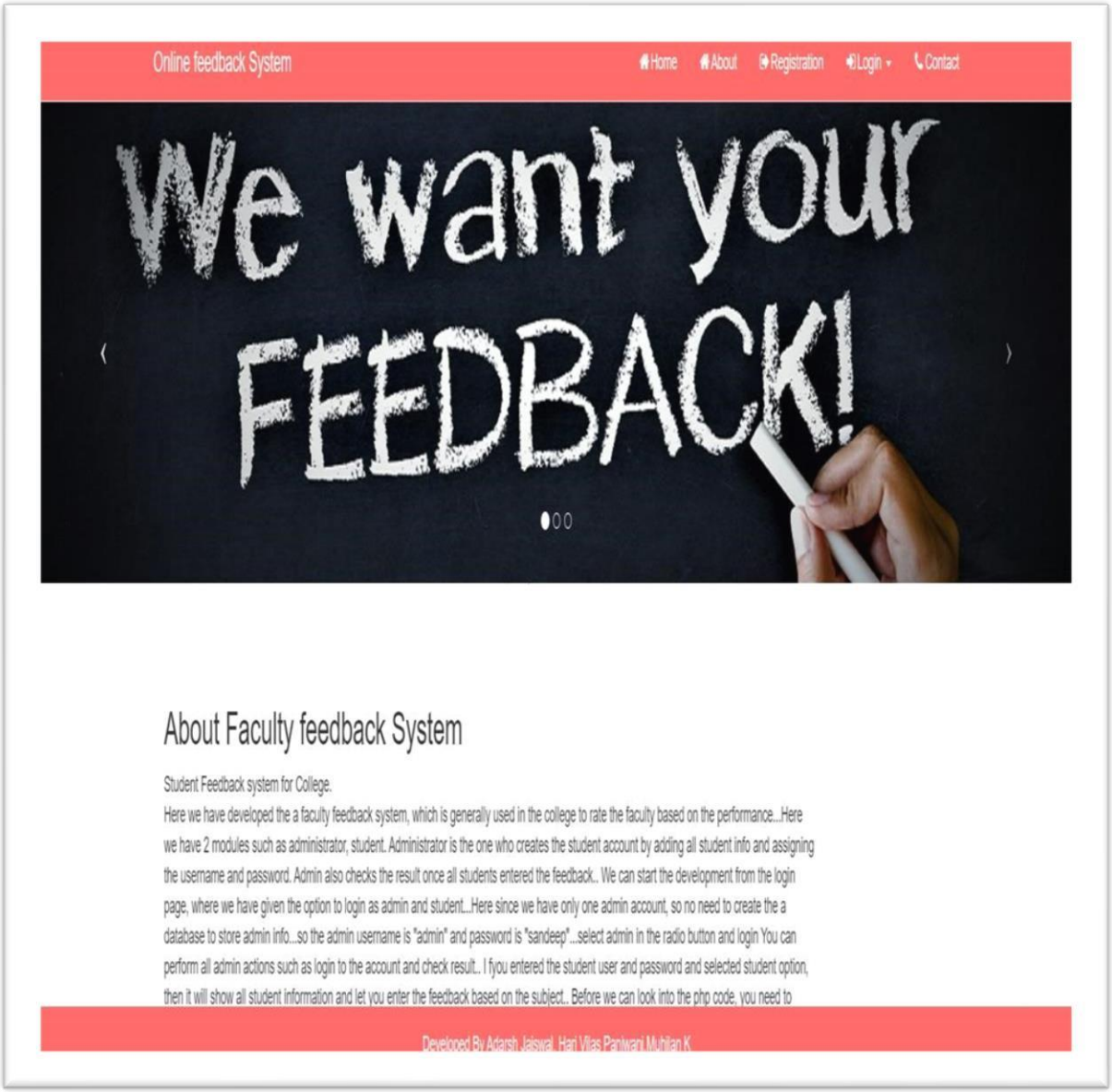

Fig 6.1: Home Page

6.2 Admin login page

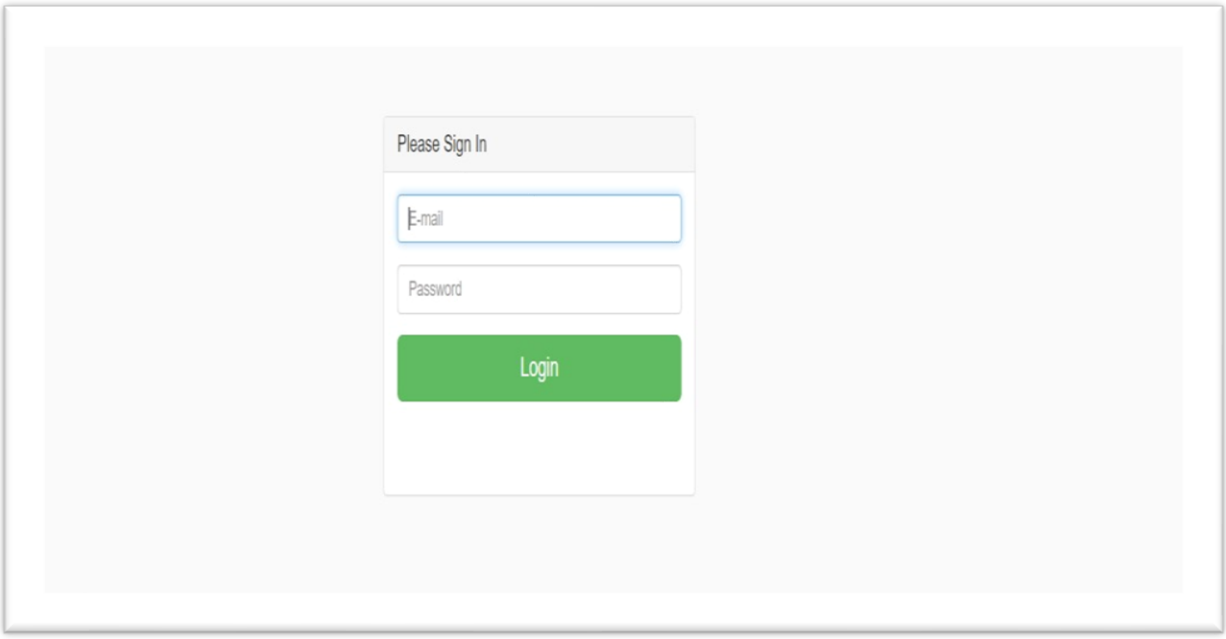

Fig 6.2: Admin login page

6.3 Faculty login page

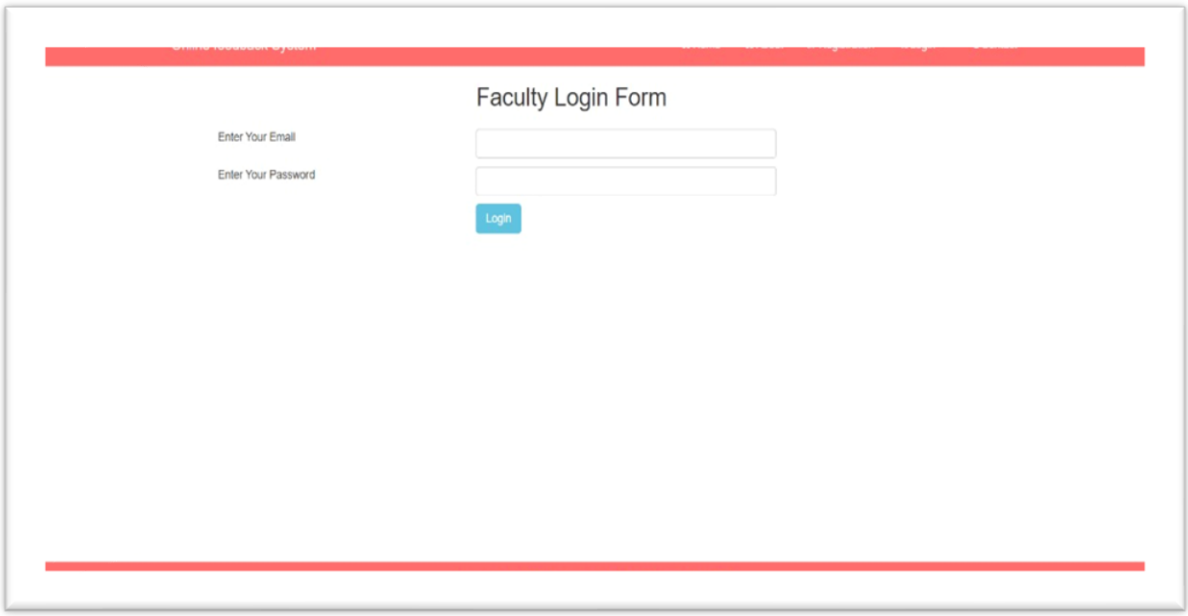

Fig 6.3: Faculty login page

