

# **STUDENT FEEDBACK SYSTEM**

## **A MINI PROJECT REPORT**

[ CASE STUDY ]

*Submitted by*

**Pranav Srivastava [RegNo:RA2211003010647]**

**Devraj Goswami [RegNo: RA2211003010612]**

*Under the Guidance of*

**Rajalakshmi M**

[ Assistant Professor, Department of Computing Technologies ]



**DEPARTMENT OF COMPUTING TECHNOLOGIES  
FACULTY OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
KATTANKULATHUR– 603 203**

**NOV 2024**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**KATTANKULATHUR-603 203**  
**BONAFIDE CERTIFICATE**

Certified that **21CSE354T – FULL STACK WEB DEVELOPMENT - Mini project report**

titled “**STUDENT FEEBACK SYSTEM**” is the bonafide work of Pranav Srivastava  
[RegNo:RA2211003010647] and Devraj Goswami [Reg No:RA2211003010612]

who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation.

**Rajalakshmi M**

**COURSE FACULTY**

Assistant Professor

Department of Computing Technologies

**Dr G. NIRANJANA**

**HEAD OF THE DEPARTMENT**

Department of Computing Technologies

**DEPARTMENT OF COMPUTING TECHNOLOGIES**  
**SCHOOL OF COMPUTING**  
**College of Engineering and Technology**  
**SRM Institute of Science and Technology**

**MINI PROJECT REPORT**

**ODD Semester, 2024-2025**

**Subject code &**

**Sub Name** : 21CSE354T & Full stack Web Development

**Year & Semester:**

III & V

**Project Title** :

Student Feedback System

**Course Faculty Name**

: Rajalakshmi M

**Team Members** :

Pranav Srivastava[RA2211003010647], Devraj Goswami[RA2211003010612]

Particulars	Max. Marks	Marks Obtained
FRONT END DEVELOPMENT	2.5	
BACK END DEVELOPMENT	2.5	
IMPLEMENTATION	3	
REPORT	2	

**Date** :

**Staff Name** :

**Signature** :

## ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to **Dean-CET**, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman, Professor and Chairperson**, School of

Computing and **Dr M. Pushpalatha, Associate chairperson**, Department of Computing Technologies, SRM Institute of Science and Technology for their invaluable support.

We are grateful to our **Head of the Department, Dr G. Niranjana, Professor**, Dept of CTECH, SRMIST, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our **Audit Professor, Dr R. K Pongiannan, Professor**, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

Our inexpressible respect and thanks to our course faculty, **Rajalakshmi M**, Assistant Professor, Department of CTECH, SRMIST, for providing us with an opportunity to pursue our project under [his / her](#) mentorship. [He / She](#) provided us with the freedom and support to explore the research topics of our interest. [His /Her](#) passion for solving problems and making a difference in the world has always been inspiring.

**Pranav Srivastava[Reg. No:RA2211003010647]**

**Devraj Goswami[Reg. No: RA2211003010612]**

# TABLE OF CONTENTS

	<b>ABSTRACT</b>	<b>6</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>7</b>
<b>2.</b>	<b>PROJECT OVERVIEW AND OBJECTIVES</b>	<b>8</b>
	2.1 Objectives	
<b>3</b>	<b>ARCHITECTURE DIAGRAM AND TECHNOLOGIES USED</b>	<b>9</b>
	3.1 Architecture Diagram	
	3.2 Frontend Design	
	3.3 Backend Design	
<b>4</b>	<b>PROJECT PLANNING</b>	<b>21</b>
	4.1 Requirements	
	4.2 Database Design	
<b>5</b>	<b>FRONTEND DEVELOPMENT</b>	<b>30</b>
	5.1 HTML/CSS Structure	
	5.2 JavaScript and Frontend Frameworks (React JS)	
<b>6</b>	<b>BACKEND DEVELOPMENT</b>	<b>34</b>
	6.1 • API Design, Authentication, and Database Connectivity	
	6.2 Database Management – Spring boot Concepts and CRUD Operations	
<b>7</b>	<b>TESTING AND DEPLOYMENT</b>	<b>41</b>
<b>8.</b>	<b>RESULTS AND CONCLUSION</b>	<b>46</b>
	6.1 Conclusion	
	6.2 Future Scope	
	<b>REFERENCES</b>	<b>48</b>
	<b>APPENDIX 1 - CODE</b>	<b>49</b>
	<b>APPENDIX 2</b>	<b>50</b>

## **ABSTRACT**

Online feedback system is web based system which provides a way for colleges to allow students to gives feedback for staff online to improve their teaching.Students are requires to gives feedback using one standard feedback form. In our project, the security is also maintain by result of feedback is only visible to authentic user. This project also includes time portal. This system helps teachers to improve the performance by analyzing the feedback given by student. The main aim of this project is to apply the knowledge the of MySQL taught under DBMS Subject in real life project.This software package has been developed using the powerful coding tools of HTML, CSS, Bootstrap at the Front End and PHP, MySQL Server at Back End.The software is very user friendly. This version of the software has multi-user approach. For further enhancement or development of the package, user's feedback will be considered.

## **1. INTRODUCTION**

The Online Feedback System is used to manages feedback provided by students. Online Feedback System allows students to select particular subject and respective teacher to give feedback about teacher and subject. A Online Feedback System is an feedback generation system which gives proper feedback to teacher provides the proper feedback to the teachers about their teaching quality on basis of rating very poor, poor, average, good, very good. In the existing system students requires giving feedback manually. In existing system report generation by analyzing all feedback form is very time consuming. By online feedback system report generation is consumes very less time. In online feedback system student gives feedback for teacher of particular subject for particular period of time may be at month end. Feedback is send to HOD of particular department as well as all departments' feedback to principal. HOD has rights to whether feedback shows to respected teacher or not. After analyzing report HOD or principle conducts the meetings for staff by send mail to them.

## **2. PROJECT OVERVIEW AND OBJECTIVES**

### **2.1 Objectives**

In the existing feedback system, collecting feedback on lecturers relies entirely on a manual, paper-based process. At the end of a specified period, students are required to provide feedback by filling out forms with pen and paper, commenting on various aspects of their lecturers' teaching quality, subject expertise, interaction levels, and overall effectiveness. Once all feedback forms are completed, they are handed over to the Head of Department (HOD). The HOD then undertakes the labor-intensive task of aggregating responses, calculating average scores or grades for each lecturer, and summarizing the overall ratings for each subject. This process, being completely manual, is time-consuming, prone to human error, and requires considerable resources in terms of time and manpower. Following the consolidation of feedback, the final grade reports, which provide an overview of each lecturer's performance, are submitted to the principal. This principal then uses the grades to assess the strengths and areas for improvement of each lecturer. With these insights, appropriate guidance or counseling sessions are arranged to address any identified shortcomings, which aims to enhance the quality of teaching. However, the inefficiency of this manual feedback system presents challenges in promptly addressing students' concerns and implementing timely improvements.



### 3. ARCHITECTURE DIAGRAM AND TECHNOLOGIES USED

#### 3.1 Architecture Diagram

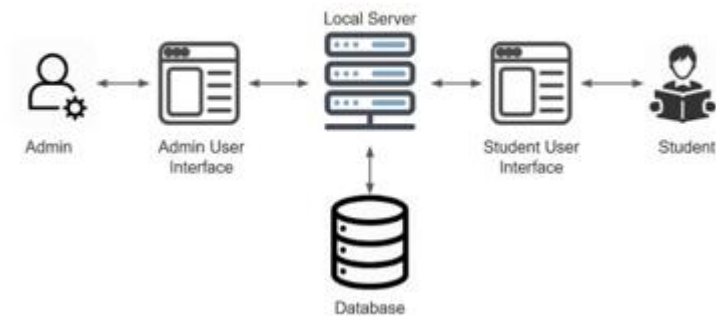


Fig: 3.1 System Architecture

- Student - initiates the interaction by asking questions and receiving answers.
- Question and Answer - This could be a portal or a database where students can access information or submit questions.
- Admin Login Portal - teachers could access this portal to answer student questions and manage the system.
- Feedback stored in DB - This could represent the process of storing questions, answers, and feedback for future use.
- information and interactions between students and teachers.

## 3.2 Frontend Design

### 3.2.1 Student Module

- **Registration Page**

Any new student to this web-portal must first register himself. Without registration he/she can not use this web-portal. Student can register into the web-portal by just proving his/her basic details like name, email-id, semester, program etc..

A screenshot of a web application's registration form. The form is titled "Registration Form" and is overlaid on a background image of graduates in caps and gowns. The form fields include: Name, Email Address, Mobile Number, Program, Semester, Gender, a file upload button labeled "Choose File" with the text "No file chosen", a date field with the placeholder "mm/dd/yyyy", and a Password field. The top left of the page says "Welcome to Online Feedback System" and the top right has a "Menu" icon.

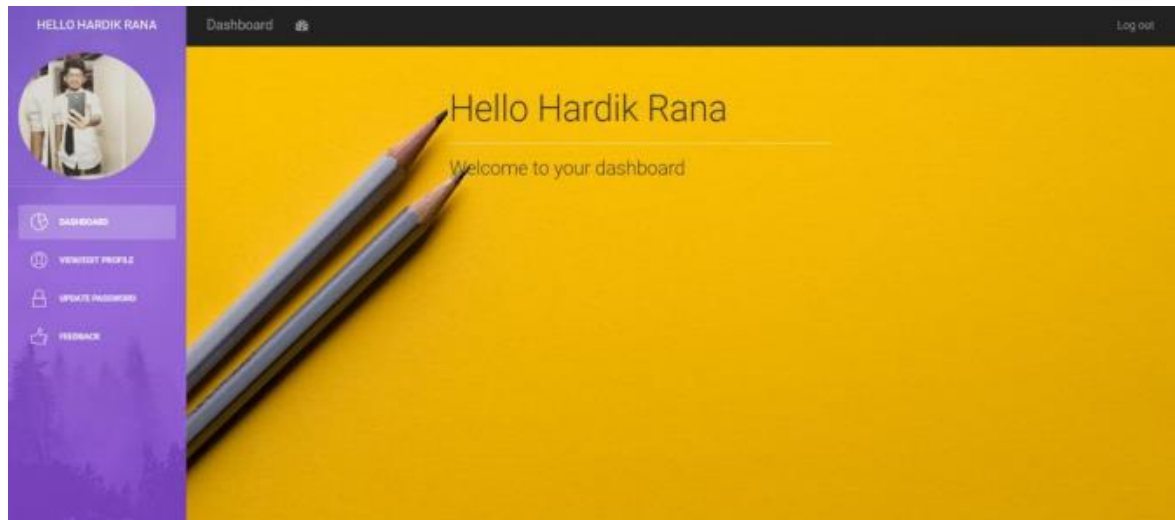
- **Login Page**

After successful registration any student can register into this web-portal by login into the login page. While doing login he need to just provide email-address and password he/she used while doing registration.

