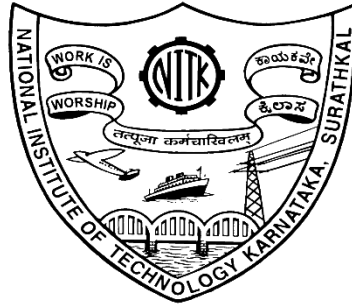


National Institute of Technology Karnataka, Surathkal



Database Management Systems Lab Project

Report On Online Student Feedback System

Submitted By

Himanshu Nagdive (16CO127)

Hardik Rana (16CO138)

To whom it may concern

This is to certify that Hardik Rana and Himanshu Nagdive have worked on the project title 'Online feedback system' ,Under my guidance at the computer science and engineering department, National Institute of technology Karnataka as a part of course project.

Signature
(Dr. M Venkateshan)

DECLARATION

We certify that the report on 'Online student feedback system' which is being submitted as record of our DBMS course project is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or Institution for the award of any degree.

Himanshu Nagdive
16CO127
16co127.himanshu@nitk.edu.in

Hardik Rana
16CO138
16co138.hardik@nitk.edu.in

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.

CONTENTS

1. Introduction
 - 1.1 Purpose
 - 1.2 Objective
2. Requirement specifications
 - 2.1 Hardware specifications
 - 2.2 Software specifications
3. Database Design
 - 3.1 ER Diagram
 - 3.2 Relational Database Design
 - 3.3 Constraints in Relation Schema
4. Project Components
 - 4.1 Front End Design
 - 4.2 Security Measures
5. Results & Discussions
6. Conclusion

List of Tables and Figures

1. Entity Table	06
2. ER Model	10
3. Relational Model	13
4. Front-End Design.....	16

Introduction

1.1 Purpose:

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.

1.2 Objective :

In Existing System the feedback is done by the manual process. In the Existing System students can give the feedback about the lecturers by using paper and pen.

After giving feedback by every student papers are collected by HOD's and calculates the overall grade for each subject and each lecturer. After that those all grade report is viewed by the principal which is given by HOD's. Hence estimating the performance of lecturers and giving counseling can be done.

Requirement Analysis and specifications

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

2.1 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.

2.2 Hardware Requirements

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.

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.

3.1. ER Diagram

Database Design

This ER diagram represents 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

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

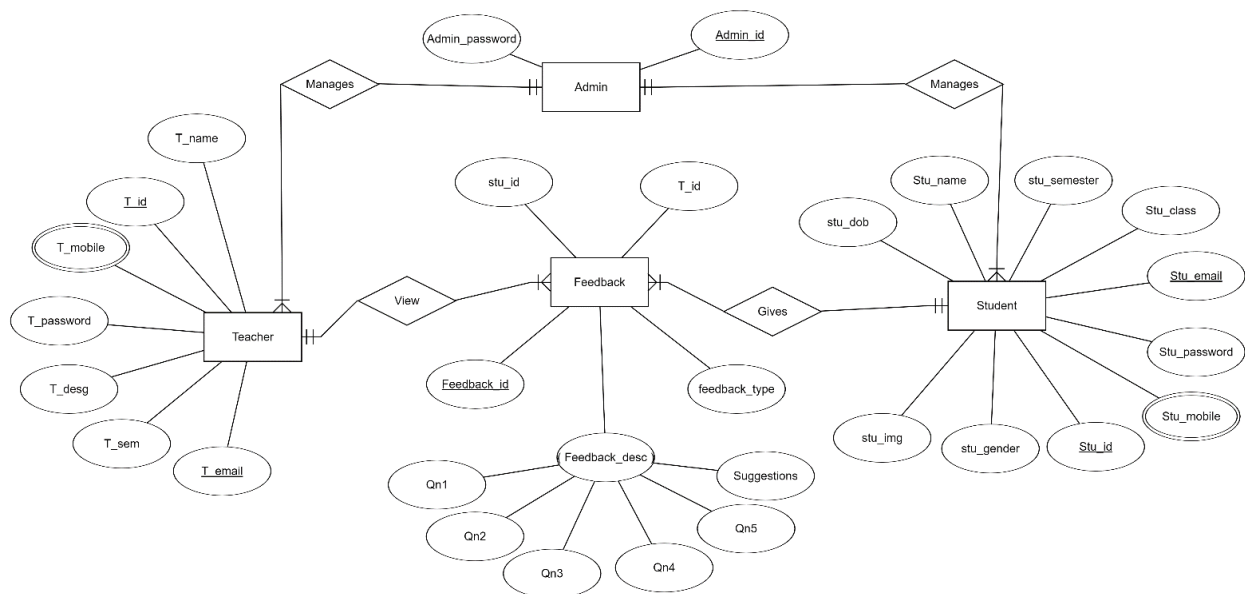
SR. NO.	ATTRIBUTE NAME	ATTRIBUTE MEANING	ATTRIBUTE VALUE
(1)	Id	Student's id	Varchar(10)
(2)	Password	Student's Password	Varchar(30)
(3)	Mobile	Mobile number of Student	Varchar(11)
(4)	Sem	Semester of student	Int(2)
(5)	Class	Branch of the student	Varchar(20)
(6)	Email	Email id of Student	Varchar(50)
(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