6 .4 Registration Form

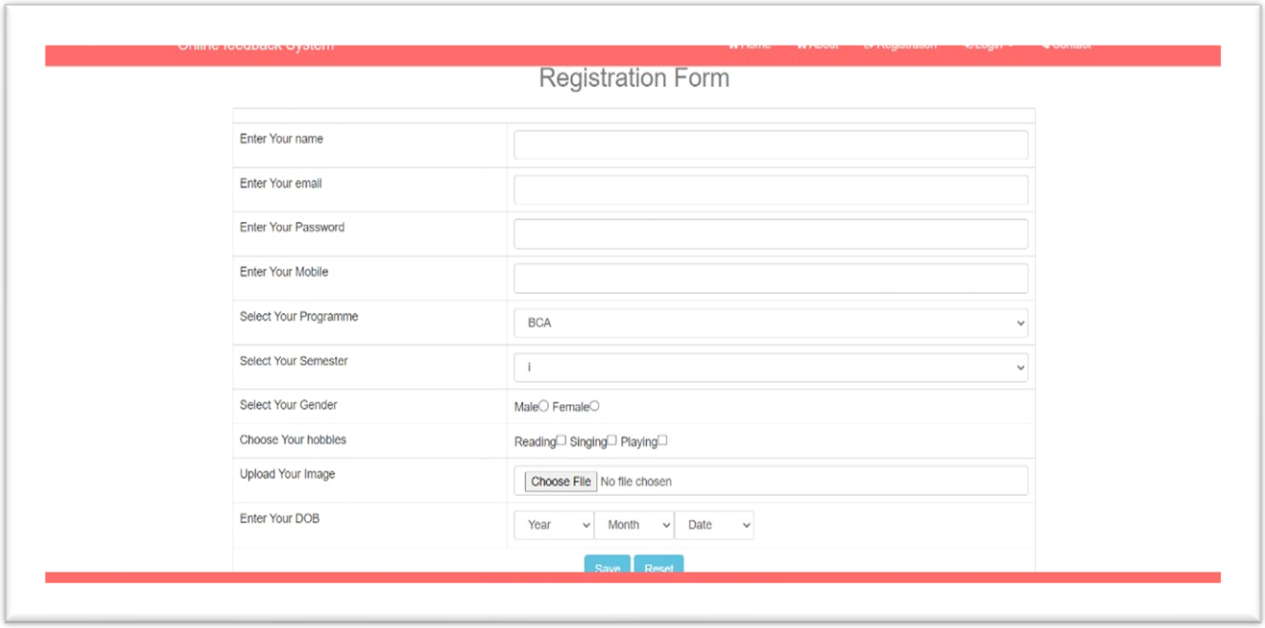

Fig 6.4: Registration Form

6.5 Student login Form

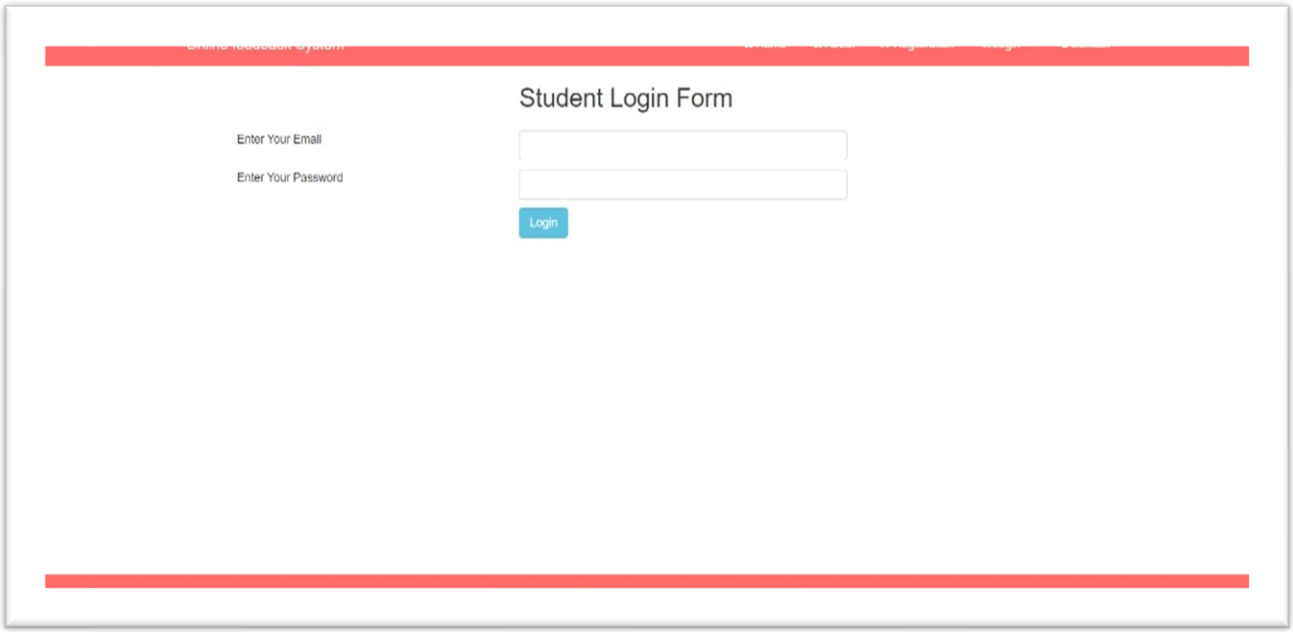

Fig 6.5: Student login

6
.6 Admin dashboard

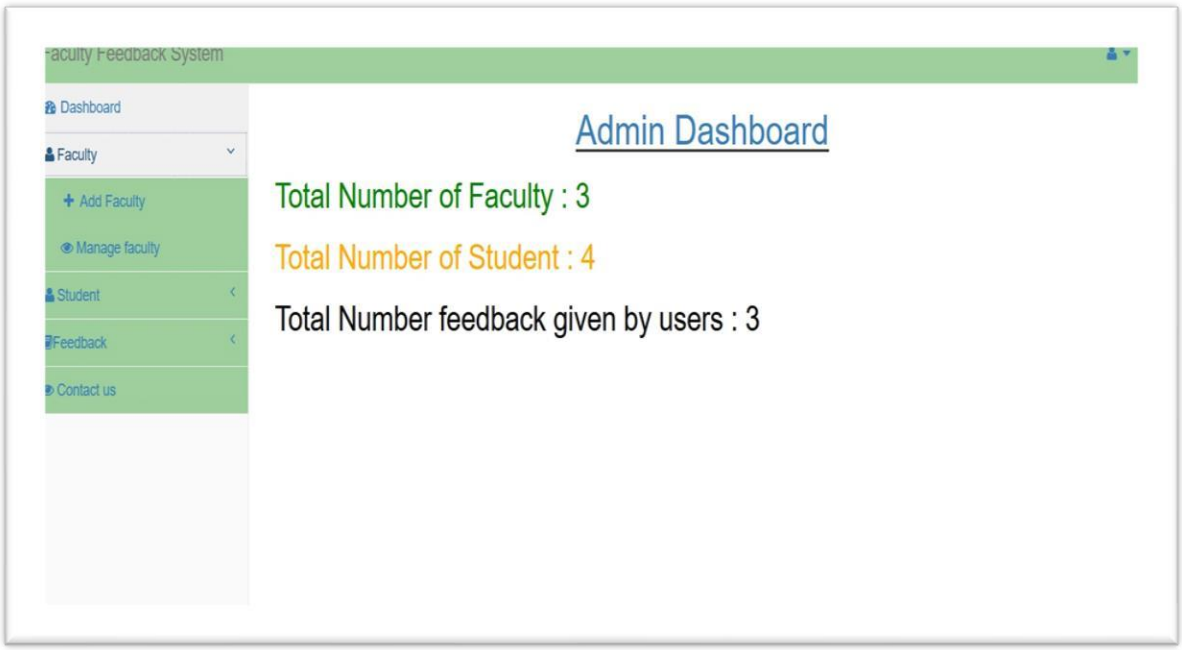