A screenshot of a web application's student login page. The page is titled "Student Login" and is overlaid on a background image of an open book on a wooden desk. The login form contains two input fields: "Email Address" and "Password", followed by a blue "Login" button. The top left of the page says "Welcome to Online Feedback System" and the top right has a "Menu" icon.

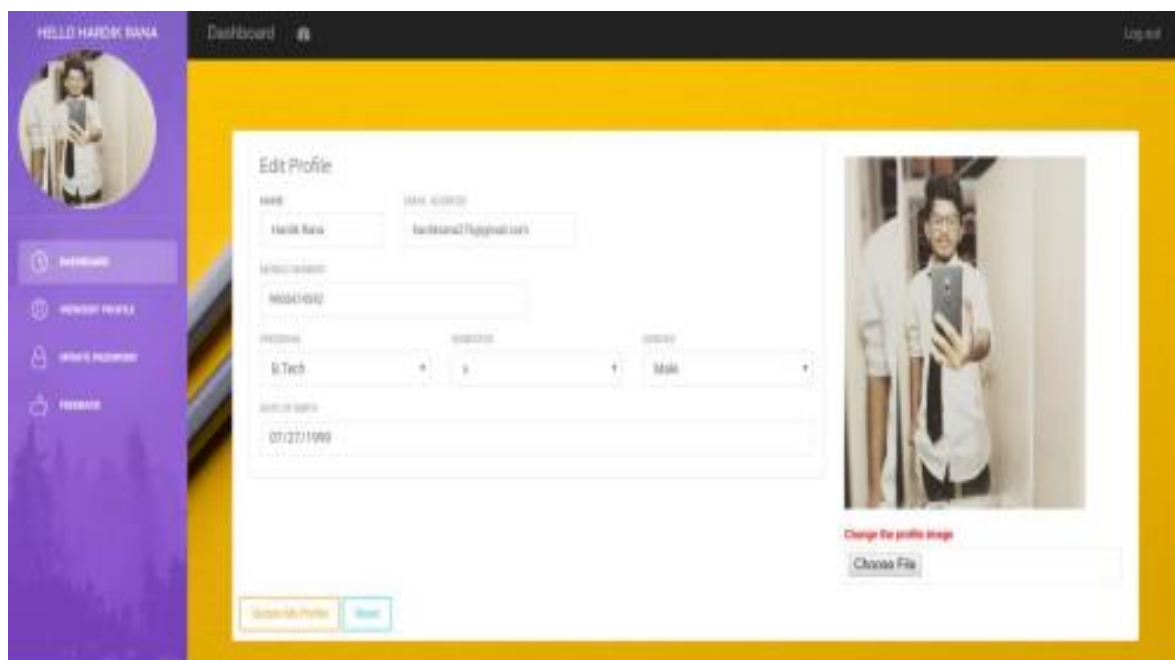
- **Dashboard**

After successful login a student will be redirected to his/her dashboard [which will look like this].



- **View/Edit Profile**

In this section a student can view and can also update it's details.



## ● Update Password

In this section a student can update his/her password by proving old password.

The screenshot shows a student dashboard with a purple sidebar and a yellow main area. The sidebar contains the user's name 'HELLO HARDIK RANA', a profile picture, and navigation links: 'Dashboard', 'VIEW MY PROFILE', 'UPDATE PASSWORD', and 'FEEDBACK'. The main area displays the 'Update Password' form, which includes three input fields: 'ENTER YOUR OLD PASSWORD', 'ENTER YOUR NEW PASSWORD', and 'ENTER YOUR CONFIRM PASSWORD'. Below the fields are two buttons: 'Update My Password' and 'Reset'.

## ● Give Feedback

In this section student can give feedback to the faculty he/she has selected. We will show him only the professors which are taking courses in their semesters.

The screenshot shows the 'Give Your Feedback' form in the student dashboard. At the top, a message says 'Please give your answer about the following question by circling the given grade on the scale.' Below this are five colored buttons: 'Strongly Agree 5', 'Agree 4', 'Neutral 3', 'Disagree 2', and 'Strongly Disagree 1'. The form has a 'SELECT FACULTY' dropdown menu with 'Manu Batvaraju' selected. It is divided into two sections: '1.Course Material' and '2-Class Teaching'. Each section contains three statements with corresponding rating scales (5, 4, 3, 2, 1) for feedback.

Section	Statement	Rating Scale
1.Course Material	Teacher provided the course outline having weekly content plan with list of required text book:	5 4 3 2 1
	Course objectives, learning outcomes and grading criteria are clear to me:	5 4 3 2 1
	Course integrates theoretical course concepts with the real world examples:	5 4 3 2 1
2-Class Teaching	Teacher is punctual, arrives on time and leaves on time:	5 4 3 2 1
	Teacher is good at stimulating the interest in the course content:	5 4 3 2 1
	Teacher is good at explaining the subject matter:	5 4 3 2 1
	Teacher's presentation was clear, loud and easy to understand:	5 4 3 2 1
	Teacher is good at using innovative teaching methods/ways:	5 4 3 2 1

### 3.2.2 Faculty Module

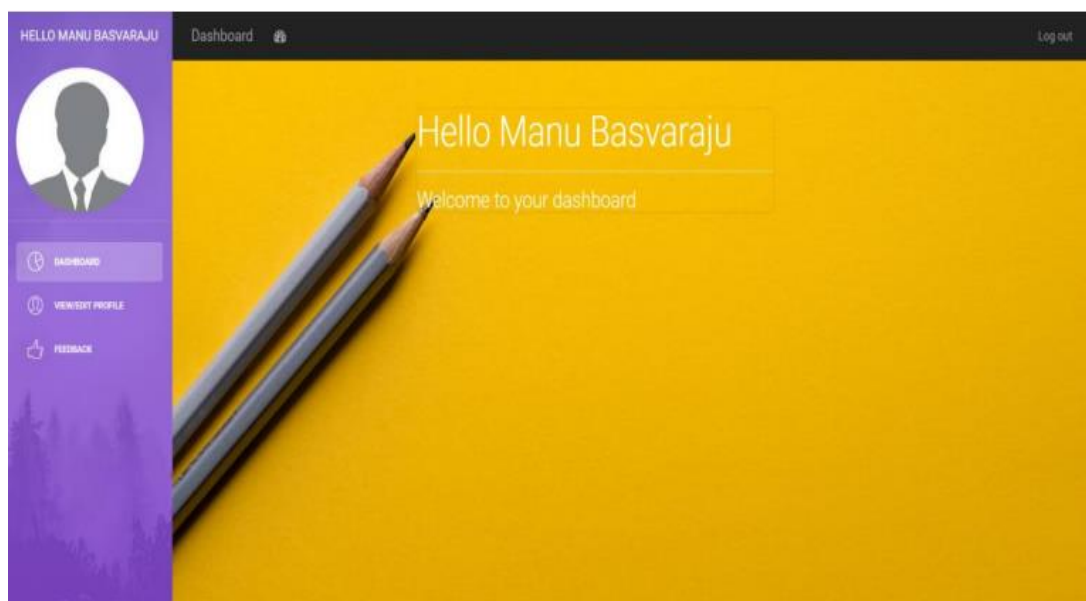
- **Login Page**

In our system admin will add faculty into the system. After that with that email-id and password a faculty can login into the system.



- **Dashboard**

After successful login a faculty will be redirected to his/her dashboard [which will look like this].



## ● View/Edit Profile

In this section a faculty can view and can also update his/her details.

HELLO MANU BASVARAJU Dashboard Log out

**Edit Profile**

NAME: Manu Basvaraju DESIGNATION: Head professor

EMAIL ADDRESS: mb@gmail.com

MOBILE NUMBER: 9999999999

PROGRAM: B.tech SEMESTER: V

PASSWORD: 123456

Change the profile image

Choose File No file chosen

Update Profile Reset

## ● View Feedback's

In this section the faculty can see all the feedback of student's given to him. We will not show him/her the name of the students. So it will be completely anonymously

HELLO MR.BASVARAJ TALWAR Dashboard Log out

**Feedback from student's**

S.NO	QUEST1	QUEST2	QUEST3	QUEST4	QUEST5	QUEST6	QUEST7	QUEST8	QUEST9	QUEST10	QUEST11	QUEST12	QUEST13	QUEST14
1	5	5	5	5	5	5	5	5	5	5	5	5	Best course	None
2	5	5	5	5	5	5	5	5	5	5	5	5	Good	None
3	5	4	5	5	5	5	5	5	5	5	5	5	Good	None
Average	5	4	5	5	5	5	5	5	5	5	5	5	---	---



### 3.2.3 Admin Module [HOD]

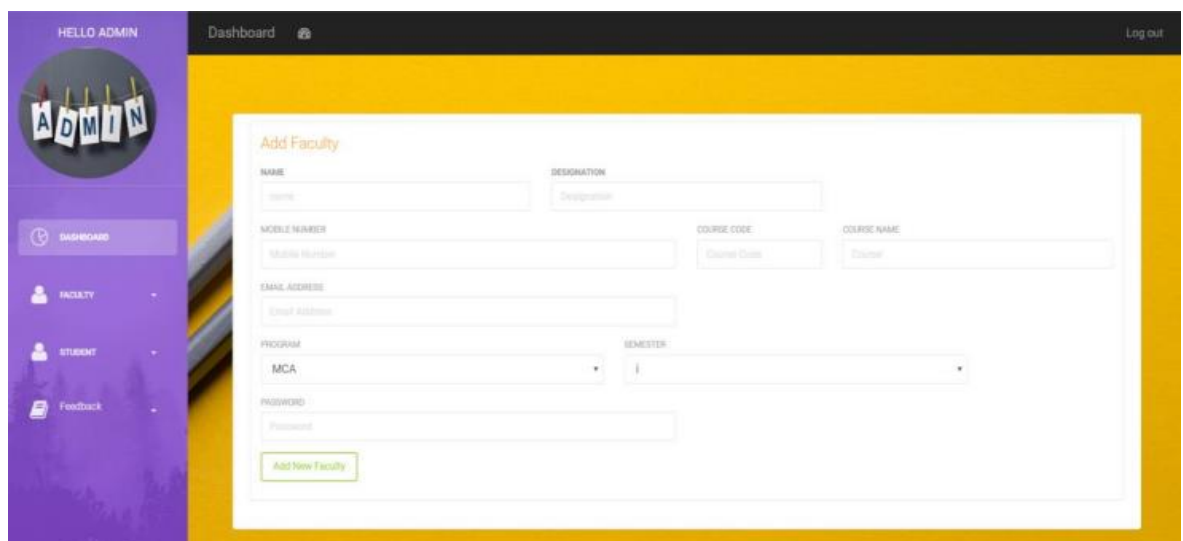
- **Login Page**

We will have only one admin in our system. Admin can login into the system by entering the unique email-id and password provided to him.



