**A Report**

**On**

**Development of**

**Online Feedback System**

Prepared in partial fulfilment of the requirement of MCA 310P: **Minor Project Work**

**By**

**B. Sreekanth (2321256)**

**Here add Student Name (student hall tick no)**



**Department of Computer Applications**

**KMM INSTITUTE OF POST GRADUATE STUDIES**

(Affiliated to Sri Venkateshwara University, Tirupati)

Rami Reddy Palle, TIRUPATI-517 102

**APRIL, 2023**

**A Report**

**On**

**Development of**

**Online Feedback System**

Prepared in partial fulfilment of the requirement of MCA310P: **Minor Project Work**

**By**

**B. Sreekanth (2321256)**

**Student Name (student hall tick no)**

Under the Guidance of Ms. C. YAMINI



**Department of Computer Applications**

**KMM INSTITUTE OF POST GRADUATE STUDIES**

(Affiliated to Sri Venkateshwara University, Tirupati)

Rami Reddi Palle, TIRUPATI-517 102

**APRIL, 2023**

**To whom it may concern**

**This is** to certify that B. Sreekanth have worked on the project title ‘Online feedback system’, Under my guidance at the KMM Institute of PG Studies, RamireddyPalli as a part of Minor project.

Signature

(Dr.K. Venkataramana)

**DECLARATION**

We certify that the report on ‘Online student feedback system’ which is being submitted as record of our MCA 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.

B. Sreekanth

2321256

sreekanthbalireddi@gmail.com

## 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 requiring too gives feedback using one standard feedback form.

In our project, the security is also maintained 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 analysing 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. Purpose
   2. Objective
2. Requirement specifications
   1. Hardware specifications
   2. Software specifications
3. Database Design
   1. ER Diagram
   2. Relational Database Design
   3. Constraints in Relation Schema
4. Project Components
   1. Front End Design
   2. Security Measures
5. Project Source Code
6. Results & Discussions
7. 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. An Online Feedback System is a 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 analysing 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 sent 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 analysing 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 all student papers are collected by HOD’s and calculates the overall grade for each subject and each lecturer. After that all-grade report is viewed by the principal which is given by HOD’s. Hence estimating the performance of lecturers and giving counselling 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.
* Visual Studio Code as development tool.

**4. Other Development Software:**

* Windows 11 operating system
* MS Office and Google docs for reports or any other documents.
* Lucid-Chart, periplus 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 x-series 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.

1. Teacher
2. Student
3. 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) |

1. **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 teachers | 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) |

1. **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) | Qn6 | Response of Question 6 | Enum (‘1’,’2’,’3’,’4’,’5’) |
| (11) | Qn7 | Response of Question 7 | Enum (‘1’,’2’,’3’,’4’,’5’) |
| (12) | Qn8 | Response of Question 8 | Enum (‘1’,’2’,’3’,’4’,’5’) |
| (13) | Suggestion | Any personal suggestion the student want to give to the teacher. | Varchar (200) |
| (14) | Suggestion | Any personal suggestion the student want to give to the teacher. | Varchar (200) |

1. **STUDENT**

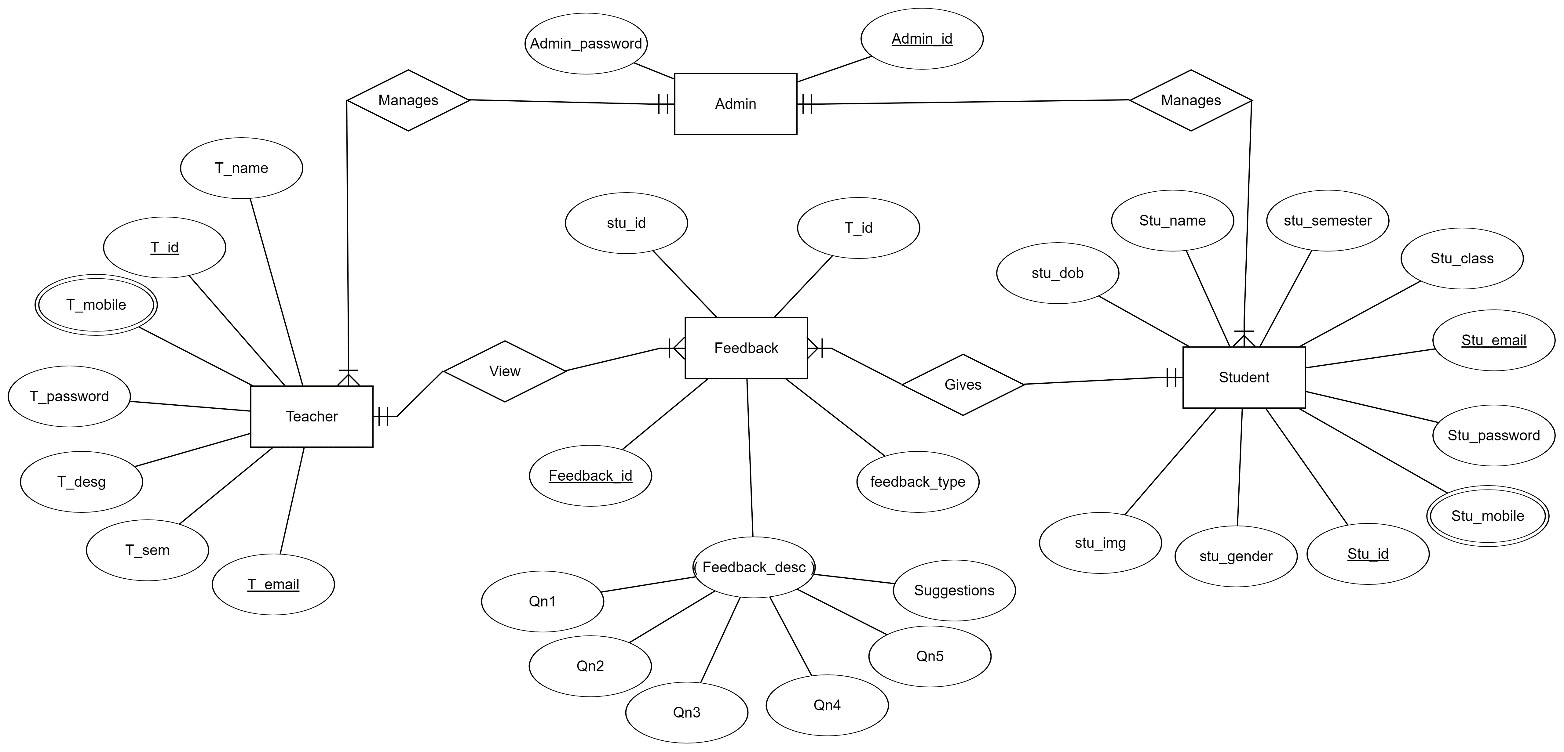
|  |  |  |  |
| --- | --- | --- | --- |
| SR. NO. | ATTRIBUTE NAME | ATTRIBUTE MEANING | ATTRIBUTE VALUE |
| (1) | Id | Student’s id | Varchar (10) |
| (2) | Password | Student’s Password | Varchar (30) |
| (3) | Mobile | Mobile number of Student | Varchar (11) |
| (4) | Sem | Semester of student | Int (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.

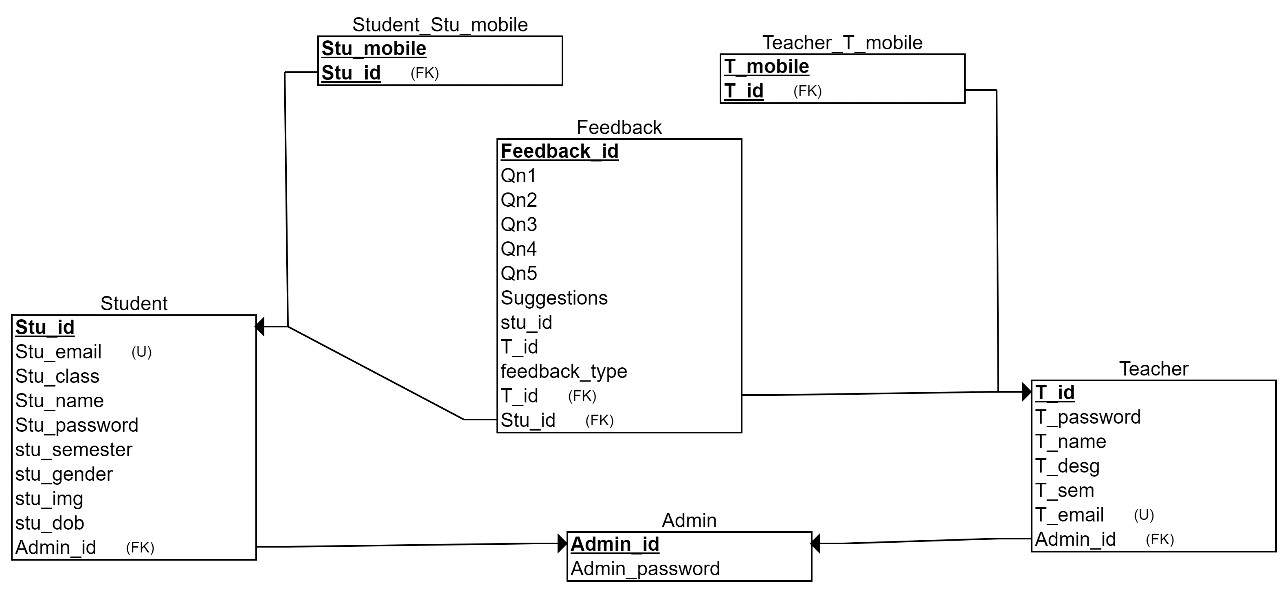
There are three types of relationship between entities. They are as follows:

* **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 | 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 |

**EXPLAINATION**

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 admins 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 teachers and views feedback of many students.

##### (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 had 1: N cardinality ratio with Feedback entity. Teacher can view feedback of many students but, only feedback given to that particular teacher. One teacher cannot see the feedback of another 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 teachers. 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.