Description of Students Feedback System :

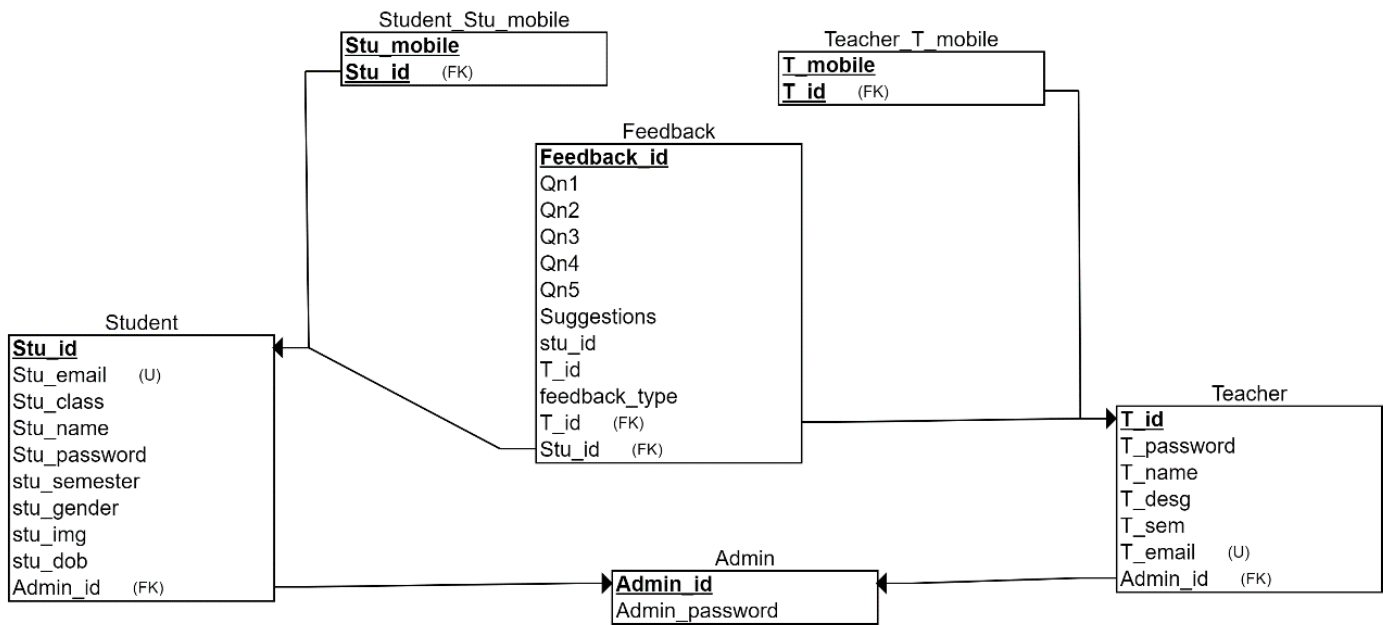
The entity is a concept or object in which the piece of information can be stored.

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.



3.2 Relational Database design



Relational schematic diagram for Online Feedback System

3.3 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

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.

Project Components

4.1 Front End Design:

4.1.1 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..