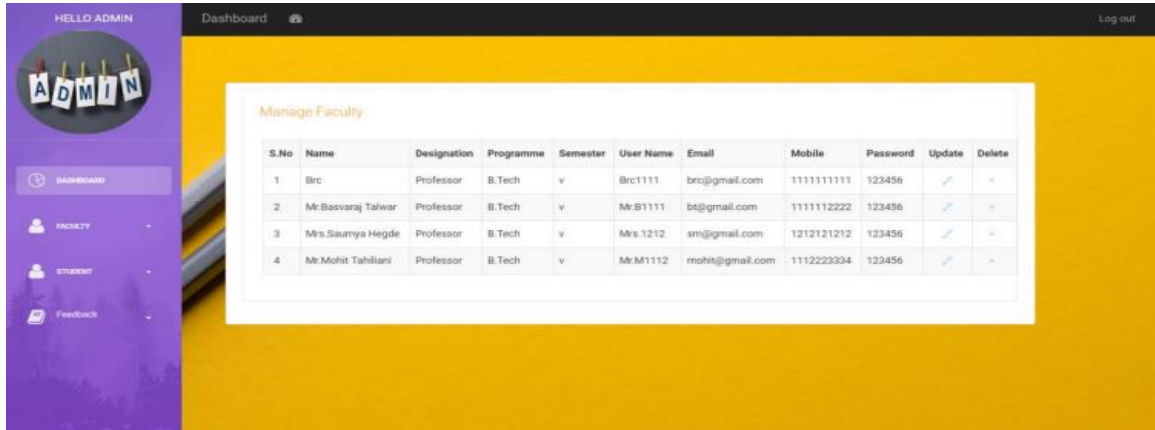
- **Add Faculty**

Admin can add new faculty in the system by entering his/her email-id, course-code, course-name.



- Manage Faculty [Update/Remove Faculty]

Admin can update the details for faculty's in the system and can also remove faculty's from the system



HELLO ADMIN

Dashboard

Log out

ADMIN

DASHBOARD

FACULTY

STUDENT

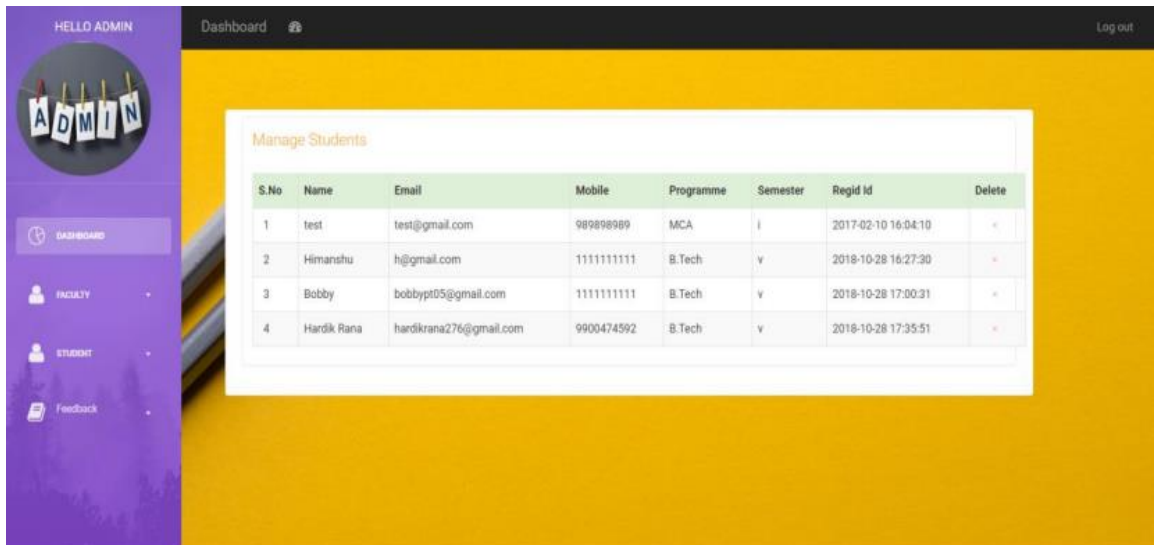
Feedback

Manage Faculty

S.No	Name	Designation	Programme	Semester	User Name	Email	Mobile	Password	Update	Delete
1	Brc	Professor	B.Tech	v	Brc1111	brc@gmail.com	1111111111	123456		
2	Mr.Basvaraj Talwar	Professor	B.Tech	v	Mr-B1111	bt@gmail.com	1111112222	123456		
3	Mrs.Saumya Hegde	Professor	B.Tech	v	Mrs.1212	sm@gmail.com	1212121212	123456		
4	Mr.Mohit Tahlilani	Professor	B.Tech	v	Mr.M1112	mohit@gmail.com	1112223334	123456		

- Manage Students [View/Update Students]

Admin can view all the student's registered in the system and can also remove students from the system.



HELLO ADMIN

Dashboard

Log out

ADMIN

DASHBOARD

FACULTY

STUDENT

Feedback

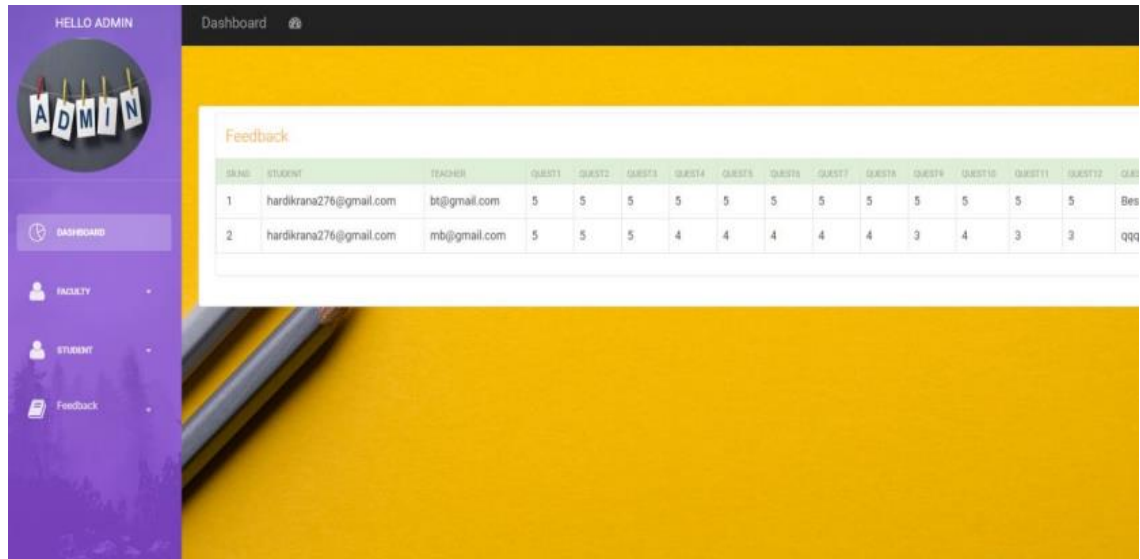
Manage Students

S.No	Name	Email	Mobile	Programme	Semester	Regid Id	Delete
1	test	test@gmail.com	989898989	MCA	i	2017-02-10 16:04:10	
2	Himanshu	h@gmail.com	1111111111	B.Tech	v	2018-10-28 16:27:30	
3	Bobby	bobbypt05@gmail.com	1111111111	B.Tech	v	2018-10-28 17:00:31	
4	Hardik Rana	hardikrana276@gmail.com	9900474592	B.Tech	v	2018-10-28 17:35:51	



- View All Feedbacks

Admin can also view all the feedbacks given to all the faculty's.

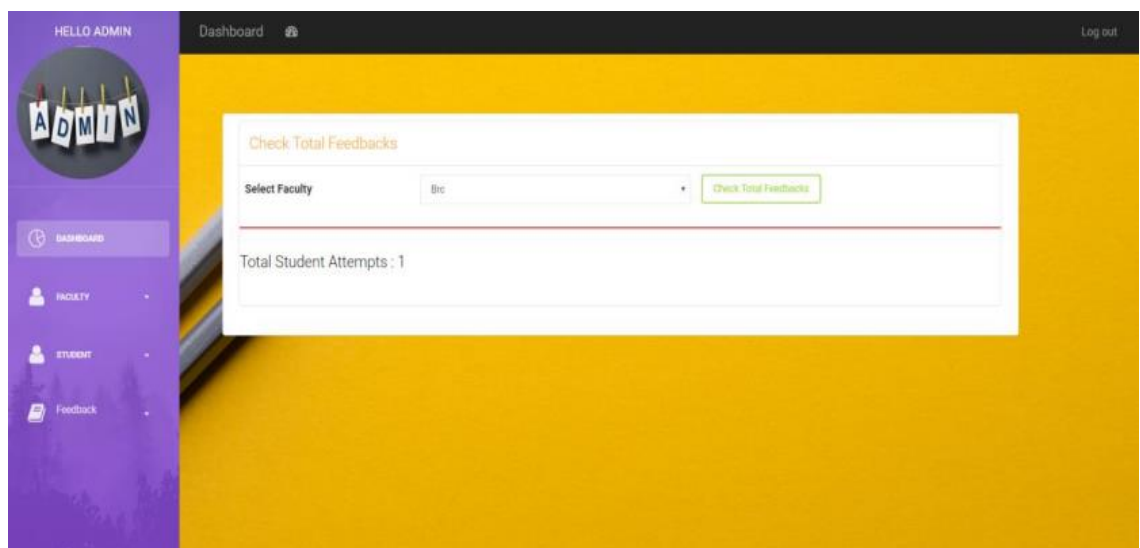


Feedback

SID	STUDENT	TEACHER	QUEST1	QUEST2	QUEST3	QUEST4	QUEST5	QUEST6	QUEST7	QUEST8	QUEST9	QUEST10	QUEST11	QUEST12	QUEST13
1	hardikrana276@gmail.com	bt@gmail.com	5	5	5	5	5	5	5	5	5	5	5	5	5
2	hardikrana276@gmail.com	mb@gmail.com	5	5	5	4	4	4	4	4	3	4	3	3	qqq

- View Total Feedback Per Faculty

Admin can also check how many total feedback each particular faculty have.