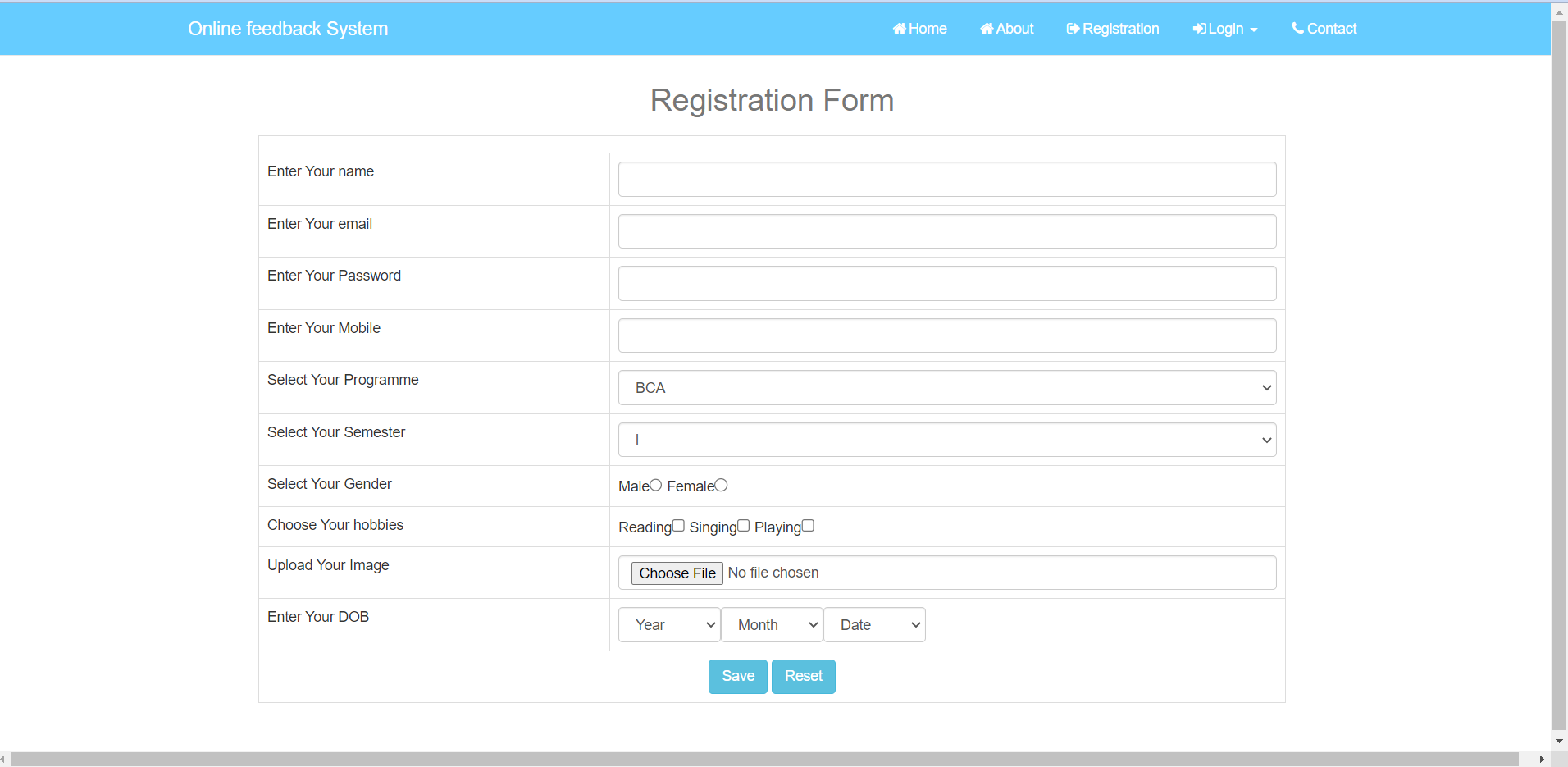
## Project Components

**4.1 Front End Design:**

### 4.1.1 Student Module

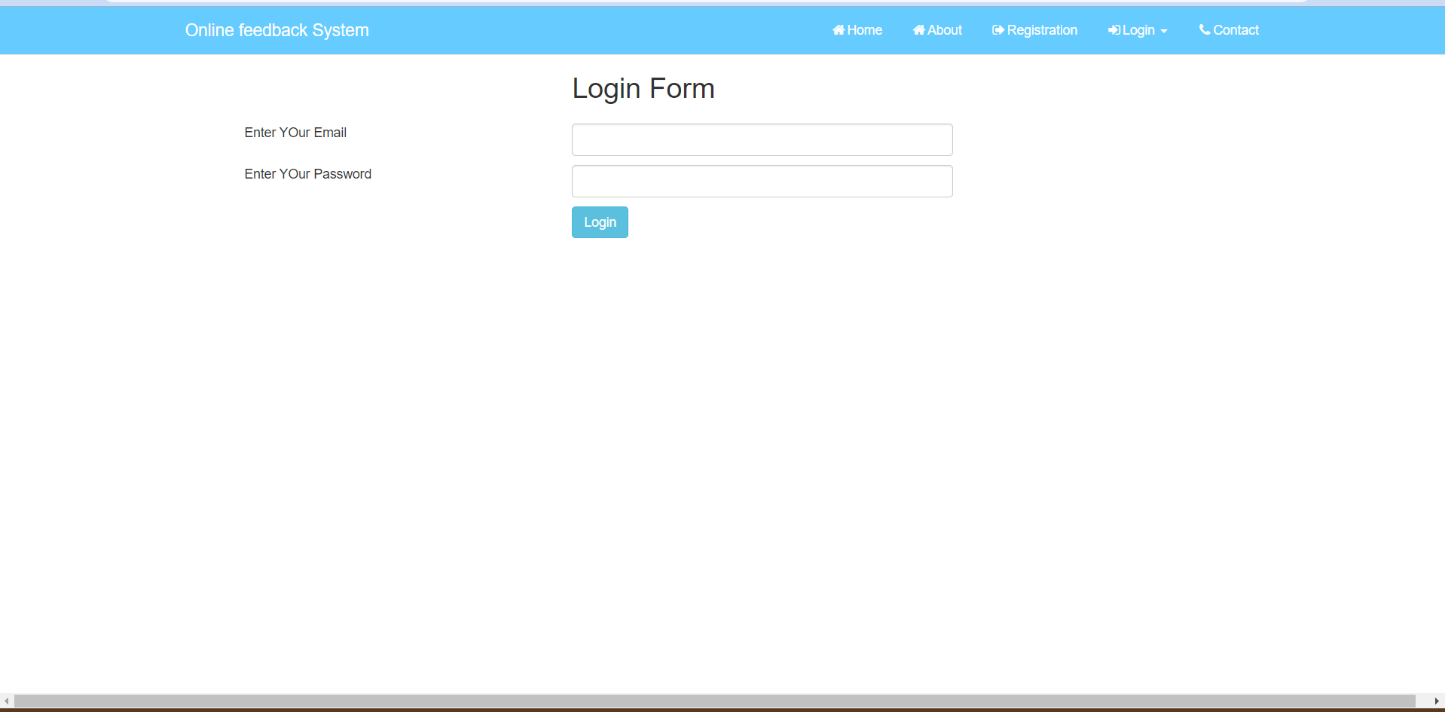
* **Registration Page**

Any new student to this web-portal must first register himself. Without registration he/she cannot 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..

****

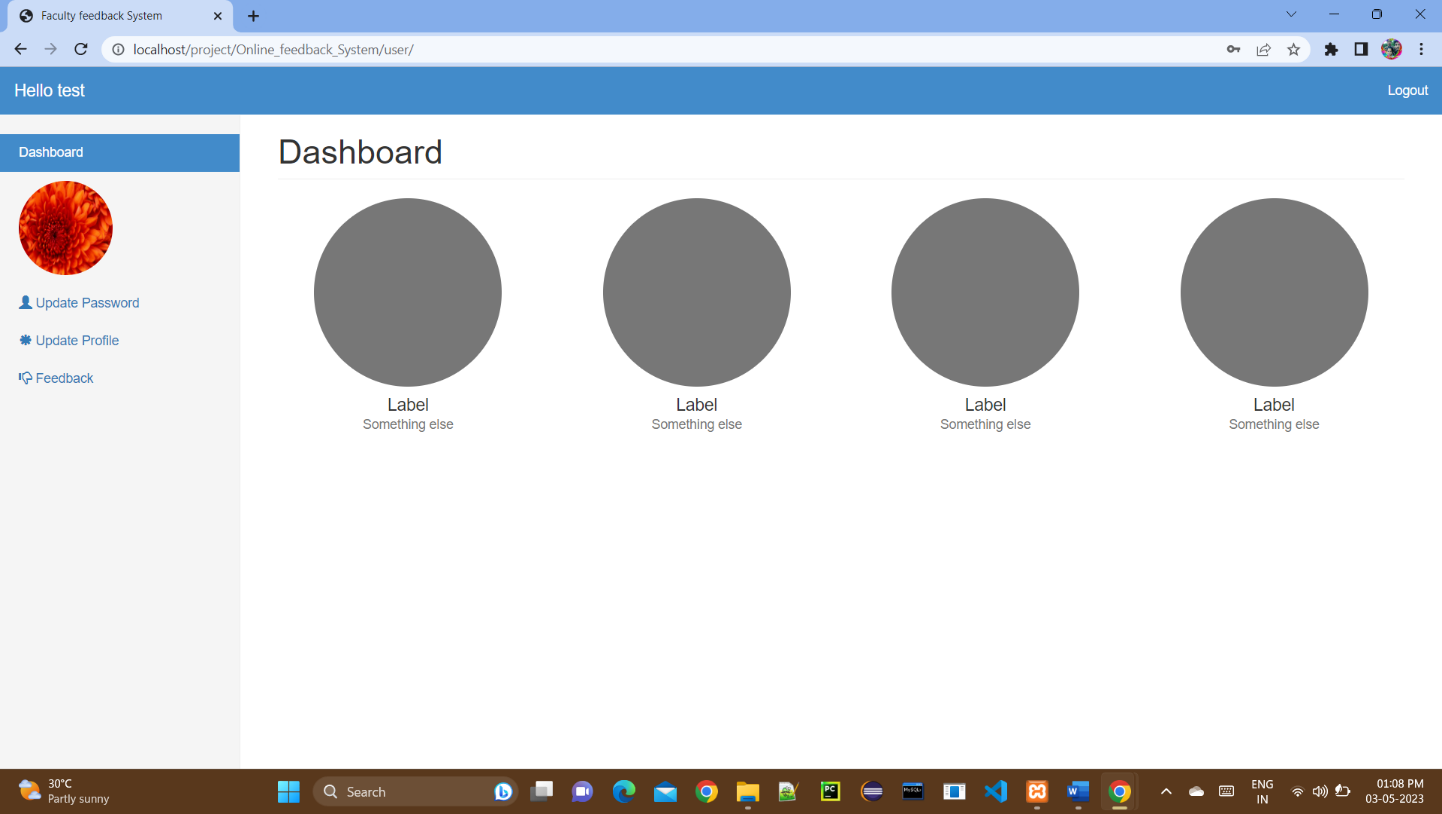
* **Login Page**

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.