Welcome to Online Feedback System

Registration Form

Menu

Name

Email Address

Mobile Number

Program

Semster

Gender

Choose File No file chosen

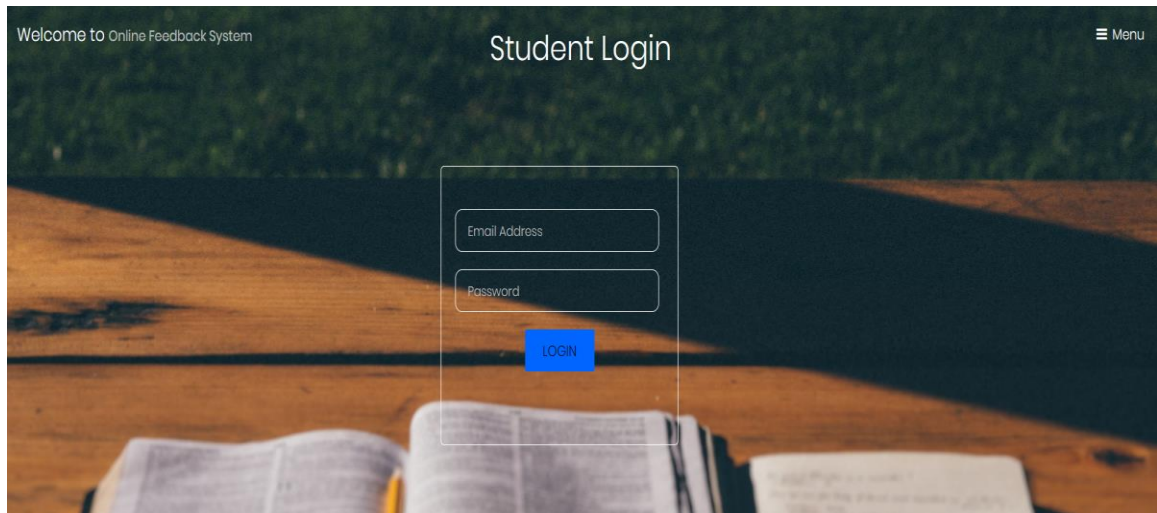
mm/dd/yyyy

Password

Activate Windows
Go to Settings to activate Windows.

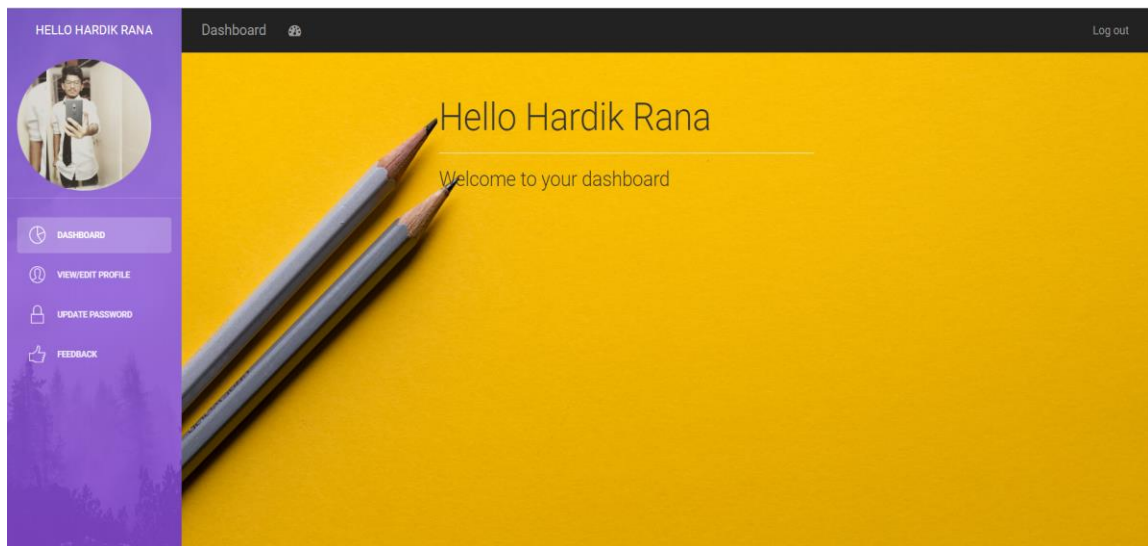
- **Login Page**

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



- **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.

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/EDIT PROFILE', 'UPDATE PASSWORD', and 'FEEDBACK'. The main area displays the 'Edit Profile' form. The form includes fields for 'NAME' (Hardik Rana), 'EMAIL ADDRESS' (hardikrana276@gmail.com), 'MOBILE NUMBER' (9900474592), 'PROGRAM' (B.Tech), 'SEMESTER' (V), 'GENDER' (Male), and 'DATE OF BIRTH' (07/27/1999). There is a 'Choose File' button for the profile image and 'Update My Profile' and 'Reset' buttons at the bottom.

- **Update Password**

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

The screenshot shows the same student dashboard as the previous one, but with the 'Update Password' form displayed. The form has three input fields: 'ENTER YOUR OLD PASSWORD', 'ENTER YOUR NEW PASSWORD', and 'ENTER YOUR CONFIRM PASSWORD'. At the bottom, there are 'Update My Password' and 'Reset' buttons.

- **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.

HELLO HARDIK RANA Dashboard Log out

Please give your answer about the following question by circling the given grade on the scale:

Strongly Agree 5 **Agree 4** **Neutral 3** **Disagree 2** **Strongly Disagree 1**