Check Total Feedbacks

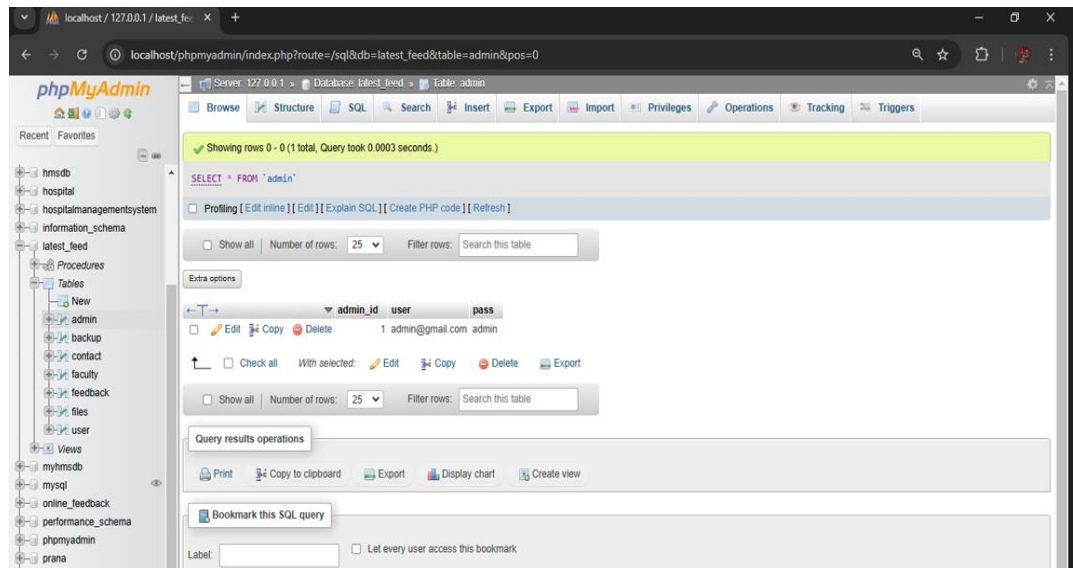
Select Faculty:

Total Student Attempts : 1

### 3.3 Backend Design

- **Admin (HOD & Principal) Table**

Stores admin users with roles, login credentials, and permissions.



- **Backup Table**

A Backup Table (or "Soft Delete Table") in a feedback system functions like a "recycle bin," storing data on students and teachers whose records have been marked for deletion. Instead of permanently removing this data from the database, the backup table acts as a temporary storage area, providing a secure and accessible location for potentially recoverable records. This approach helps prevent accidental or unauthorized deletion and allows administrators to restore information when needed.

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

SELECT \* FROM `backup`

Extra options

id	user_alias	Name	designation	programme	semester	email	password	mobile	date	status	prg	course_code
11	sanj9015	sanjeev kuma	aaaa	B.tec	i	sanjeevtech2@gmail.com	dfdd	901550189	2016-07-13 14:40:35	1	NULL	
17		Dr. R Tamizamuthu	Assistant Professor	MCA	i	rt@gmail.com	12345	7878787878	0000-00-00 00:00:00	0	NULL	18cs2011

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view, Bookmark this SQL query

- **Faculty Table**

Stores informations about the faculty members including username, password, contact numbers, designation, time table and many more. These all information are stored in a database.

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

SELECT \* FROM `faculty`

Extra options

	id	user_alias	Name	designation	programme	semester	email	password	mobile	date	status	prg	course_code
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	sanj9015	sanjeev kumar	Developer	B.tech	ii	sanjeevtech2@gmail.com	2dd9e1	9015501897	2016-07-13 14:37:35	0	NULL	
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12		Dr.Asathy K.Cheran	Assistant Professor	B.Tech	vi	aswathy@gmail.com	12	9898989898	2024-04-30 18:09:43	1	NULL	21CS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13		Dr. Svaraj Paul	Assistant Professor	B.Tech	ii	sp@gmail.com	123	8115391129	0000-00-00 00:00:00	0	NULL	21MA
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15		JANE OLIVE	Assistant Professor	B.Tech	v	jo@gmail.com	12345	6767676767	0000-00-00 00:00:00	0	NULL	21CS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16		Dr.Manjula	Assistant Professor	MCA	i	dm@gmail.com	12345	8989898989	0000-00-00 00:00:00	0	NULL	21cs

Check all With selected: Edit Copy Delete Export

- **Student Table**

Stores informations about the students including username, password, contact numbers, course and semester, time table and many more. These all information are stored in a database.

id	name	email	pass	mobile	programme	semester	gender	hobbies	image
10	sanjeev kumar	sanjeevtech2@gmail.com	98d34c1758b15b5a359b69c2b08c5767	9015501897	B.Tech	3rd	m	reading, playing	Jellyfish.jpg
12	ravi	ravi@gmail.com	63dd3e154ca6d948fc380fa576343ba6	9015501897	M.Tech	ii	m	reading	Desert.jpg
13	warda	warda@yahoo.com	827ccb0eea8a706c4c34a16891f84e7b	32457895212	BCA	ii	f	reading	Koala - Copy.jpg
14	test	test@gmail.com	098f6bcd4621d373cade4e832627b4f6	989898989	MCA	i	m	reading, singin	Chrysanthemum
15	PRANAV SRIVASTAVA	ps@gmail.com	d8578edf8458ce06b5bb76a58c5ca4	9898989898	B.Tech	vi	Male		png-transparent-man-office-work-emoji-icon.png
16	PRANAV SRIVASTAVA	ps2956@gmail.com	d8578edf8458ce06b5bb76a58c5ca4	8115391129	B.Tech	iv	Male		images.png
17	Adarsh Vats	av@gmail.com	d8578edf8458ce06b5bb76a58c5ca4	7897899870	B.Tech	viii	Male		images.png
18	suman	su@gmail.com	d8578edf8458ce06b5bb76a58c5ca4	5656565656	BCA	vii	Male		images.png
19	ARPIT VATS	aa@gmail.com	fee2feca780588782596f0c04eaa8f7	7426800210	B.Tech	i	Male		images.png
20	SUBHASH BISHNOI	ss@gmail.com	d8578edf8458ce06b5bb76a58c5ca4	6969696969	M.Tech	iv	Male		images.png

- **Feedback**

Here in this table feedback is stored which is given by the students to their respective subject faculty.

id	student_id	faculty_id	Teacher provided the course outline having weekly content plan w	Course objectives, learning outcomes and grading criteria are cle	Course integrates theoretical course concepts with the real world	Teacher is punctual, arrives on time and leaves on time	Teacher is good at stimulating the interest in the course conten	Teacher is good at explaining the subject matter	Teacher's presentation was clear, loud and easy to understand	Tea- goo innr test met
16	ravi@gmail.com	ravi@gmail.com	5	5	5	5	5	5	5	5
17	sanjeevtech2@gmail.com	ravi@gmail.com	5	3	1	5	5	3	3	3
18	warda@yahoo.com	ravi@gmail.com	5	5	5	2	1	5	5	4
19	as@gmail.com		5	4	5	5	5	5	5	5
21	as@gmail.com	aswathy@gmail.com	1	1	4	5	4	4	4	5

## **4. PROJECT PLANNING**

Here we aimed to design the online web application for giving the feedback about the lecturers, particular subject, etc. by students to teachers. The system is supposed to be used as a subsystem in a large universities, school and colleges. In addition to that we also provide the rating system due to which the student will rate the teacher based on his/her teaching, we also provide the interactive user interface for student and teacher.

### **4.1 Requirements**

#### **4.1.1 Software Requirements**

1. Web Server: The platform is going to be hosted on the web with php as back-end and the server will also be powered by Xampp and Apache server.
2. DBMS: All the data will be stored in structured tables which will be implemented using MySQL, an open source relational database management system.
3. Development: For development phase of our system we decided again on php. Our development platform will be .php and we are planning to use the following tools and languages.
  - Php, MySQL programming language for main development
  - HTML, CSS, JavaScript, Bootstrap and Font Awesome is used to improve the front-end and user experience of the website.
  - Sublime Text and Atom as development tool.

#### 4. Other Development Software:

- Windows 10 operating system
- MS Office and Google docs for reports or any other documents.
- Lucid-Chart, erdplus for diagrams
- Xampp server, XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

#### **4.1.2 Hardware Requirements**

1. Web Server: We need a reliable web server for our system. This machine must be fast and must show high performance in all situations. At least 512MB RAM and Pentium4 2000MHz processor seems to be the minimum requirements for this machine. Any IBM, HP machine can be selected for this purpose.
2. Database Server: Since our system requires a huge amount of data to be stored, we will need an extra machine that will serve as a database. At least 40GB storage capacity is needed for this system. And this machine must also be a high performance machine. An IBM machine like xseries 382 may be a suitable choice for this purpose.

## 4.2 Database Design

The ER diagram representing the model of Online Feedback System. The entity-relationship diagram of Online Feedback system show all the visual instruments of database tables and relationship between Student, Teacher, Feedback and Admin. It used structured data and define relationship between structured data groups of online feedback system functionalities. The Relations are Adds, Manages, Views etc.