****

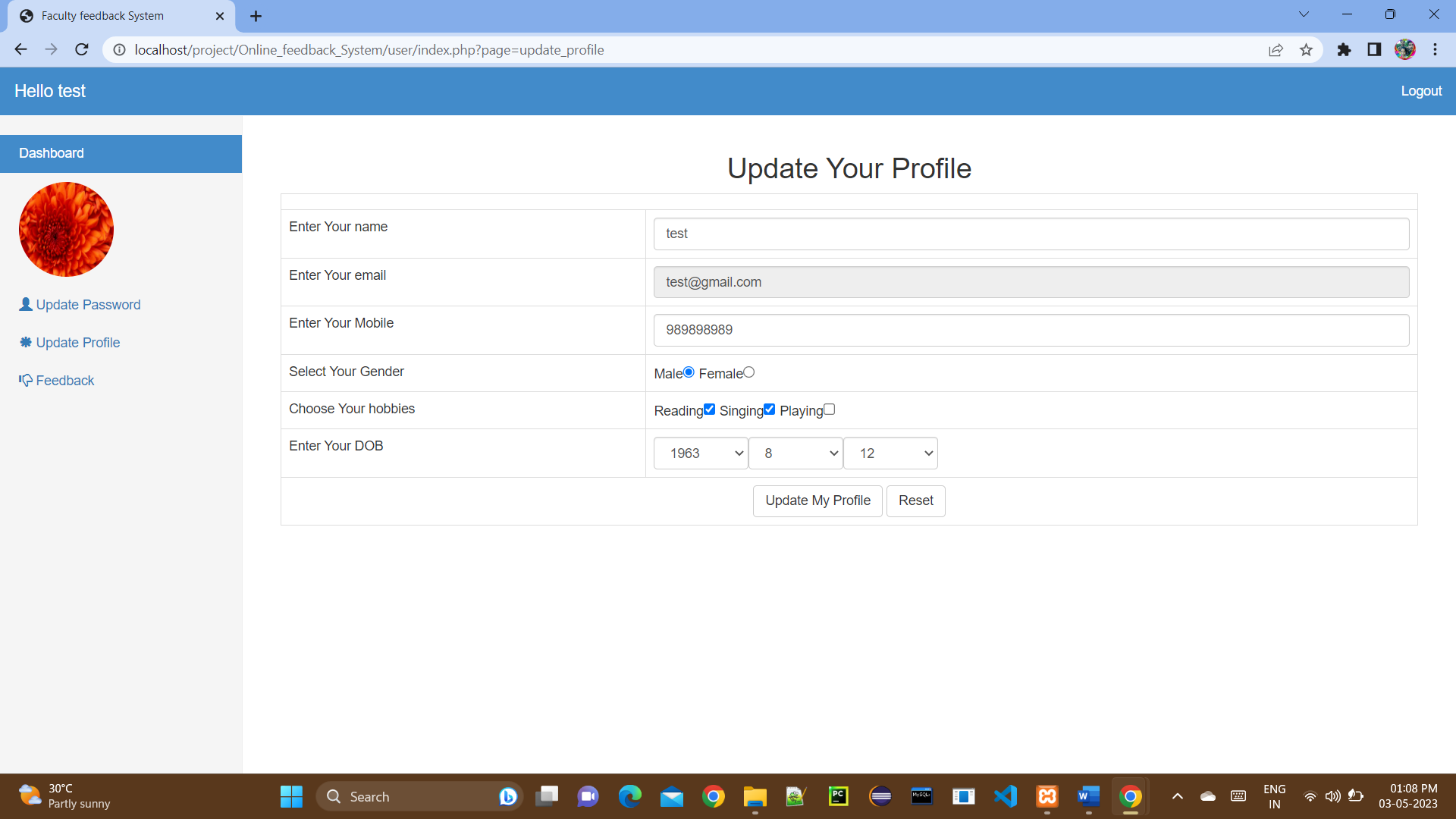
* **Dashboard**

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

****

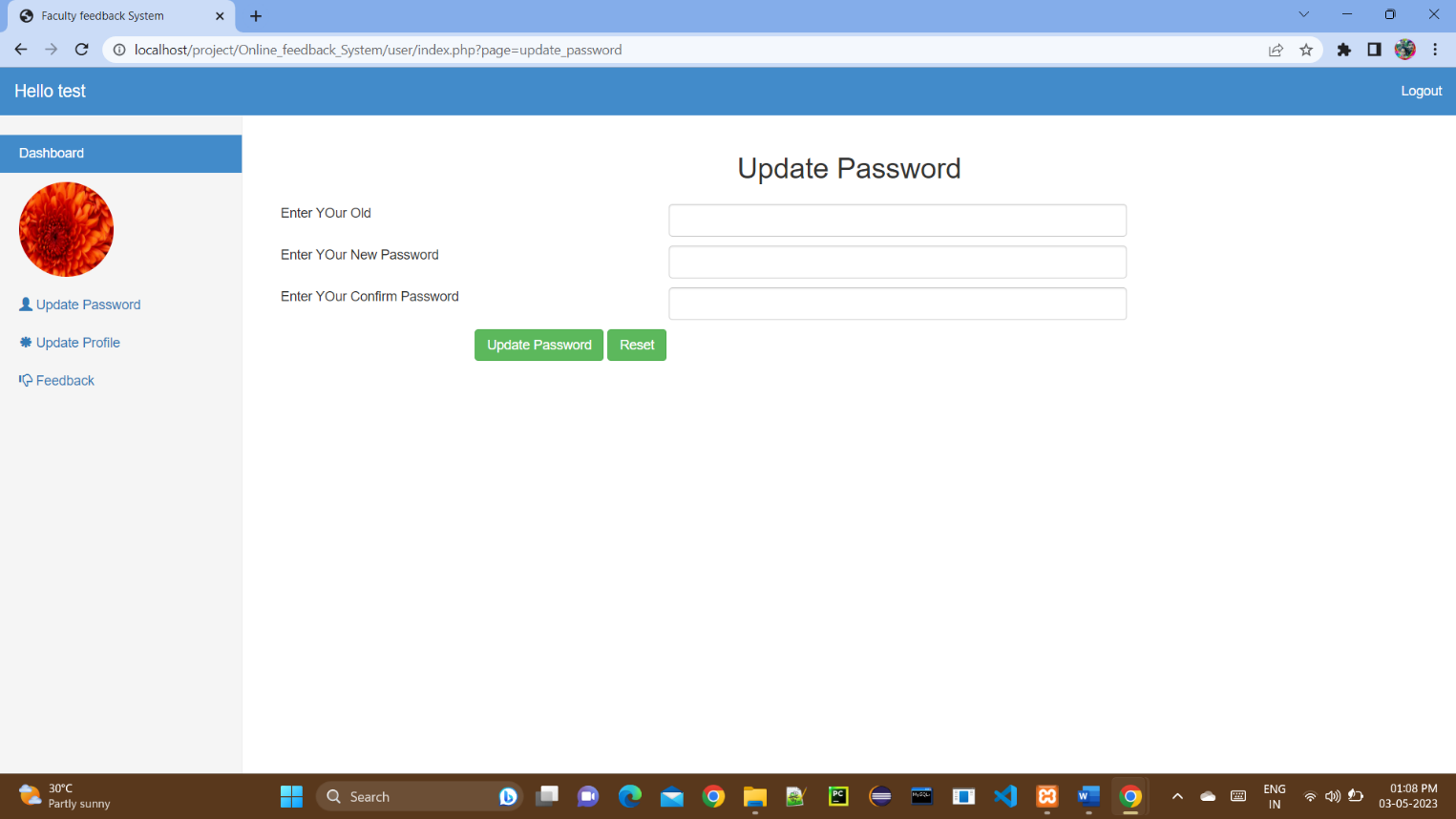
* **View/Edit Profile**

In this section a student can view and can also update its details.