Give Your Feedback

SELECT FACULTY
Manu Basvaraju

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

Teacher is available and helpful during counseling hours: 5 4 3 2 1

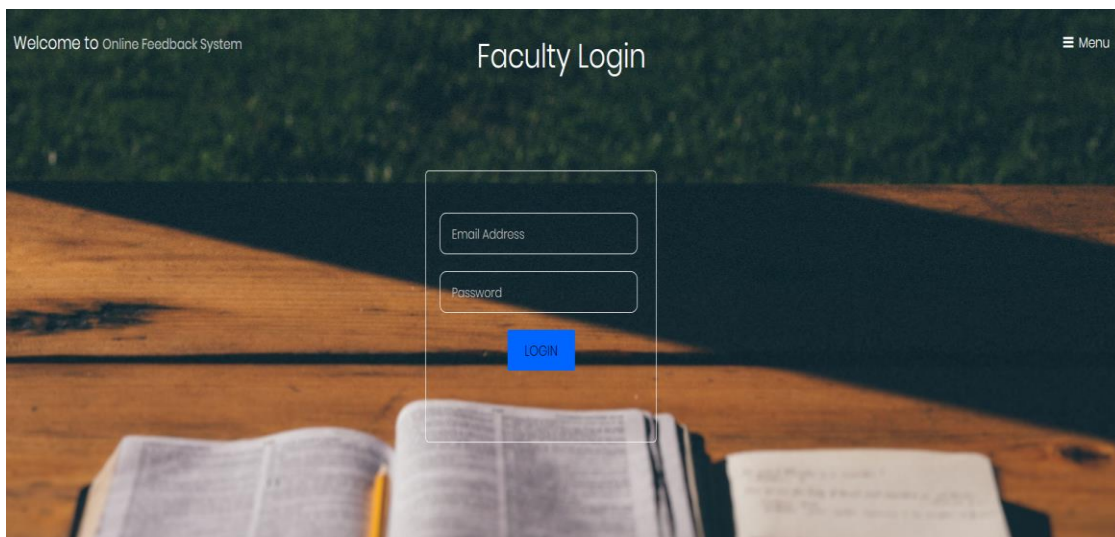
Teacher has completed the whole course as per course outline: 5 4 3 2 1

Activate Windows
Go to Settings to activate Windows.

4.1.2 Faculty Module

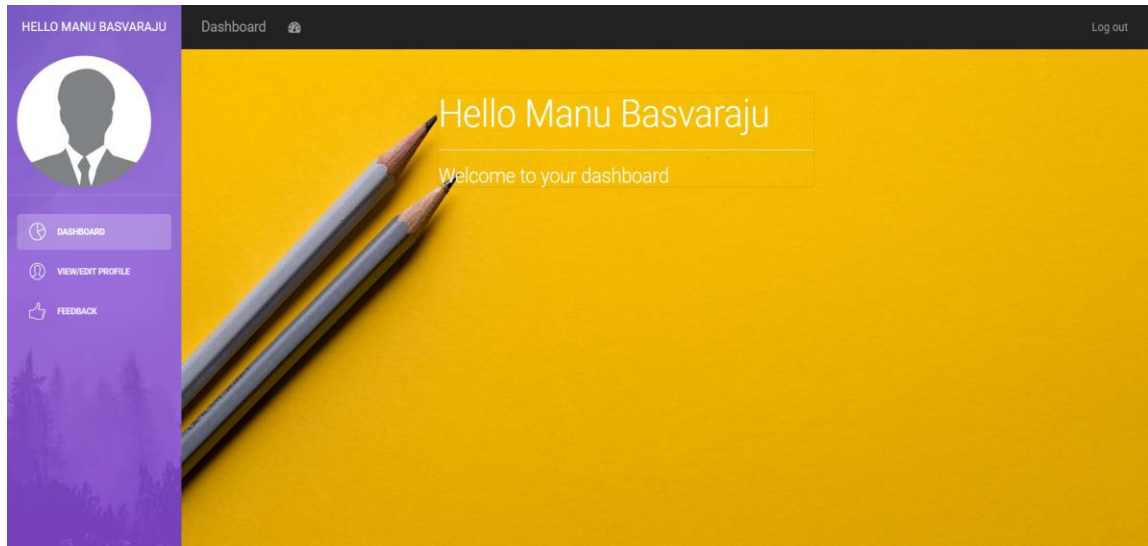
- **Login Page**

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].