The Entities involved in the ER-diagram are

1. Admin
2. Teacher
3. Student
4. Feedback

### 4.2.1 TABLE DESCRIPTION

1. Admin

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE MEANING	ATTRIBUTE VALUE
(1)	Id	Admin's id	Varchar(10)
(2)	Password	Admin's Password	Varchar(30)

## 2. Teacher

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE MEANING	ATTRIBUTE VALUE
(1)	Id	Teacher's id	Varchar(10)
(2)	Password	Teacher's Password	Varchar(30)
(3)	Mobile	Mobile number of teacher	Varchar(11)
(4)	Designation	Teacher's Designation	Varchar(10)
(5)	Sem	Semester the teacher is teaching	Int(10)
(6)	Email	Email id of teacher	Varchar(50)
(7)	Name	Teacher's Name	Varchar(50)

## 3. Feedback

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE MEANING	ATTRIBUTE VALUE
(1)	Id	Feedback's id	Varchar(10)
(2)	T_id	Teacher's id	Varchar(10)
(3)	Stu_id	Student's Id	Varchar(10)
(4)	Type	Feedback Type	Varchar(10)
(5)	Qn1	Response of Question 1	Enum ('1','2','3','4','5')



(6)	Qn2	Response of Question 2	Enum ('1','2','3','4','5')
(7)	Qn3	Response of Question 3	Enum ('1','2','3','4','5')
(8)	Qn4	Response of Question 4	Enum ('1','2','3','4','5')
(9)	Qn5	Response of Question 5	Enum ('1','2','3','4','5')
(10)	Suggestion	Any personal suggestion the student want to give to the teacher.	Varchar(200)

#### 4. Student

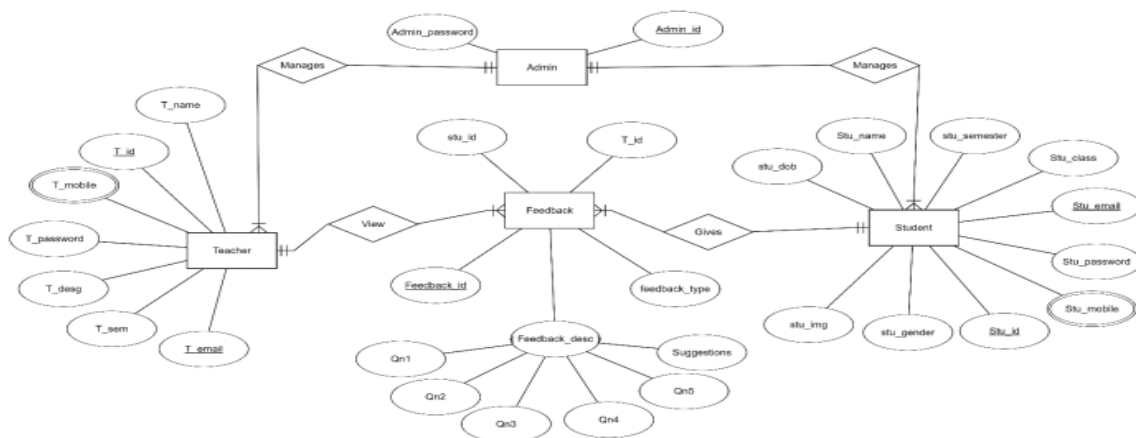
SR. NO.	ATTRIBUTE NAME	ATTRIBUTE MEANING	ATTRIBUTE VALUE
(1)	Id	Student's id	Varchar(10)
(2)	Password	Student's Password	Varchar(30)
(3)	Mobile	Mobile number of Student	Varchar(11)
(4)	Sem	Semester of student	Int(2)
(5)	Class	Branch of the student	Varchar(20)
(6)	Email	Email id of Student	Varchar(50)
(7)	Name	Student's Name	Varchar(50)
(8)	Gender	Student's gender	Enum ('male', 'female', 'other')
(9)	Img	Student's image	Varchar(30)
(10)	Dob	Student's date of birth	time

## 4.2.2 Description of Students Feedback System

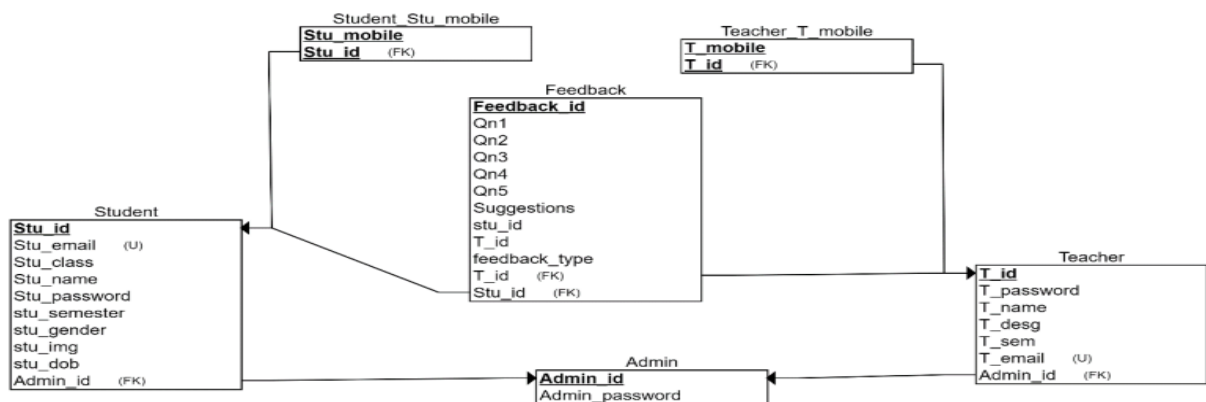
The entity is a concept or object in which the piece of information can be stored.

There are three types of relationship between entities. They are as follows:

- One to One(1-1): This relationship specifies that one instance of an entity is associated with another instance of an entity.
- One to Many(1-N): This relationship specifies that one instance of an entity is associated with zero or many other instances of another entity.
- Many to Many(N-N): This relationship specifies that one instance of an entity is associated with zero or many other instances of another entity.



## 4.2.3 Relational Database design



#### 4.2.4 Constraints in relation Schema

Give all the types of constraints with explanations that you have used for your project. For example:

- Key Constraints

Relation	Primary Key	Foreign Key
Student	Id	
Teacher	Id	
Feedback	Feedback_Id	T_id Stu_id
Admin	Id	

- Cardinality Ratio

Relation	1:1	N:1	M:N
Manages (Admin , Teacher)	No	Yes	No
Manages (Admin, Student)	No	Yes	No
Gives (Student, Feedback)	No	Yes	No
View ( Teacher, Feedback)	No	Yes	No

#### **4.2.5 EXPLANATION**

In Online Feedback System there are 4 entities Admin, Student, Feedback and Teacher.

##### **1. Admin**

The Admin can add teacher and can view the feedback given by the student. In admin's relation schema admin's id is primary key. The relationship between Student entity and teacher entity is 'manages'. Admin has 1:N cardinality ratio with both Teacher and Student. Admin can add many teacher and views feedback of many student.

##### **2. Teacher**

Teacher can view the feedback given by the student. In teacher relation schema teacher's id is the primary key. The relationship between teacher entity and feedback entity is 'Views'. Teacher has 1:N cardinality ratio with Feedback entity. Teacher can view feedback of many student but, only feedback given to that particular teacher. One teacher cannot see the feedback of other teacher.

##### **3. Student**

Student can give feedback to the teacher. In student relation schema student's id is the primary key. The relationship between student and feedback is 'gives'. Student has 1:N cardinality ratio with feedback entity. One student can give feedback to many teacher. A student can only give feedback to teacher belonging to same branch and semester.

#### **4. Feedback**

Student's feedback is stored in the feedback entity. In feedback entity feedback\_id is the primary key, teacher\_id and student\_id is the primary key. Feedback entity gives the information about what is the feedback given to a particular student by a particular student. It relates student entity to teacher entity

## 5. FRONTEND DEVELOPMENT

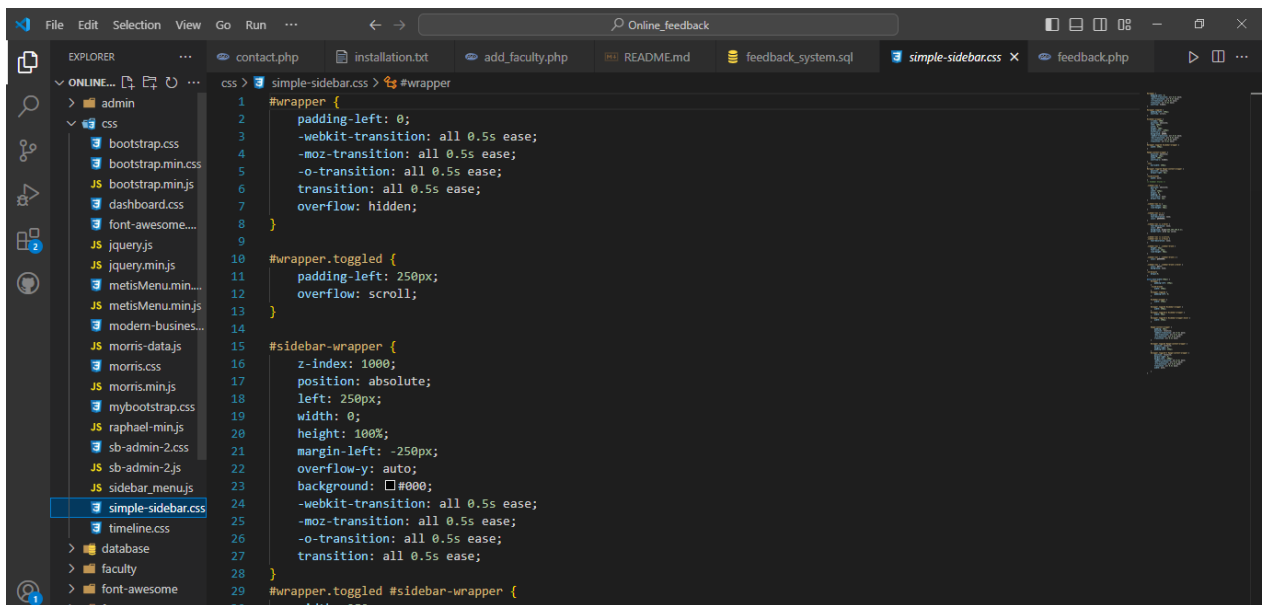
### 5.1 HTML/CSS Structure

```
1  /*  
2  * Base structure  
3  */  
4  
5  /* Move down content because we have a fixed navbar that is 50px tall */  
6  body {  
7    padding-top: 50px;  
8  }  
9  
10  
11  /*  
12  * Global add-ons  
13  */  
14  
15  .sub-header {  
16    padding-bottom: 10px;  
17    border-bottom: 1px solid #eee;  
18  }  
19  
20  /*  
21  * Top navigation  
22  * Hide default border to remove 1px line.  
23  */  
24  .navbar-fixed-top {  
25    border: 0;  
26  }  
27  
28  /*  
29  * Sidebar
```