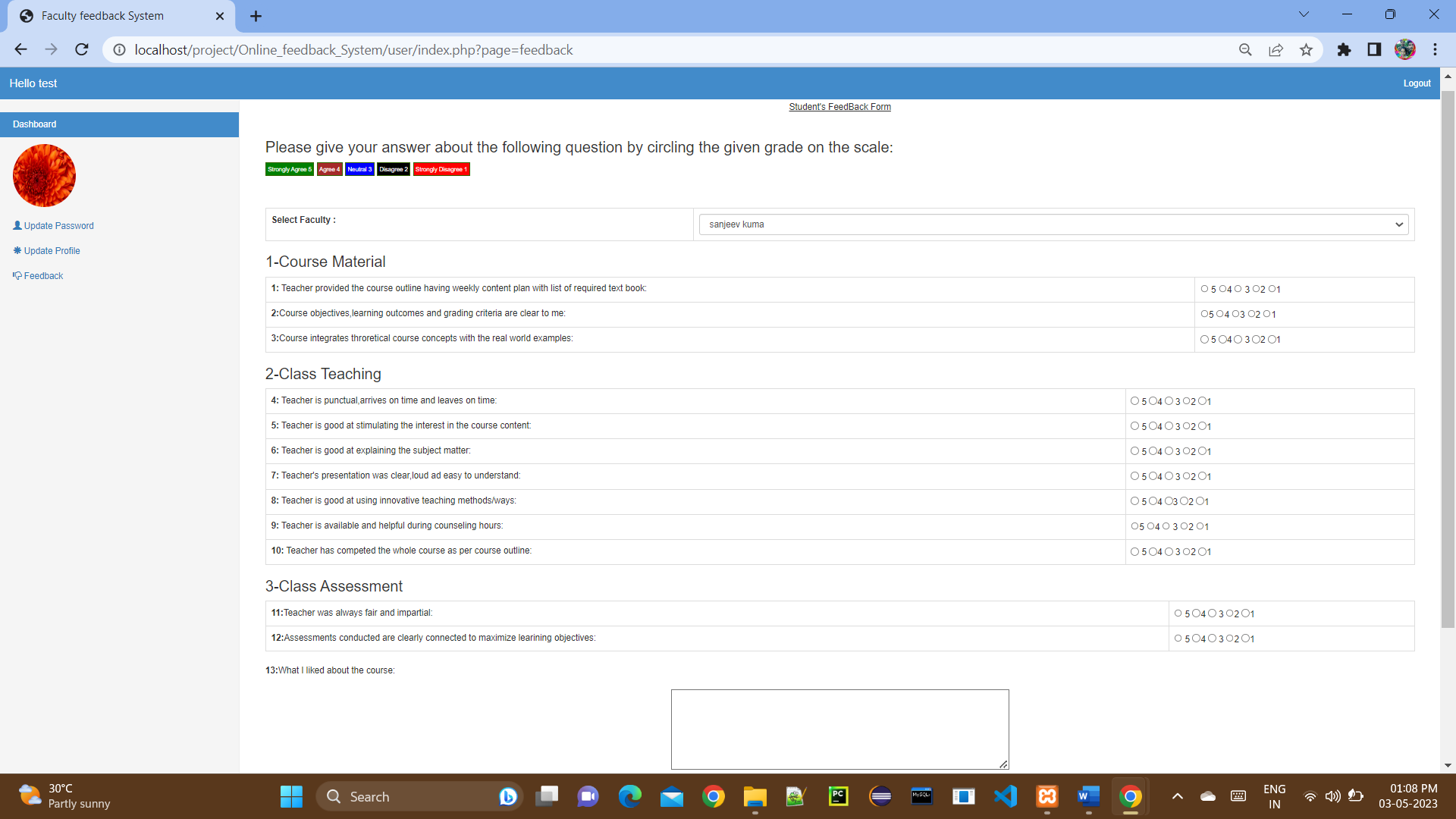
* **Update Password**

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



* **Give Feedback**

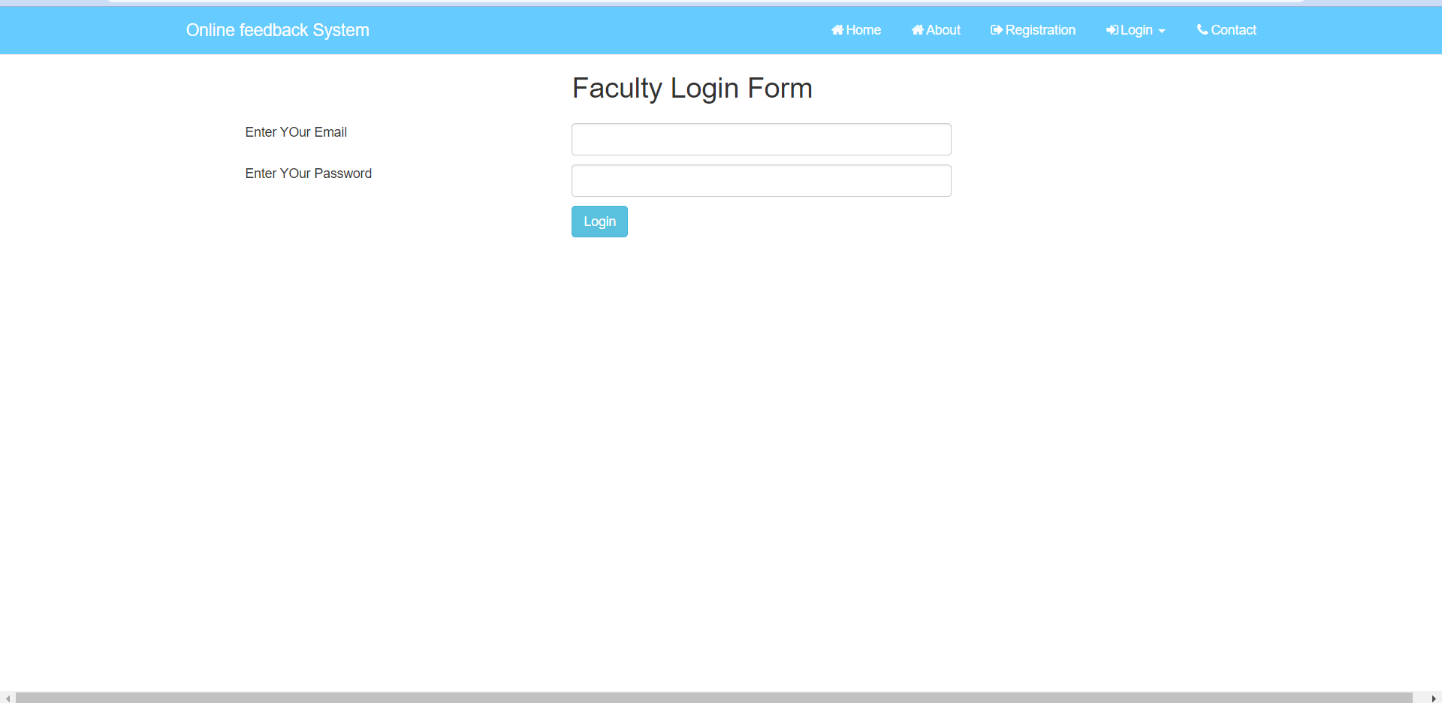
In this section student can give feedback to the faculty he/she has selected. We will show him only the professors which are taking courses in their semesters.

****

### 4.1.2 Faculty Module

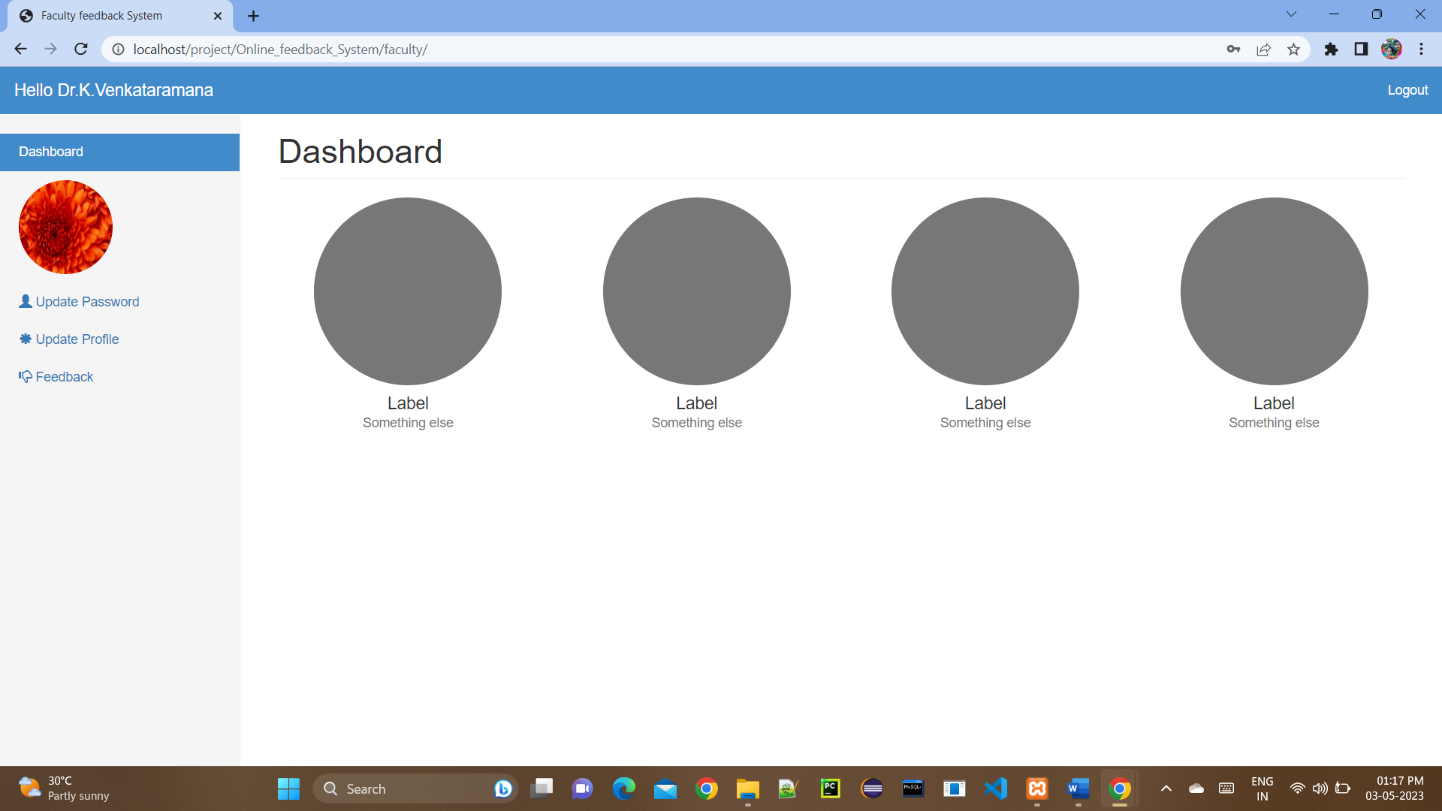
* **Login Page**

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.