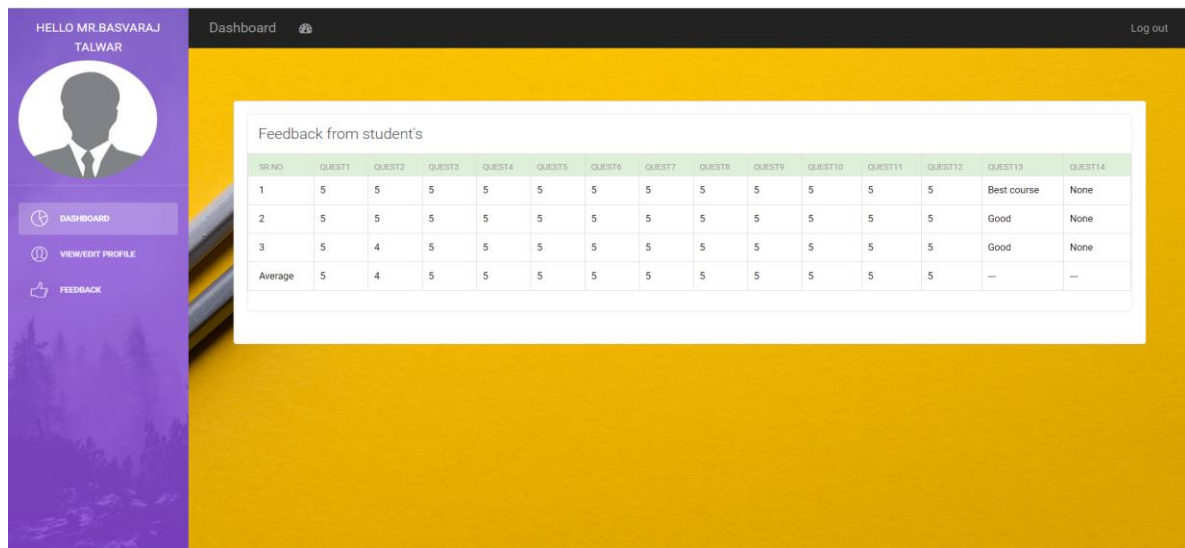
- **View/Edit Profile**

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

A screenshot of the 'Edit Profile' form within the same web application. The form is white and contains several input fields: 'NAME' (Manu Basvaraju), 'DESIGNATION' (Head professor), 'EMAIL ADDRESS' (mb@gmail.com), 'MOBILE NUMBER' (9999999999), 'PROGRAM' (B.tech), 'SEMESTER' (V), and 'PASSWORD' (123456). To the right of the form is a profile image placeholder with a 'Change the profile image' link and a 'Choose File' button. At the bottom right of the form are 'Update Profile' and 'Reset' buttons. The sidebar and navigation bar are identical to the dashboard screenshot.

- **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.



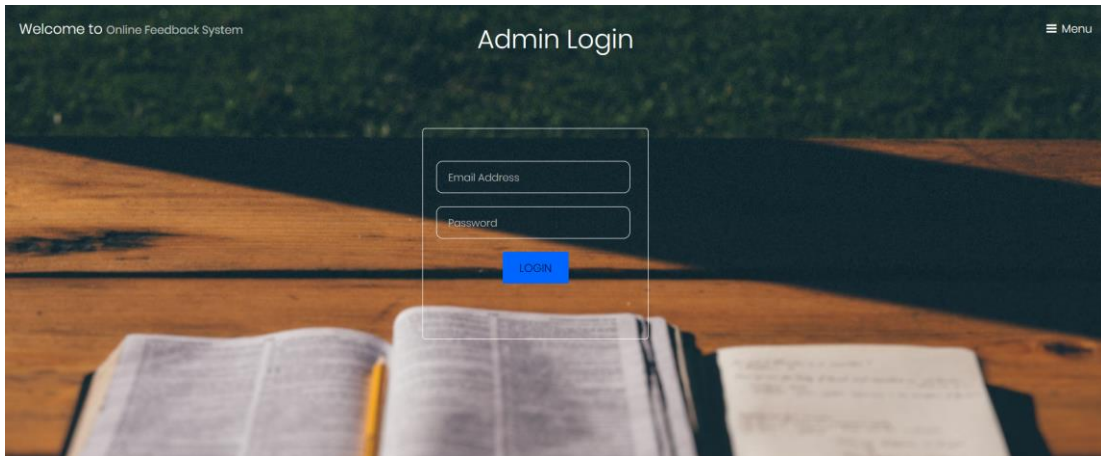
The screenshot shows a web application interface for a faculty member. The top navigation bar is dark grey with the text 'HELLO MR.BASVARAJ TALWAR' on the left, 'Dashboard' in the center, and 'Log out' on the right. A purple sidebar on the left contains a profile picture placeholder and three menu items: 'DASHBOARD' (selected), 'VIEWEDIT PROFILE', and 'FEEDBACK'. The main content area has a yellow background and displays a table titled 'Feedback from student's'. The table has 15 columns: 'SR.NO', 'QUEST1', 'QUEST2', 'QUEST3', 'QUEST4', 'QUEST5', 'QUEST6', 'QUEST7', 'QUEST8', 'QUEST9', 'QUEST10', 'QUEST11', 'QUEST12', 'QUEST13', and 'QUEST14'. It contains three rows of student feedback and an 'Average' row. The feedback data is as follows:

SR.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	---	---

4.1.3 Admin Module [HOD]

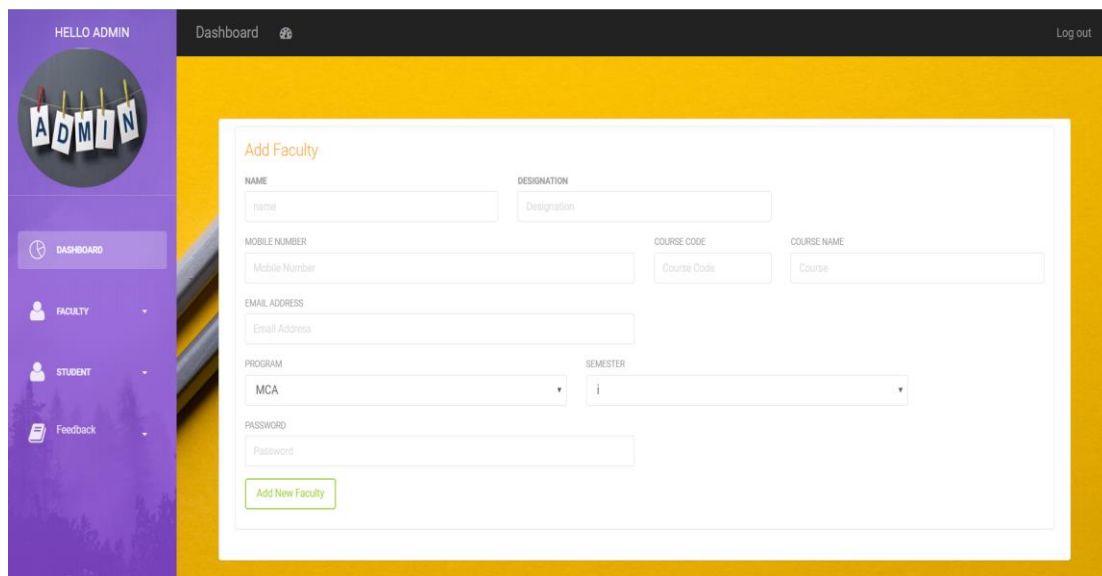
- **Login Page**

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.