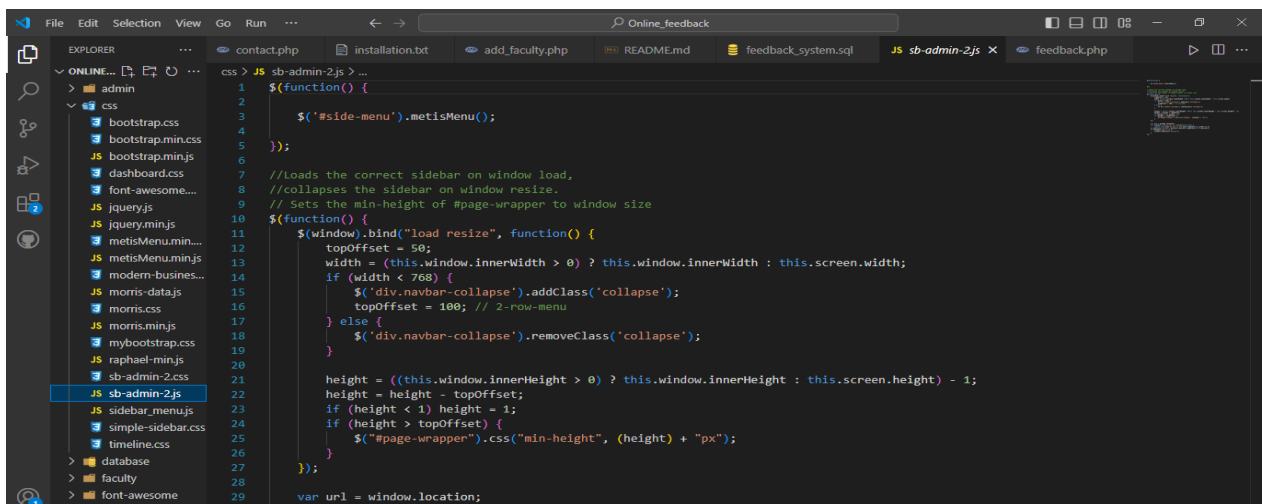
```
1  /*!  
2  * Bootstrap v3.3.6 (http://getbootstrap.com)  
3  * Copyright 2011-2015 Twitter, Inc.  
4  * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)  
5  */  
6  /*! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */  
7  html {  
8    font-family: sans-serif;  
9    -webkit-text-size-adjust: 100%;  
10   -ms-text-size-adjust: 100%;  
11  }  
12  body {  
13    margin: 0;  
14  }  
15  article,  
16  aside,  
17  details,  
18  figcaption,  
19  figure,  
20  footer,  
21  header,  
22  hgroup,  
23  main,  
24  menu,  
25  nav,  
26  section,  
27  summary {  
28    display: block;  
29  }
```

```
1  css > modern-business.css > header.carousel
2  /*
3   * Start Bootstrap - Modern Business HTML Template (http://startbootstrap.com)
4   * Code licensed under the Apache License v2.0.
5   * For details, see http://www.apache.org/licenses/LICENSE-2.0.
6   */
7  /* Global Styles */
8
9  html,
10 body {
11     height: 100%;
12 }
13
14 body {
15     padding-top: 50px; /* Required padding for .navbar-fixed-top. Remove if using .navbar-static-top. Change if height of n
16 }
17
18 .img-portfolio {
19     margin-bottom: 30px;
20 }
21
22 .img-hover: hover {
23     opacity: 0.8;
24 }
25
26 /* Home Page Carousel */
27
28 header.carousel {
29     height: 50%;
```

```
1  css > sb-admin-2.css > ...
2  /*
3   * Start Bootstrap - SB Admin 2 Bootstrap Admin Theme (http://startbootstrap.com)
4   * Code licensed under the Apache License v2.0.
5   * For details, see http://www.apache.org/licenses/LICENSE-2.0.
6   */
7  body {
8     background-color: #f8f8f8;
9  }
10
11 #wrapper {
12     width: 100%;
13 }
14
15 #page-wrapper {
16     padding: 0 15px;
17     min-height: 568px;
18     background-color: #fff;
19 }
20
21 @media(min-width:768px) {
22     #page-wrapper {
23         position: inherit;
24         margin: 0 0 0 250px;
25         padding: 0 30px;
26         border-left: 1px solid #e7e7e7;
27     }
28 }
29
```



## 5.2 JavaScript and Frontend Frameworks (React JS)

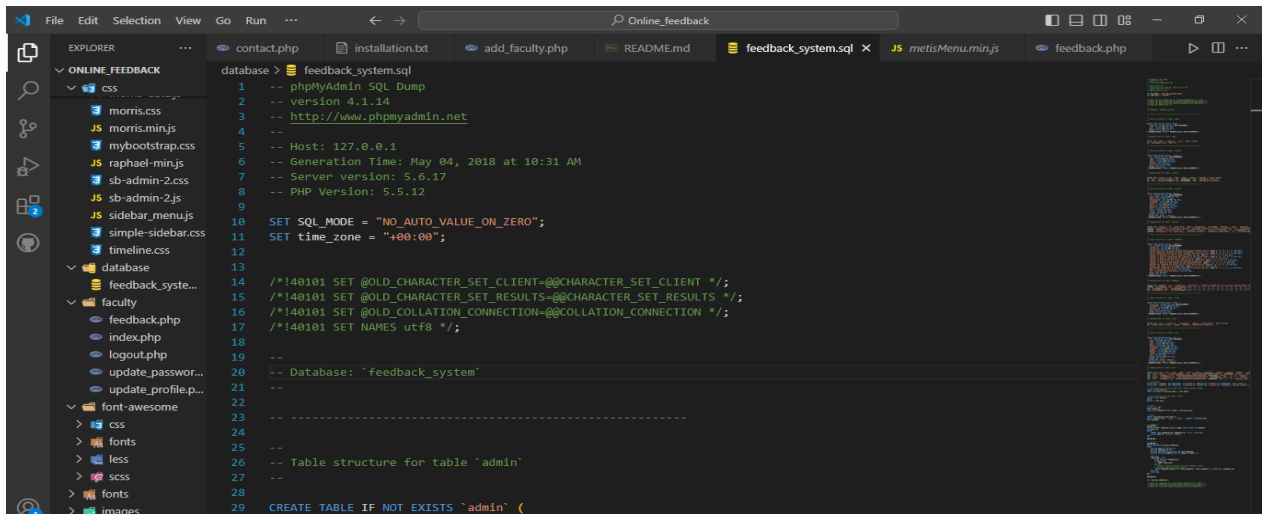




33

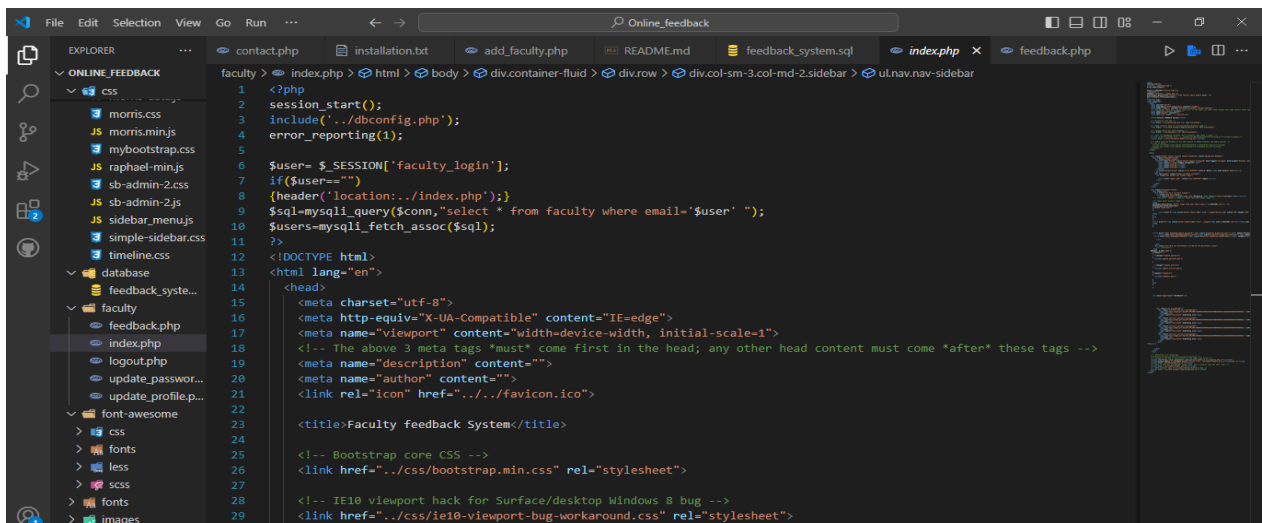
## 6. BACKEND DEVELOPMENT

### 6.1 API Design, Authentication, and Database Connectivity



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders for CSS, JS, database, and fonts. The code editor displays the contents of the feedback\_system.sql file, which is a MySQL database dump. The dump includes comments about the host, generation time, server version, and PHP version. It also includes SQL statements to set the SQL mode, time zone, and character set, and to create the 'admin' table.

```
1 -- phpMyAdmin SQL Dump
2 -- version 4.1.14
3 -- http://www.phpmyadmin.net
4 --
5 -- Host: 127.0.0.1
6 -- Generation Time: May 04, 2018 at 10:31 AM
7 -- Server version: 5.6.17
8 -- PHP Version: 5.5.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8 */;
18
19 --
20 -- Database: `feedback_system`
21 --
22
23 --
24 --
25 --
26 -- Table structure for table `admin`
27 --
28
29 CREATE TABLE IF NOT EXISTS `admin` (
```



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders for CSS, JS, database, and fonts. The code editor displays the contents of the index.php file, which is a PHP script. The script starts with a session start, includes a database configuration file, and sets error reporting. It then checks if the user is logged in. If not, it redirects to the login page. If logged in, it queries the database for the user's email and fetches the user's details. The script then outputs the HTML header and body, including the Bootstrap core CSS and the user's details.

```
1 <?php
2 session_start();
3 include('../dbconfig.php');
4 error_reporting(1);
5
6 $user= $_SESSION['faculty_login'];
7 if($user=="")
8 {header('location:../index.php');}
9 $sql=mysqli_query($conn,"select * from faculty where email='$user' ");
10 $users=mysqli_fetch_assoc($sql);
11 }
12 <!DOCTYPE html>
13 <html lang="en">
14 <head>
15 <meta charset="utf-8">
16 <meta http-equiv="X-UA-Compatible" content="IE=edge">
17 <meta name="viewport" content="width=device-width, initial-scale=1">
18 <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
19 <meta name="description" content="">
20 <meta name="author" content="">
21 <link rel="icon" href=".."../favicon.ico">
22
23 <title>Faculty feedback System</title>
24
25 <!-- Bootstrap core CSS -->
26 <link href=".."../css/bootstrap.min.css" rel="stylesheet">
27
28 <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
29 <link href=".."../css/ie10-viewport-bug-workaround.css" rel="stylesheet">
```

This screenshot shows the VS Code editor with the 'update\_password.php' file open. The Explorer sidebar on the left displays the project structure, including folders for 'css', 'database', 'faculty', 'font-awesome', 'fonts', 'images', and 'user'. The main editor area shows the PHP code for the password update functionality. The code includes session handling, password validation, and a MySQL query to update the faculty password in the 'faculty' table. The file explorer on the right is empty.