Fig 6.6: Admin Dashboard

6.7 Add Faculty

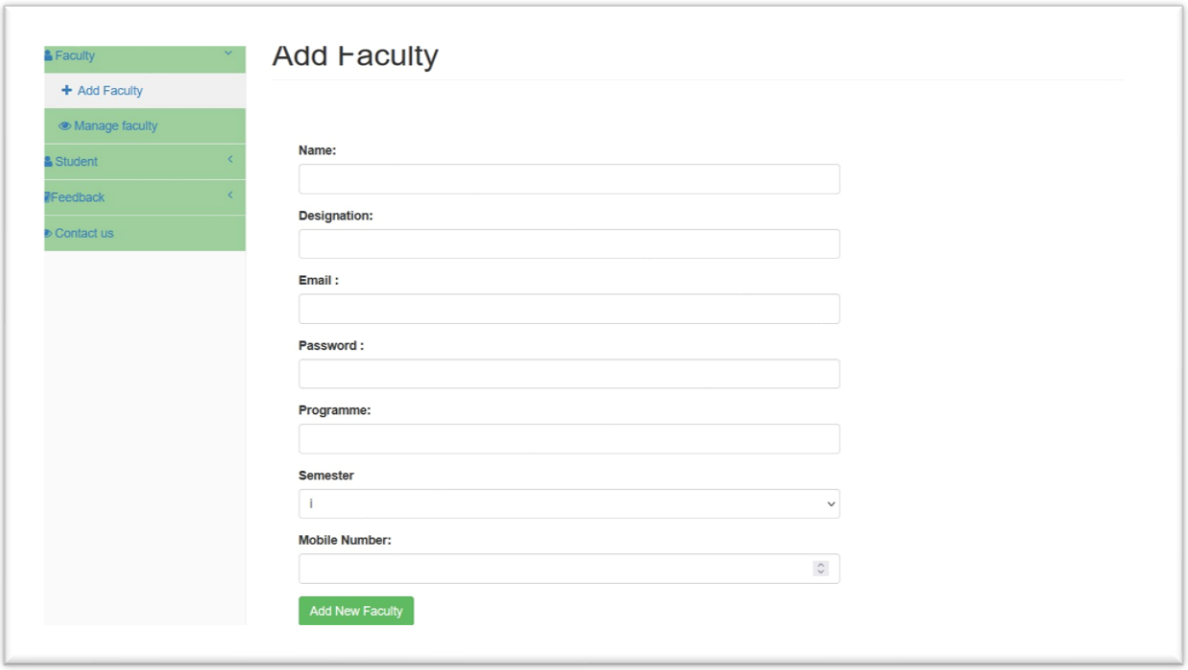

Fig 6.7: Add Faculty

6

.8 Manage Student

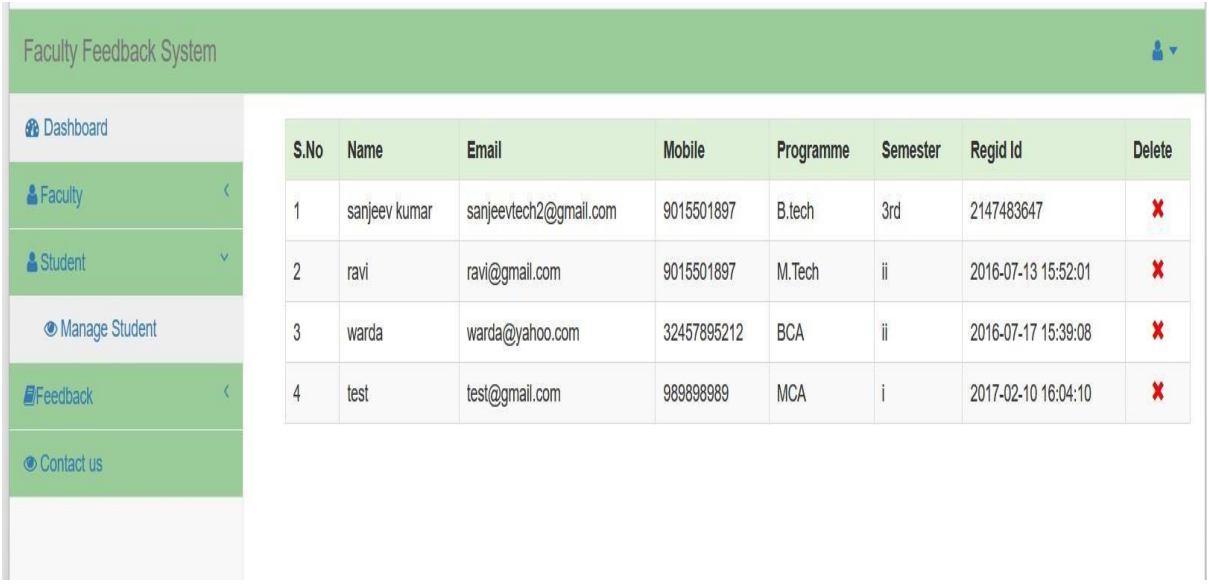

Fig 6.8: Manage Student

6.9 Student Feedback Form

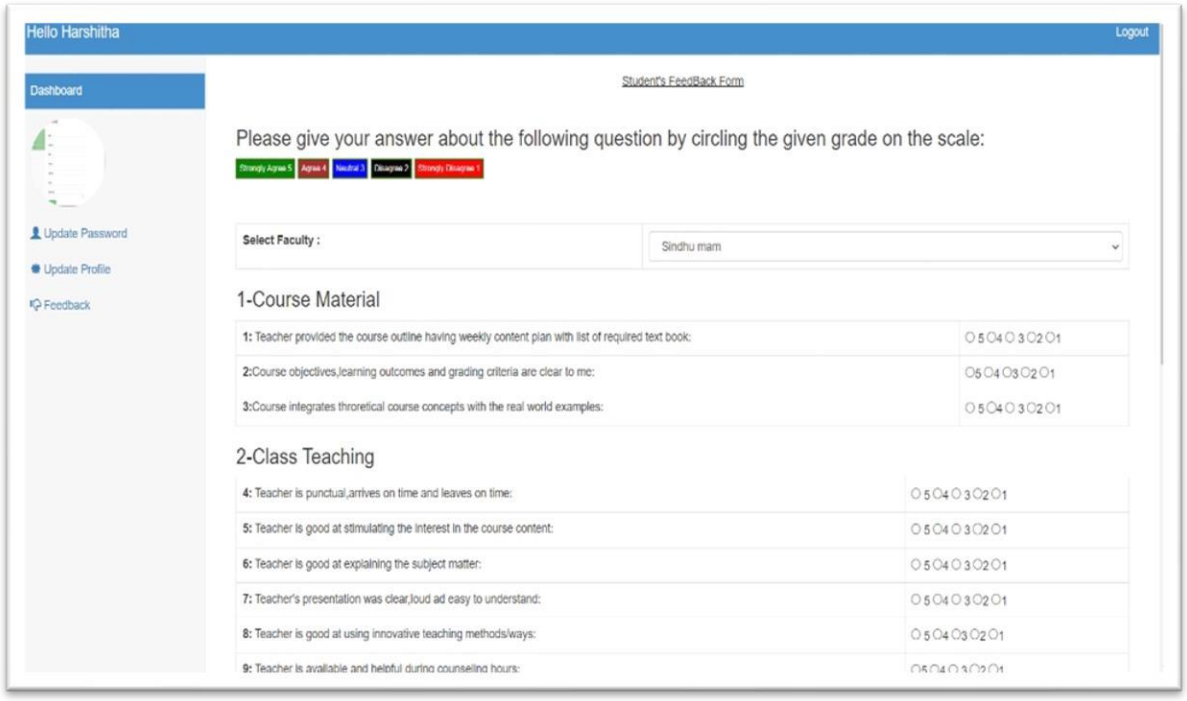