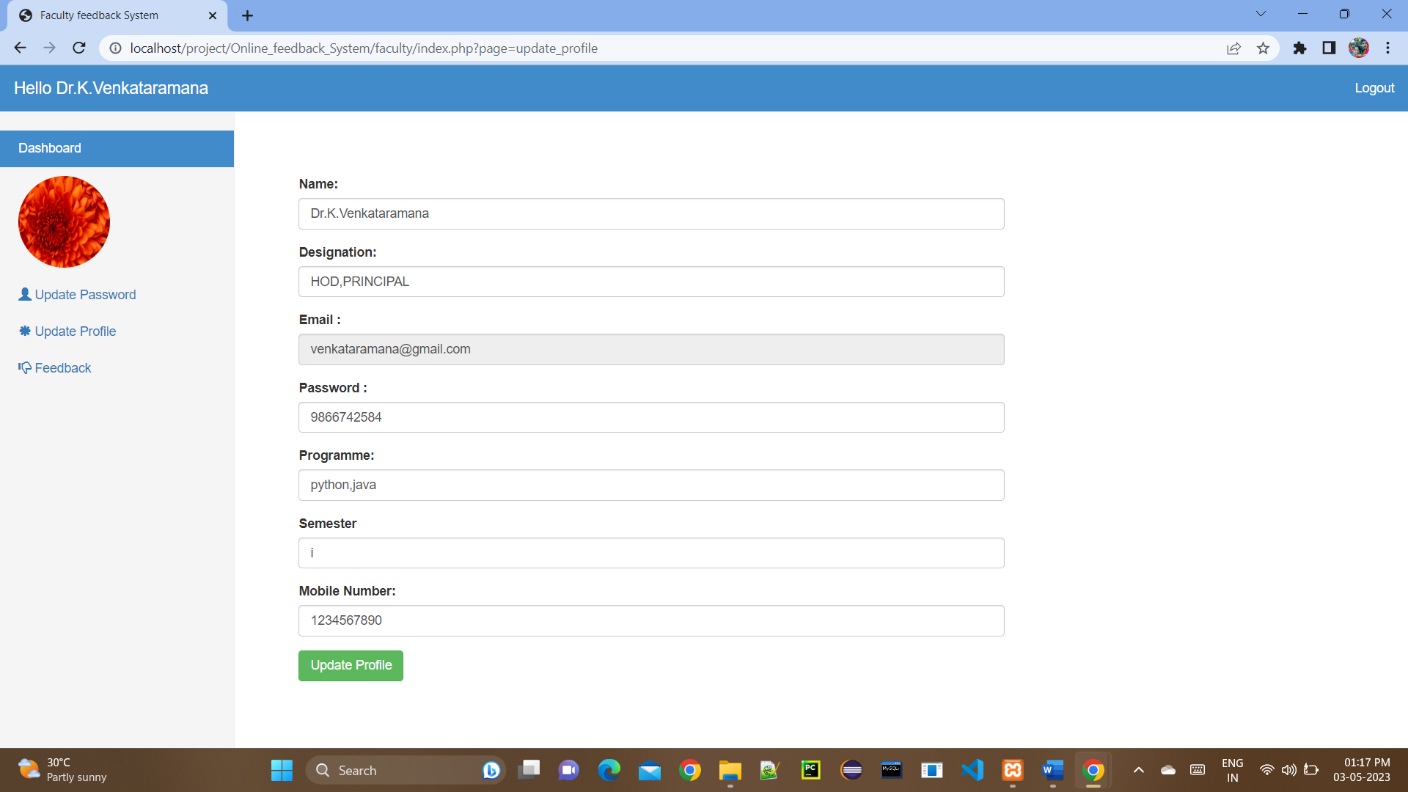
* **Dashboard**

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



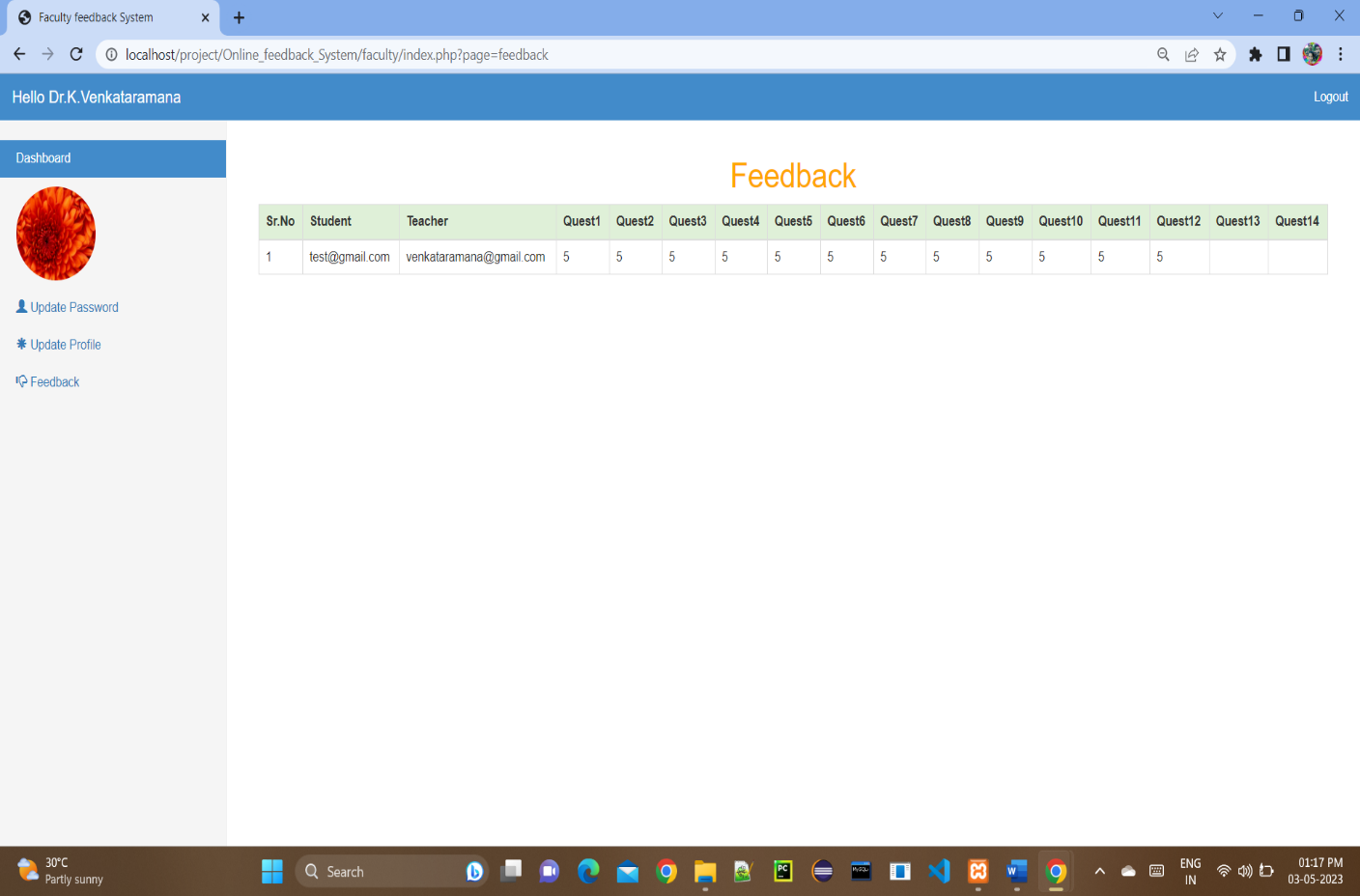
* **View/Edit Profile**

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



#### ● View Feedback’s

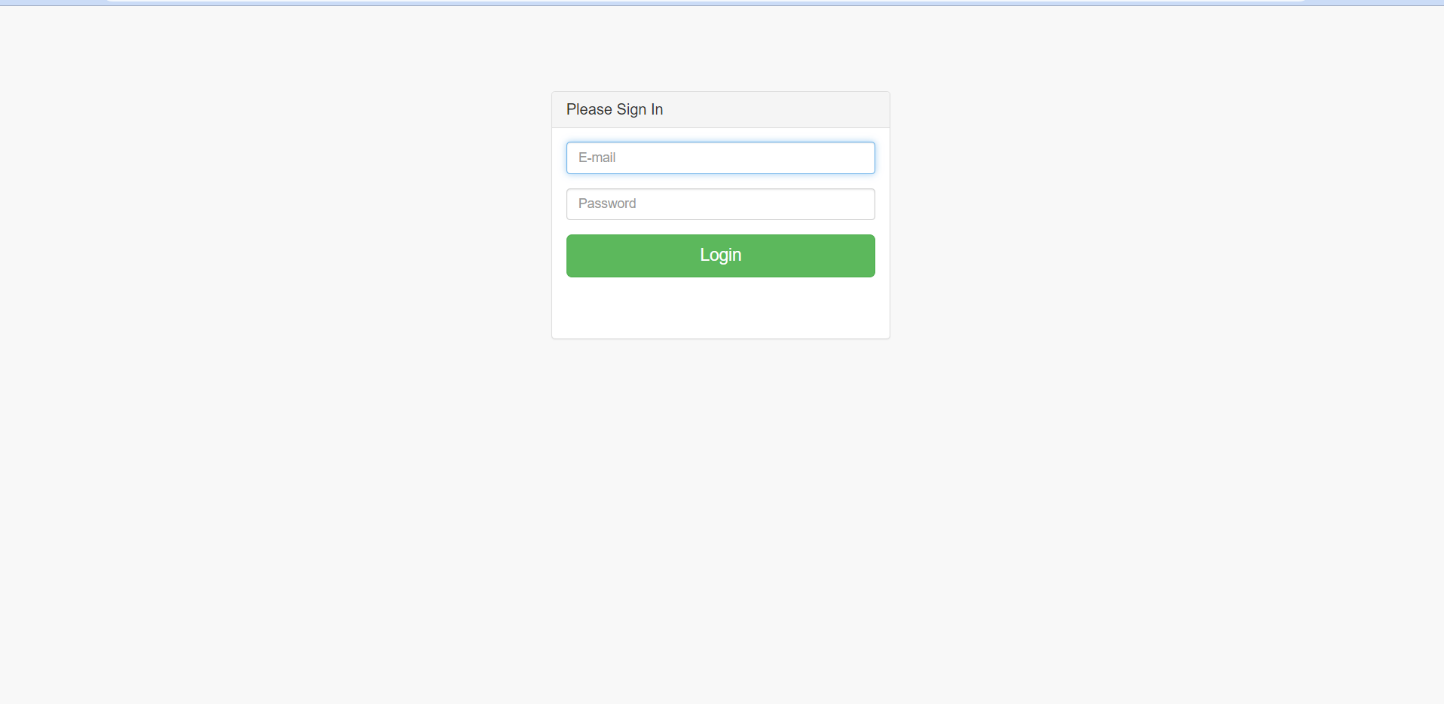
In this section the faculty can see all the feedback of student’s given to him. We will not show him/her the name of the students. So it will be completely Anonymously.