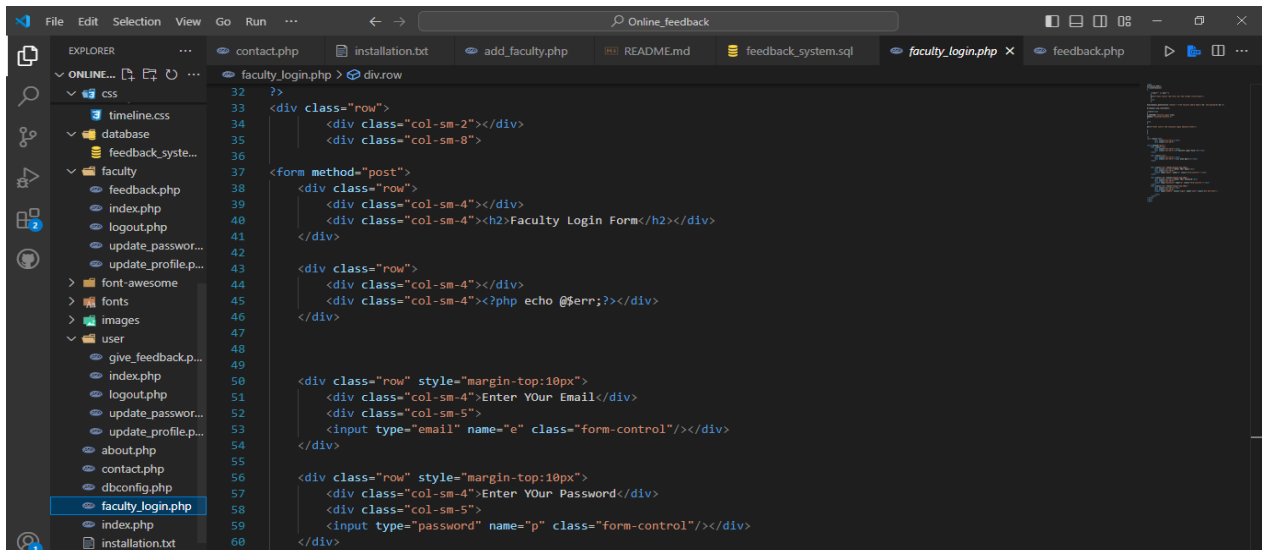
```
1 <?php
2 echo $_SESSION['faculty_login'];
3 extract($_POST);
4 if(isset($save))
5 {
6     if($np=="" || $cp=="" || $op=="")
7     {
8         $err="<font color='red'>fill all the fields first</font>";
9     }
10    else
11    {
12
13
14        $sql=mysqli_query($conn,"select * from faculty where password='$op' and email='".$_SESSION['faculty_login']."'");
15        $r=mysqli_num_rows($sql);
16        if($r==true)
17        {
18            if($np==$cp)
19            {
20                $sql=mysqli_query($conn,"update faculty set password='$np' where email='".$_SESSION['faculty_login']."'");
21                $err="<font color='blue'>Password updated </font>";
22            }
23            else
24            {
25                $err="<font color='red'>New password not matched with Confirm Password </font>";
26            }
27        }
28    }
29 }
```

This screenshot shows the VS Code editor with the 'give\_feedback.php' file open. The Explorer sidebar on the left displays the project structure. The main editor area shows the HTML and PHP code for the feedback form. The form includes a table with two columns for 'Teacher is punctual, arrives on time and leaves on time' and 'Teacher is good at stimulating the interest in the course content'. Each row contains five radio buttons for rating from 1 to 5. The file explorer on the right shows the rendered HTML output of the form.

```
26 <form method="post">
27 <fieldset>
28 <table>
29 <tr>
30 <td><b>2-Class Teaching</b></td>
31 <td><b>4:</b></td>
32 <td><b>Teacher is punctual, arrives on time and leaves on time:</td>
33 <td><input type="radio" name="quest4" value="5" required> 5
34 <input type="radio" name="quest4" value="4"> 4
35 <input type="radio" name="quest4" value="3"> 3
36 <input type="radio" name="quest4" value="2"> 2
37 <input type="radio" name="quest4" value="1"> 1
38 </td>
39 </tr>
40 <tr>
41 <td><b>5:</b></td>
42 <td><b>Teacher is good at stimulating the interest in the course content:</td>
43 <td><input type="radio" name="quest5" value="5" required> 5
44 <input type="radio" name="quest5" value="4"> 4
45 <input type="radio" name="quest5" value="3"> 3
46 <input type="radio" name="quest5" value="2"> 2
47 <input type="radio" name="quest5" value="1"> 1
48 </td>
49 </tr>
50 <tr>
51 <td><b>6:</b></td>
52 <td><b>Teacher is good at explaining the subject matter:</td>
53 <td><input type="radio" name="quest6" value="5" required> 5
54 <input type="radio" name="quest6" value="4"> 4
```

This screenshot shows the VS Code editor with the 'dbconfig.php' file open. The Explorer sidebar on the left displays the project structure. The main editor area shows the PHP code for database configuration. The code includes a MySQL connection using 'mysqli\_connect' with the following parameters: 'localhost', 'root', an empty password, and 'Online\_feedback' as the database name. The file explorer on the right is empty.

```
1 <?php
2 $conn=mysqli_connect("localhost","root","", "Online_feedback")or die(mysqli_error());
3 ?>
```



## 6.2 Database Management – Spring boot Concepts and CRUD Operations

### 1. Created Entity Class

```
import javax.persistence.*;
```

```
@Entity
```

```
@Table(name = "students")
```

```
public class Student {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long id;
```

```
    @Column(name = "name")
```

```
    private String name;
```

```
    @Column(name = "email")
```

```
    private String email;
```

```
// Getters and setters  
}
```

## 2. Created a Repository Interface

```
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface StudentRepository extends JpaRepository<Student, Long> {  
    // Custom query methods can be added here  
}
```

## 3. Creating Service Layer

```
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;
```

```
import java.util.List;  
import java.util.Optional;
```

```
@Service
```

```
public class StudentService {
```

```
    @Autowired
```

```
    private StudentRepository studentRepository;
```

```
    public List<Student> getAllStudents() {  
        return studentRepository.findAll();  
    }
```

```
    public Optional<Student> getStudentById(Long id) {
```

```

        return studentRepository.findById(id);
    }

    public Student createStudent(Student student) {
        return studentRepository.save(student);
    }

    public Student updateStudent(Long id, Student studentDetails) {
        Student student = studentRepository.findById(id).orElseThrow();
        student.setName(studentDetails.getName());
        student.setEmail(studentDetails.getEmail());
        return studentRepository.save(student);
    }

    public void deleteStudent(Long id) {
        studentRepository.deleteById(id);
    }
}

```

#### 4. At last Creating controller

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/api/students")

```

```

public class StudentController {

    @Autowired
    private StudentService studentService;

    @GetMapping
    public List<Student> getAllStudents() {
        return studentService.getAllStudents();
    }

    @GetMapping("/{id}")
    public Optional<Student> getStudentById(@PathVariable Long id) {
        return studentService.getStudentById(id);
    }

    @PostMapping
    public Student createStudent(@RequestBody Student student) {
        return studentService.createStudent(student);
    }

    @PutMapping("/{id}")
    public Student updateStudent(@PathVariable Long id, @RequestBody Student
student) {
        return studentService.updateStudent(id, student);
    }

    @DeleteMapping("/{id}")
    public void deleteStudent(@PathVariable Long id) {

```