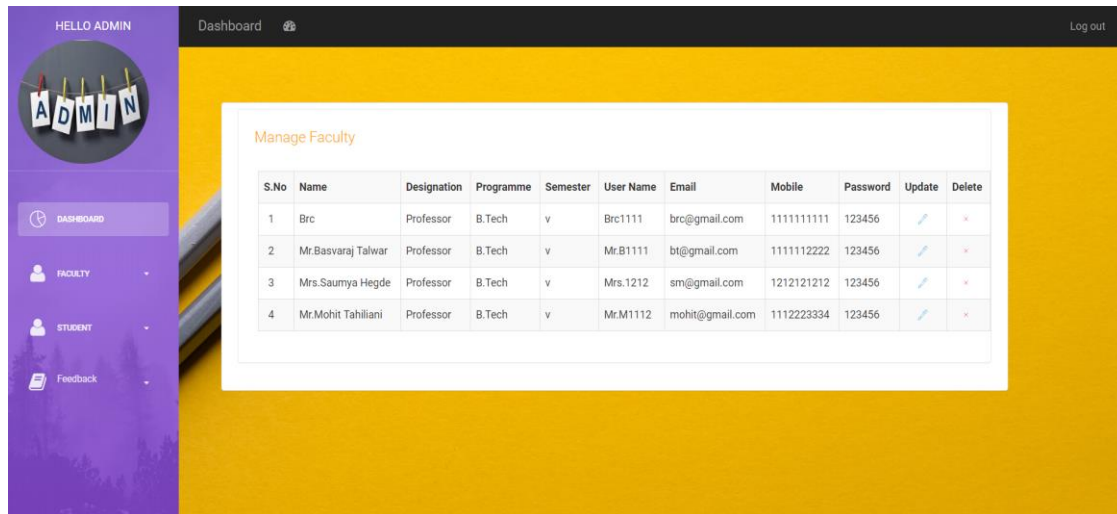
- **Add Faculty**

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



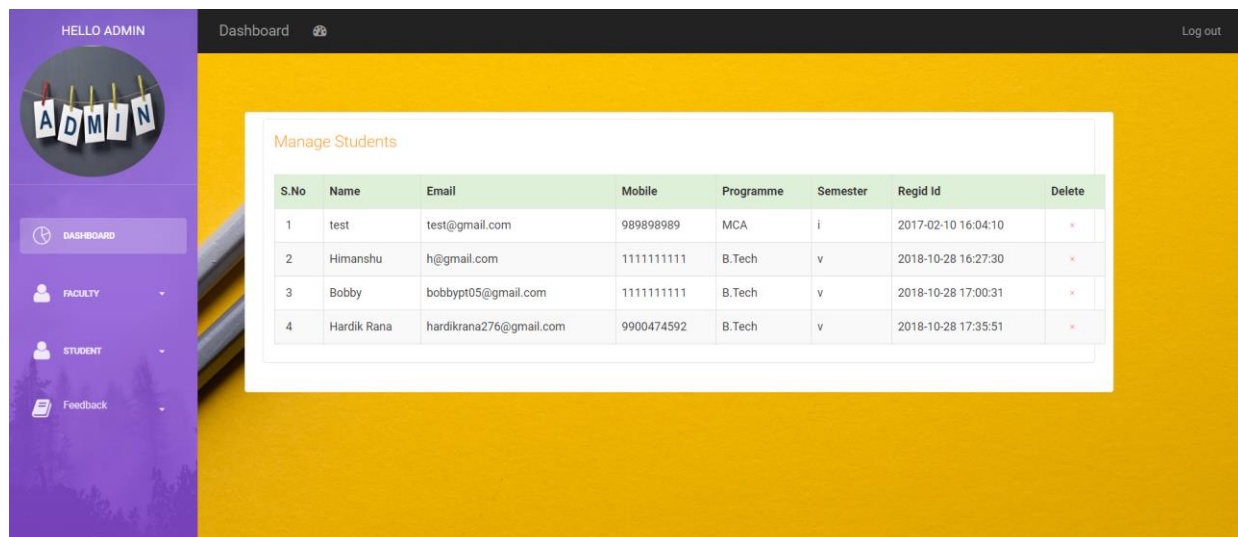
HELLO ADMIN | Dashboard | Log out

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.



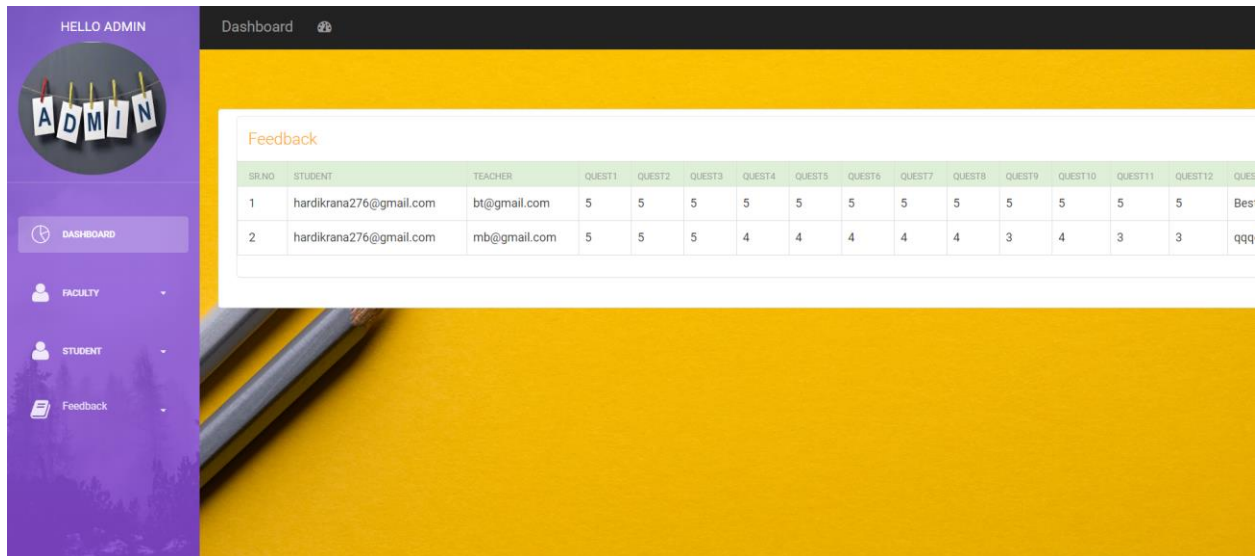
HELLO ADMIN | Dashboard | Log out

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	bobbyp105@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.