### 4.1.3 Admin Module [HOD]

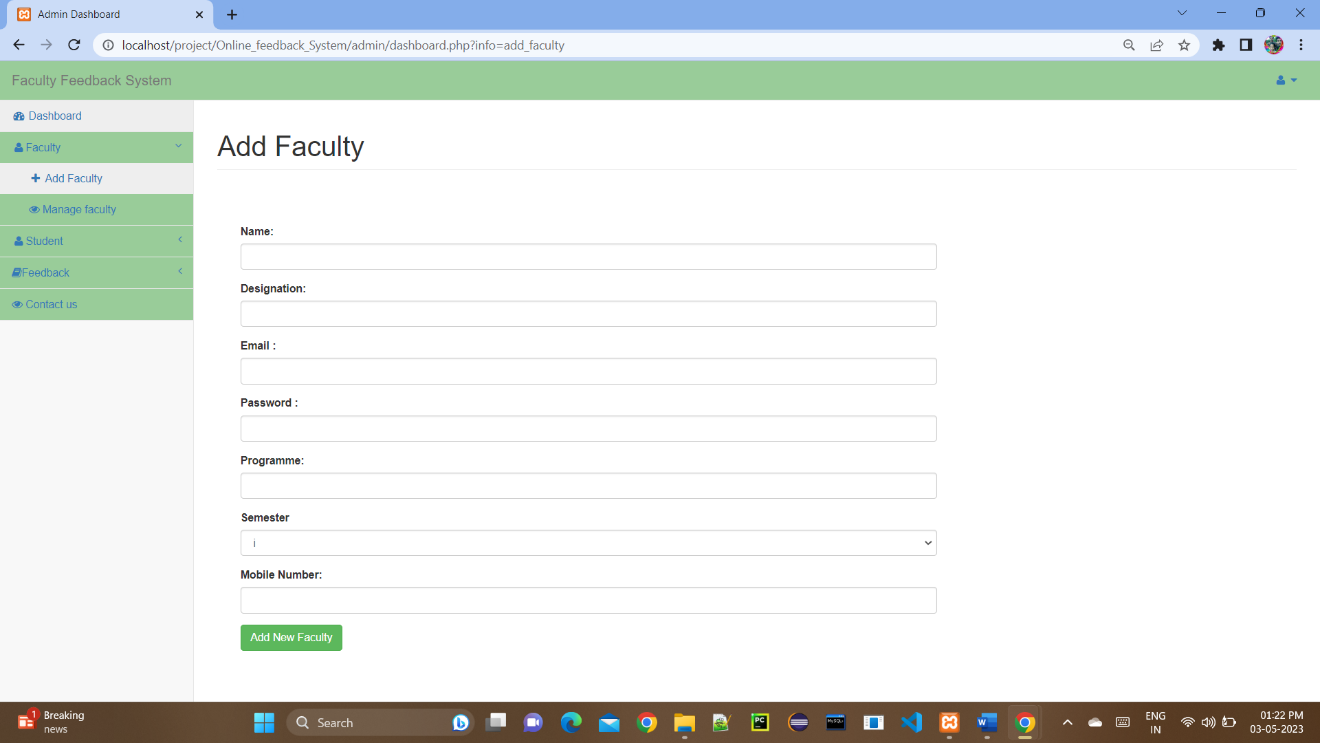
* **Login Page**

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.