```
        studentService.deleteStudent(id);  
    }  
}
```



## **7. TESTING AND DEPLOYMENT**

### **7.1 Testing**

To ensure that the Student Feedback System functions correctly, various levels of testing were conducted, including unit testing, integration testing, system testing, and user acceptance testing. Testing focused on verifying the proper functionality of core features, such as user authentication, feedback submission, and data storage in MySQL.

#### **1. Unit Testing:**

- Individual PHP functions were tested to verify that each function performed its intended task, such as user authentication, data validation, and feedback calculations.
- For instance, tests checked that login and registration functions returned the correct response based on valid or invalid user credentials.
- Form validation scripts in JavaScript were also tested to ensure that users couldn't submit empty or improperly formatted feedback forms.

#### **2. Integration Testing:**

- Integration testing focused on interactions between the PHP back-end and the MySQL database.
- Queries were tested to confirm that feedback submitted by students was correctly inserted into the database and retrieved accurately when displayed on the admin panel.
- PHP sessions were also tested to ensure user-specific data (such as viewing their submitted feedback) persisted correctly.

### 3. System Testing:

- Comprehensive end-to-end tests were conducted to simulate real user workflows, from login and feedback submission to viewing results as an admin.
- Functional testing covered each core system component, including student registration, login, submitting feedback, and generating feedback reports.
- System testing validated the overall usability and performance of the application when run on the localhost environment.

### 4. User Acceptance Testing (UAT):

- The feedback system was shared with a few sample users (students and faculty) who tested its usability and ease of navigation.
- Feedback was collected to improve the user interface and to add clarifying tooltips on certain actions like submitting feedback.
- User acceptance testing confirmed that the system was intuitive and functional, meeting the core requirements for end-users.

### 5. Bug Tracking and Resolution:

- Testing revealed common issues, such as incorrect error messages or occasional display issues in the feedback form.
- A simple bug-tracking document was maintained to record and address any issues. All identified bugs were resolved before the final deployment.

## 6. Testing Results:

- All core functionalities passed the tests successfully, with essential features performing as expected in the localhost environment.
- Testing confirmed the stability and readiness of the application for local deployment and use.

## 7.2 Deployment

Since the Student Feedback System is hosted on a local server, the deployment process was straightforward. The application was deployed on XAMPP (a local Apache server with MySQL and PHP support) to enable testing and demonstration.

### 1. Deployment Environment:

- The system was deployed on XAMPP on a localhost environment, configured with Apache and MySQL for PHP development.
- The local environment setup ensured that all dependencies were managed within the XAMPP package, allowing for easy testing and use on a single machine.

### 2. Deployment Methodology:

- PHP and HTML files were placed in the htdocs folder of the XAMPP installation.
- Database setup involved importing the initial database schema and data into MySQL through phpMyAdmin, a web-based tool included in XAMPP for managing MySQL databases.
- Configurations, including database connection details, were set in a configuration file (e.g., config.php) for easy management.

### 3. Steps in Deployment Process:

- Database Setup: A new database called `student_feedback` was created in MySQL, and the SQL schema file was imported to set up tables and initial data.
- File Placement: All PHP, HTML, CSS, and JavaScript files were copied into the `htdocs/student_feedback` folder, which made the application accessible at `http://localhost/student_feedback`.
- Configuration: The `config.php` file was updated with the MySQL database credentials and other environment-specific configurations.

### 4. Verification and Testing:

- After deployment, the application was tested directly on the localhost to verify that all features were functional in the new environment.
- Basic smoke testing was performed to confirm that pages loaded correctly, database connections were stable, and user sessions persisted as expected.

### 5. Post-Deployment Monitoring:

- Logs were enabled in PHP (using the `error_log` configuration in `php.ini`) to track any run-time errors and capture issues during testing.
- The system was monitored for any error messages or unexpected behavior in the PHP logs during testing and user acceptance demonstrations.
- A regular backup schedule for the `student_feedback` database was also set up to ensure data safety during testing and any demonstrations.

## 6. Rollback Strategy:

- In case of any critical issues, a rollback strategy involved restoring the previous version of the htdocs files and the MySQL database from the latest backup.
- Backups of both the application files and the database were created before any significant updates or modifications to the code.

## **8. RESULTS AND CONCLUSION**

### **6.1 Conclusion**

The end result of the project is a successful implementation of this online feedback web-portal. A proposed system is used to make feedback process in school/colleges through online only. The whole application was built using PHP and bootstrap. The data was stored using the open source MySQL platform and a lot of constraints were kept in mind while creating the structure of the tables. Security measures were taken into consideration, many of the security measures provided by PHP were implemented into the project without fail. One such mechanism is the session variable. The above developed application satisfies the requirement specification. Enhancement to the project can easily be made without changing the current design and programming structure.

### **6.2 Future Scope**

The future of an online student feedback system has a lot of exciting possibilities that can make it even better for everyone involved. Features like using artificial intelligence could help analyze written feedback to find common themes and suggest improvements for teachers. Adding real-time dashboards and customizable reports would make it easier for teachers and school leaders to see and understand the feedback they receive. Integrating the system with existing learning platforms would allow students to access it easily with one login, while a mobile-friendly version would let them give feedback anytime, anywhere. Improving privacy by keeping student identities anonymous would encourage more honest responses. Using fun

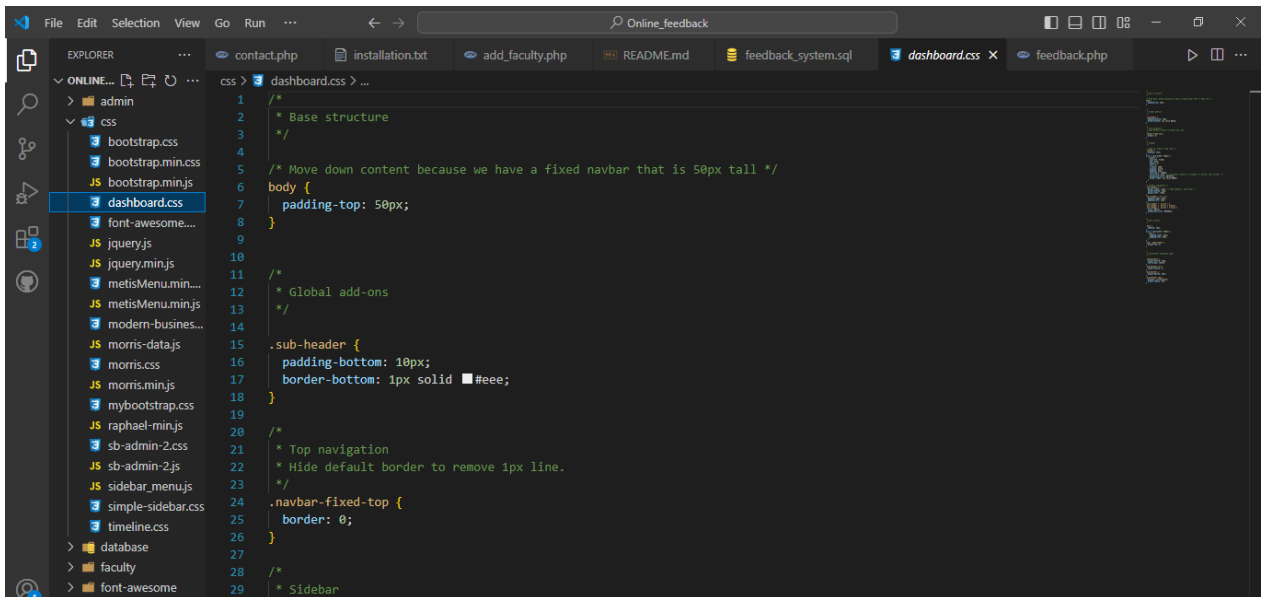
elements like badges or rewards could motivate students to participate more. Automated alerts for low scores and reminders to submit feedback would help ensure timely responses. Finally, using data to predict potential problems and tracking the success of teacher training programs would allow schools to make proactive changes. Overall, these improvements would create a more effective feedback system that helps enhance teaching quality and increase student satisfaction.

## REFERENCES

1. [www.researchgate.net](http://www.researchgate.net)
2. [www.chatgpt.com](http://www.chatgpt.com)
3. [www.youtube.com](http://www.youtube.com)

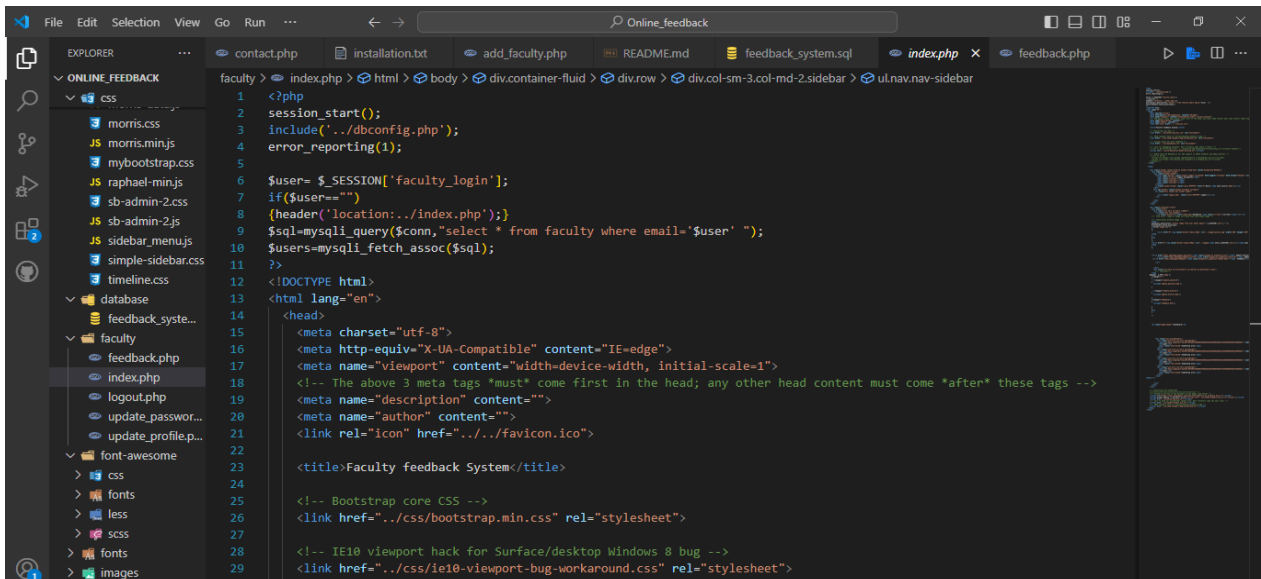


## APPENDIX – CODE



This screenshot shows a code editor with the file explorer on the left displaying a project structure. The 'css' folder is expanded, showing various CSS files, with 'dashboard.css' selected. The main editor area displays the content of 'dashboard.css', which includes comments for base structure, global add-ons, top navigation, and a sidebar.

```
1  /*  
2  * Base structure  
3  */  
4  /* Move down content because we have a fixed navbar that is 50px tall */  
5  body {  
6    padding-top: 50px;  
7  }  
8  
9  
10  
11 /*  
12 * Global add-ons  
13 */  
14  
15 .sub-header {  
16   padding-bottom: 10px;  
17   border-bottom: 1px solid #eee;  
18 }  
19  
20 /*  
21 * Top navigation  
22 * Hide default border to remove 1px line.  
23 */  
24 .navbar-fixed-top {  
25   border: 0;  
26 }  
27  
28 /*  
29 * Sidebar
```



This screenshot shows a code editor with the file explorer on the left displaying a project structure. The 'index.php' file is selected. The main editor area displays the PHP code for 'index.php', which includes session management, database connection, and HTML output for the head and body sections.

```
1  <?php  
2  session_start();  
3  include('../dbconfig.php');  
4  error_reporting(1);  
5  
6  $user= $_SESSION['faculty_login'];  
7  if($user=="")  
8  {header("location:../index.php");}  
9  $sql=mysqli_query($conn,"select * from faculty where email='$user' ");  
10 $users=mysqli_fetch_assoc($sql);  
11 ?>  
12 <!DOCTYPE html>  
13 <html lang="en">  
14 <head>  
15   <meta charset="utf-8">  
16   <meta http-equiv="X-UA-Compatible" content="IE=edge">  
17   <meta name="viewport" content="width=device-width, initial-scale=1">  
18   <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->  
19   <meta name="description" content="">  
20   <meta name="author" content="">  
21   <link rel="icon" href="../../favicon.ico">  
22  
23   <title>Faculty feedback System</title>  
24  
25   <!-- Bootstrap core CSS -->  
26   <link href="../../css/bootstrap.min.css" rel="stylesheet">  
27  
28   <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->  
29   <link href="../../css/ie10-viewport-bug-workaround.css" rel="stylesheet">
```

## APPENDIX – SCREENSHOTS