HELLO ADMIN

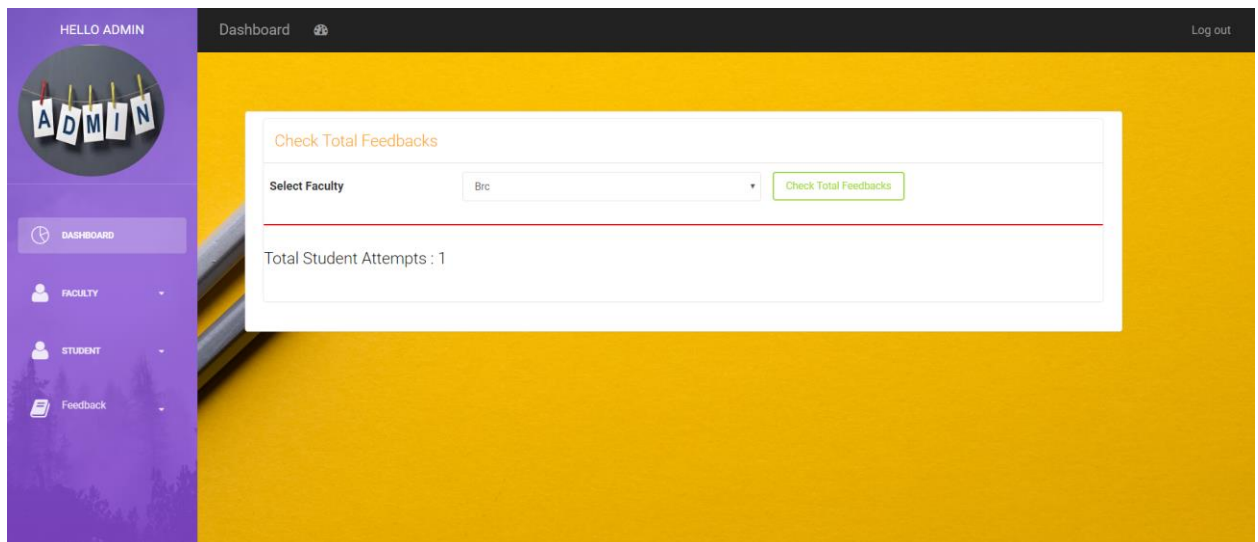
Dashboard

Feedback

SIR.NO	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	Bes
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.



HELLO ADMIN

Dashboard

Log out

Check Total Feedbacks

Select Faculty: Brc

Check Total Feedbacks

Total Student Attempts : 1

4.2 Security Measures:

Below there are a list of possible vulnerability that may occur in our application:

1.SQL INJECTIONS

The SQL injection is a type of code injection, where attackers make full use of the vulnerabilities in the website security measures to send special SQL queries to the database which can modify it and tables within it or delete the whole database.

To prevent it we are doing following:

- We are validating and also varifying each and every data entered by user [student] while registration,while login and while giving feedback.
- We are also validating and varifying each data of faculty.
- We are also securing the confidential information Of user.For example we are encrypting each user's password using md5 in php.
- We are also using `mysqli_real_escape_string()` which will avoid all the special character and we have also used `htmlentities()` to avoid html entities.
- We are avoiding using words such as 'insert', 'update', 'drop', and 'union' from being added to the database, as these all being words can alter tables and databases.
- We are also limiting the permissions granted on the database, because fewer permissions will result in fewer chances of hacking attack.

2.BROKEN AUTHENTICATION & SESSION MANAGEMENT

Broken authentication and session management encompass several security issues, all of them having to do with maintaining the identity of a user. If authentication credentials and session identifiers are not protected at all times an attacker can hijack an active session and assume the identity of a user.

To prevent it we are doing following:

- **Password Strength:** We are always ensuring that user password must be of some minimum length required.
- **Password Change Controls:** We are using a single password change mechanism where users are allowed to change a password, regardless of the

situation. Users should always be required to provide both their old and new password when changing their password.

- **Password Storage:** All the passwords in our system are stored in either hashed form to protect them from exposure, regardless of where they are stored. Hashed form is preferred since it is not reversible. We are also ensuring that passwords should never be hardcoded in any source code.
- **Session ID Protection:** We are doing session ID protection as following:
 1. We start session when student/faculty/admin login to website and store their ID in global session array.
 2. If any user want to access any webpage, and if given webpage should be access by specific user type then according to ID set in session array. If user is allowed to access the page then we show them the page, else we redirect user to their home page.
 3. If user logout the website then we destroy all session variable so if they try to access previous webpage they can't access.

3. INSECURE DIRECT OBJECT REFERENCES

Insecure Direct Object References occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

Insecure Direct Object References allow attackers to bypass authorization and access resources directly by modifying the value of a parameter used to directly point to an object. Such resources can be database entries belonging to other users, files in the system, and more. This is caused by the fact that the application takes user supplied input and uses it to retrieve an object without performing sufficient authorization checks.

To prevent it we are doing following:

- Before giving access of any server resource, we are checking whether user is authorized or not.

- We are not taking the id from any parameter and directly identify the user based on session object over the server. We are creating a session variable over the server and storing user specific information in it[`email-id`] and checking it in each request.
- Accessing each page of the website which requires authenticated user we are making sure that the user is logged in to the system by checking with session variable.

4.CROSS SITE SCRIPTING (XSS)

Cross-site scripting (XSS) targets an application's users by injecting code, usually a client-side script such as JavaScript, into a web application's output. The concept of XSS is to manipulate client-side scripts of a web application to execute in the manner desired by the attacker. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, or redirect the user to malicious sites.

To prevent it we are doing following:

- We are validating every GET and POST value which comes from user and When outputting any of these values, we are also escaping them so they will not be evaluated in an unexpected way.
- PHP provides a few ways to escape output depending on the context. We are using php's filter functions to do this. [PHPs Filter Functions](#) will allow the input data to the php script to be [sanitized](#) or [validated](#) in [many ways](#). They are useful when saving or outputting client input.

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.