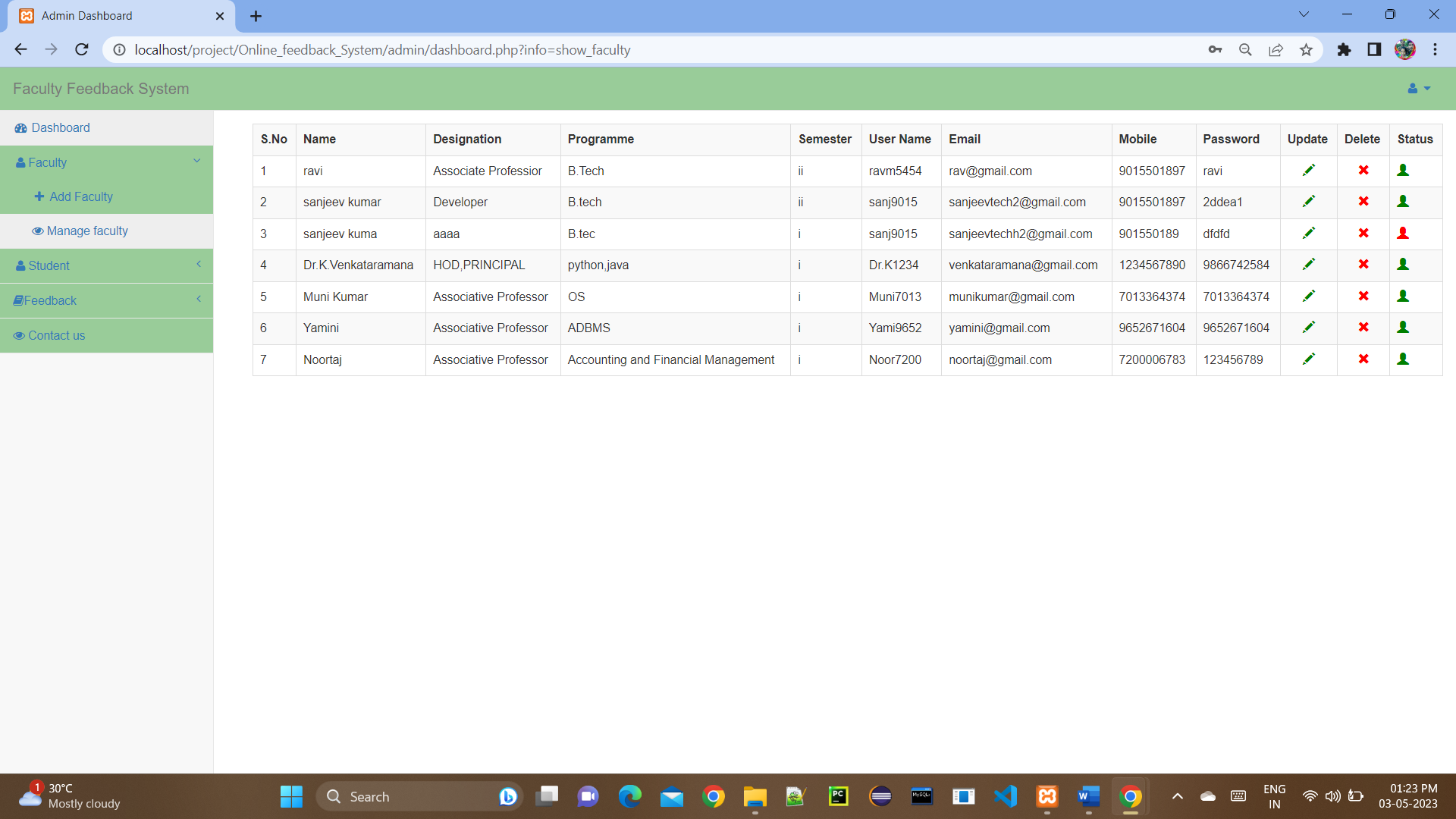
* **Add Faculty**

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



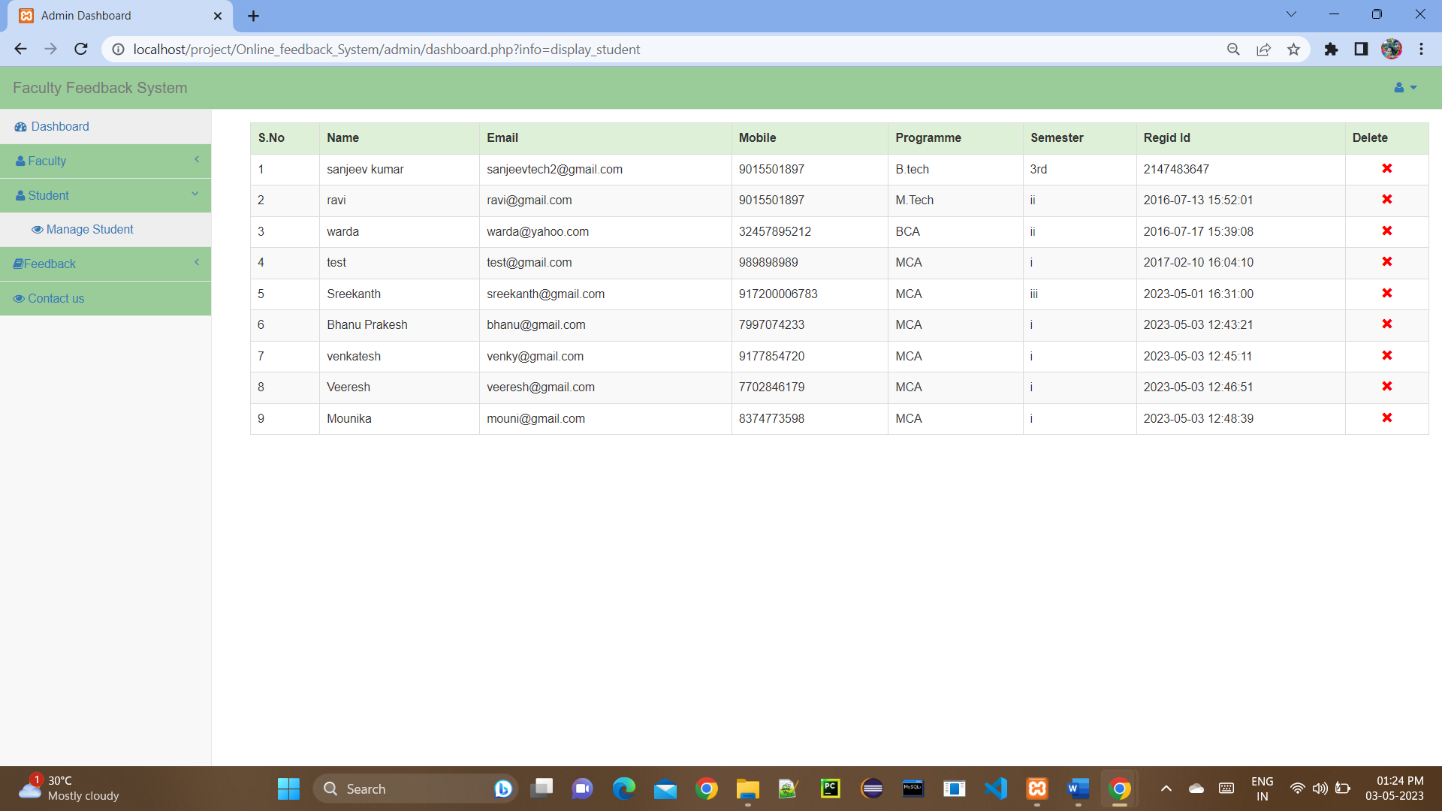
* **Manage Faculty [Update/Remove Faculty]**

Admin can update the details for faculties in the system and can also remove faculties from the system



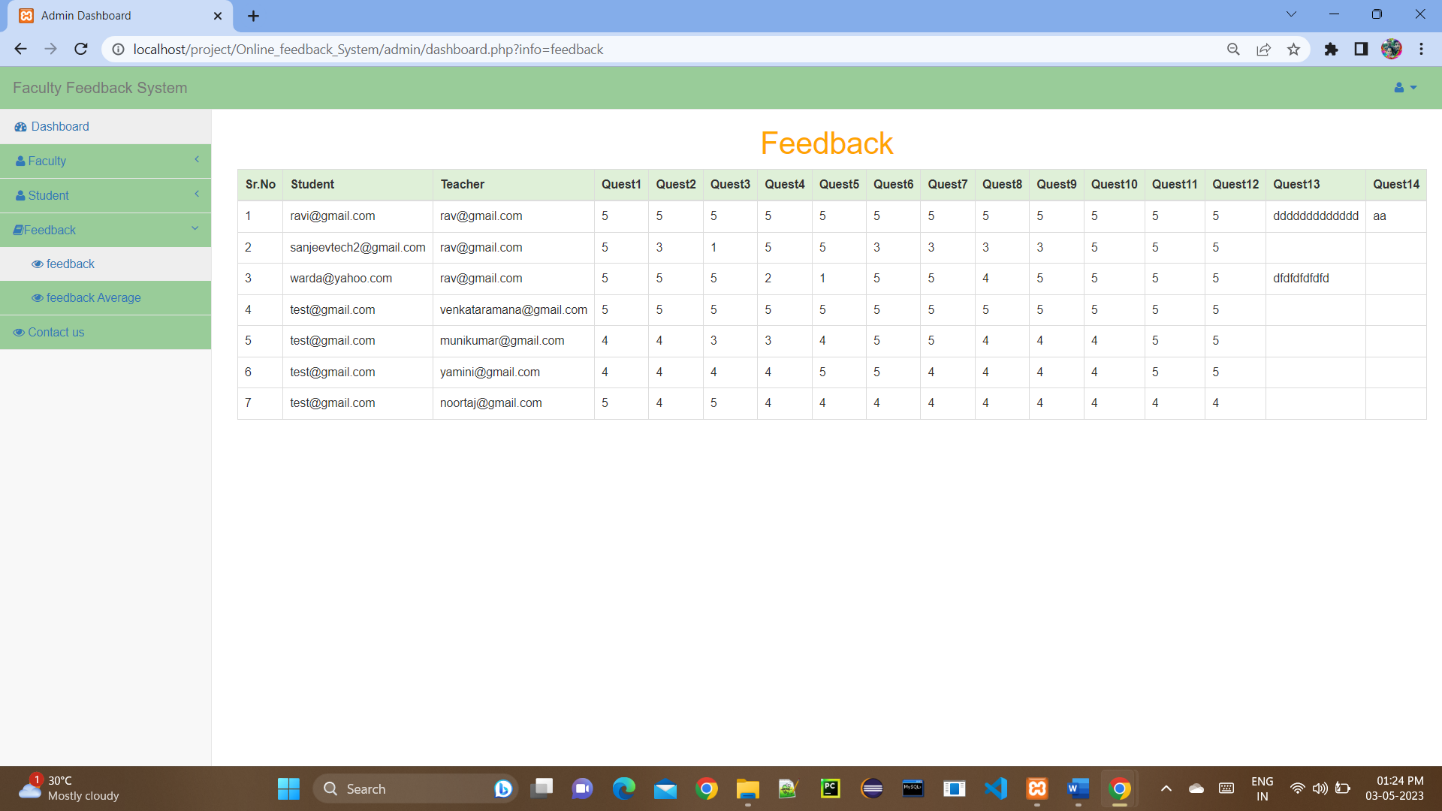
* **Manage Students [View/Update Students]**

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



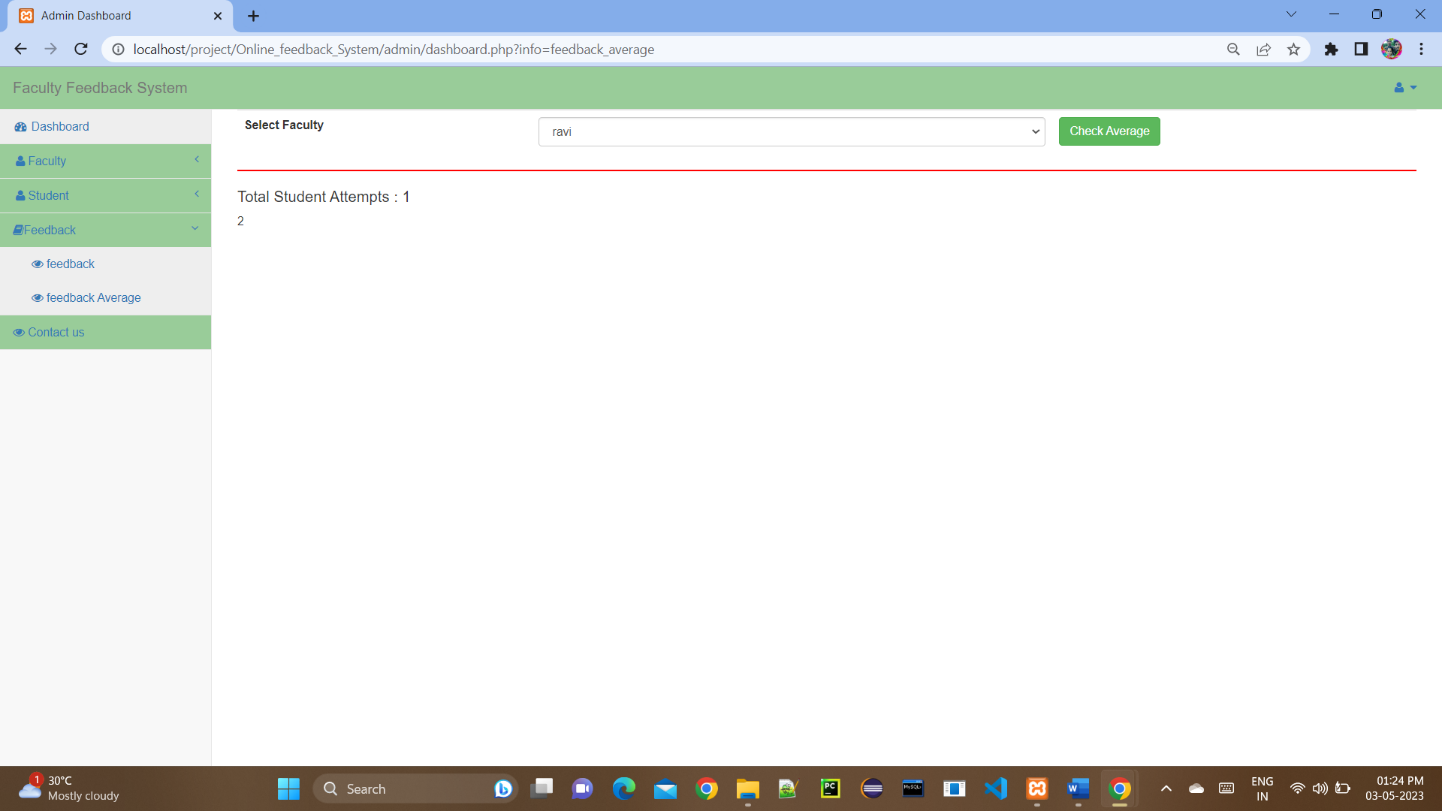
* **View All Feedbacks**

Admin can also view all the feedbacks given to all the faculties.