Fig 6.9: Student Feedback Form

6

.10 Student Feedback

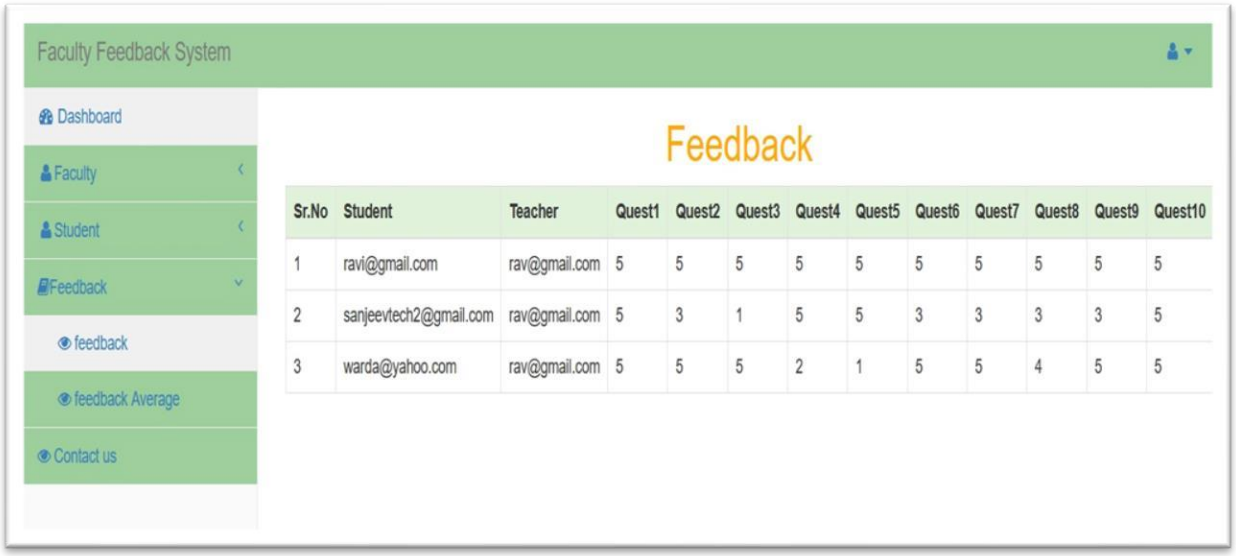

Fig 6.10: Student Feedback

CONCLUSION

Finally, we conclude that our mini project student feedback system is an efficient system to take the feedback from the students and generate report for each faculty, consolidation report for the entire class. These reports help the management and faculty to effectively teach students. These systems are tested on different test cases and showed positive result. By facilitating transparent feedback exchange, the system has enabled administrators to make data-driven decisions. Moving forward, ensuring the sustainability and scalability of the system will be crucial, along with fostering a culture of continuous improvement. Overall, the Faculty Feedback System represents a significant advancement towards creating a more response.

REFERENCES

- http://stackoverflow.com/questions/
- http://w3schools.com/
- http://fpdf.org/
- www.github.com