* **View Total Feedback Per Faculty**

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



**Project Source Code**

**PHP code for Home Page:**

<?php

session start();

require('dbconfig.php'); ?>

<!DOCTYPE html>

<html lang="eng">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content="">

<meta name="author" content="">

<title>Online feedback System</title>

<! -- Bootstrap Core CSS -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<! -- Custom CSS -->

<link href="css/modern-business.css" rel="stylesheet">

<! -- Custom Fonts -->

<link href="font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">

</head>

<body>

<! -- Navigation -->

<nav class="navbar navbar-default navbar-fixed-top" role="navigation" style="background: #66CCFF">

<div class="container" >

<! -- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="index. Php" style="color: #FFFFFF">Online feedback System</a>

</div>

<! -- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-right">

<li style="color: #FFFFFF">

<a style="color: #FFFFFF" href="index. Php"><i class="fa fa-home fa-fw"></i>Home</a>

</li>

<li style="color: #FFFFFF">

<a style="color: #FFFFFF" href="index.php?info=about"><i class="fa fa-home fa-fw"></i>About</a>

</li>

<li><a style="color: #FFFFFF" href="index.php?info=registration"><i class="fa fa-sign-out fa-fw"></i>Registration</a></li>

<li class="dropdown">

<a style="color: #FFFFFF" href="#" class="dropdown-toggle" data-toggle="dropdown" href="#"><i class="fa fa-sign-in fa-fw"></i>Login

<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="index.php?info=login">Student</a></li>

<li><a href="index.php?info=faculty login">Faculty</a></li>

<li><a href="admin">Admin</a></li>

</ul>

</li>

<li>

<a style="color: #FFFFFF" href="index.php?info=contact"><i class="fa fa-phone fa-fw"></i>Contact</a>

</li>

</ul>

</div>

<! -- /. navbar-collapse -->

</div>

<! -- /.container -->

</nav>

<?php

@$info=$\_GET['info'];

If ($info! ="")

{

if($info=="about")

{

Include ('about. Php');

}

else if($info=="contact")

{

Include ('contact. Php ');

}

else if($info=="login")

{

Include ('login. Php');

}

else if ($info=="faculty login")

{

include('faculty\_login.php');

}

else if($info=="registration")

{

Include ('registration. Php');

}

}

else

{

?>

<! -- slider start -->

<header id="my Carousel" class="carousel slide">

<! -- Indicators -->

<ol class="carousel-indicators">

<li data-target="#my Carousel" data-slide-to="0" class="active"></li>

<li data-target="#my Carousel" data-slide-to="1"></li>

<li data-target="#my Carousel" data-slide-to="2"></li>

</ol>

<! -- Wrapper for slides -->

<div class="carousel-inner">

<div class="item active">

<div class="fill" style="background-image:url('images/feedback.jpg');"></div>

<div class="carousel-caption">

<header>

<p>Welcome To</p>

<h2>Online Feedback System</h2>

</header>

</div>

</div>

<div class="item">

<div class="fill" style="background-image:url('images/feedback1.jpg');"></div>

<div class="carousel-caption">

<header>

<p>Creating</p>

<h2>A new way to give feedback</h2>

</header>

</div>

</div>

<div class="item">

<div class="fill" style="background-image:url('images/feedback3.jpg');"></div>

<div class="carousel-caption">

<header>

<p>Our moto</p>

<h2>We bring changes</h2>

</header>

</div>

</div>

<! -- Controls -->

<a class="left carousel-control" href="#my Carousel" data-slide="prev">

<span class="icon-prev"></span>

</a>

<a class="right carousel-control" href="#my Carousel" data-slide="next">

<span class="icon-next"></span>

</a>

</header>

<! -- slider -->

<! -- Page Content -->

<div class="container">

<div class="row">

<div class="col-lg-12">

<div class="col-sm-10" style="margin-top:60px; margin-bottom:80px">

<h2>About Faculty feedback System</h2>

Student Feedback system for College in PHP (Source code)

Here we have developed a faculty feedback system, which is generally used in the college to rate the faculty based on the performance...Here we have 2 modules such as administrator, student.

Administrator is the one who creates the student account by adding all student info and assigning the username and password.

Admin als0 checks the result once all students entered the feedback.

We can start the development from the login page, where we have given the option to login as admin and student...Here since we have only one admin account, so no need to create a database to store admin info...so the admin username is "admin" and password is "Sandeep"...select admin in the radio button and login

You can perform all admin actions such as login to the account and check result.

If you entered the student user and password and selected student option, then it will show all student information and let you enter the feedback based on the subject.

Before we can look into the php code, you need to create a database called "feed" with two tables in if. One as student and another one as take

</div>

<?php}?>

</div>

</div>

<! -- /.container -->

<! --<div class="navbar-fixed-bottom nav navbar-inverse text-center" style="padding:15px; height:40px; background: #66CCFF">

<span style="color: #FFFFFF">Developed By ....... <a href="http://www.phptpoint.com">Phptpoint.com</a> </span>

</div>---->

<! -- jQuery -->

<script src="css/jquery.js"></script>

<! -- Bootstrap Core JavaScript -->

<script src="css/bootstrap.min.js"></script>

<! -- Script to Activate the Carousel -->

<script>

$(‘. carousel'). carousel ({

interval: 5000 //changes the speed

})

</script>

</body>

</html>

**PHP code for Registration:**

<?php

extract($\_POST);

if(isset($save))

{

//check user already exists or not

$sql=mysqli\_query ($conn,"select \* from user where email='$e'");

$r=mysqli\_num\_rows($sql);

if($r==true)

{

$err= "<font color='red'><h3 align='center'>This user already exists</h3></font>";

}

else

{

//dob

$dob=$yy."-”. $mm."-”. $dd;

//hobbies

$hob=implode (",”, $hob);

//image

$image Name=$\_FILES['img’] ['name'];

//encrypt your password

$pass=md5($p);

$query="insert into user values ('','$n','$e','$pass','$mob','$pro','$sem','$gen','$hob','$imageName','$dob', now ())";

mysqli\_query ($conn, $query);

//upload image

mkdir("images/$e");

move\_uploaded\_file($\_FILES['img']['tmp\_name'],"images/$e/”. $\_FILES['img']['name']);

$err="<font color='blue'><h3 align='center'>Registration successful !!<h3></font>";

}

}

?>

<div class="row">

<div class="col-sm-2"></div>

<div class="col-sm-8">

<form method="post" enctype="multipart/form-data">

<table class="table table-bordered" style="margin-bottom:50px">

<caption><h2 align="center">Registration Form</h2></caption>

<Tr>

<Td colspan="2"><?php echo @$err;></Td>

</Tr>

<tr>

<td>Enter Your name</td>

<Td><input type="text" name="n" class="form-control" required/></td>

</tr>

<tr>

<td>Enter Your email </td>

<Td><input type="email" name="e" class="form-control" required/></td>

</tr>

<tr>

<td>Enter Your Password </td>

<Td><input type="password" name="p" class="form-control" required/></td>

</tr>

<tr>

<td>Enter Your Mobile </td>

<Td><input type="text" name="mob" class="form-control" required/></td>

</tr>

<tr>

<td>Select Your Programme</td>

<Td><select name="pro" class="form-control" required>

<option>BCA</option>

<option>MCA</option>

<option>B.Tech</option>

<option>M.Tech</option>

</select>

</td>

</tr>

<tr>

<td>Select Your Semester</td>

<Td><select name="sem" class="form-control" required>

<option>i</option>

<option>ii</option>

<option>iii</option>

<option>iv</option>

<option>v</option>

<option>vi</option>

<option>vii</option>

<option>viii</option>

</select>

</td>

</tr>

<tr>

<td>Select Your Gender</td>

<Td>

Male<input type="radio" name="gen" value="m"/>

Female<input type="radio" name="gen" value="f"/>

</td>

</tr>

<tr>

<td>Choose Your hobbies</td>

<Td>

Reading<input value="reading" type="checkbox" name="hob []"/>

Singing<input value="sing in" type="checkbox" name="hob []"/>

Playing<input value="playing" type="checkbox" name="hob []"/>

</td>

</tr>

<tr>

<td>Upload Your Image </td>

<Td><input type="file" name="img" class="form-control" required/></td>

</tr>

<tr>

<td>Enter Your DOB</td>

<Td>

<select style="width:100px; float: left" name="yy" class="form-control" required>

<option value="">Year</option>

<?php

for ($i=1950; $i<=2016; $i++)

{

echo "<option> “. $i."</option>";

}

?>

</select>

<select style="width:100px; float: left" name="mm" class="form-control" required>

<option value="">Month</option>

<?php

for ($i=1; $i<=12; $i++)

{

echo "<option> “. $i."</option>";

}

?>

</select>

<select style="width:100px; float: left" name="dd" class="form-control" required>

<option value="">Date</option>

<?php

for ($i=1; $i<=31;$i++)

{

echo "<option>".$i."</option>";

}

?>

</select>

</td>

</tr>

<tr>

<Td colspan="2" align="center">

<input type="submit" value="Save" class="btn btn-info" name="save"/>

<input type="reset" value="Reset" class="btn btn-info"/>

</td>

</tr>

</table>

</form>

</div>

<div class="col-sm-2"></div>

</div>

</body>

</html>

**PHP code for Login Page:**

<?php

extract($\_POST);

if(isset($save))

{

if($e=="" || $p=="")

{

$err="<font color='red'>fill all the fields first</font>";

}

else

{

$pass=md5($p);

$sql=mysqli\_query ($conn,"select \* from user where email='$e' and pass='$pass'");

$r=mysqli\_num\_rows($sql);

if($r==true)

{

$\_SESSION['user']=$e;

header('location: user');

}

else

{

$err="<font color='red'>Invalid login details</font>";

}

}

}

?>

<div class="row">

<div class="col-sm-2"></div>

<div class="col-sm-8">

<form method="post">

<div class="row">

<div class="col-sm-4"></div>

<div class="col-sm-4"><h2>Login Form</h2></div>

</div>

<div class="row">

<div class="col-sm-4"></div>

<div class="col-sm-4"><?php echo @$err;?></div>

</div>

<div class="row" style="margin-top:10px">

<div class="col-sm-4">Enter Your Email</div>

<div class="col-sm-5">

<input type="email" name="e" class="form-control"/></div>

</div>

<div class="row" style="margin-top:10px">

<div class="col-sm-4">Enter Your Password</div>

<div class="col-sm-5">

<input type="password" name="p" class="form-control"/></div>

</div>

<div class="row" style="margin-top:10px">

<div class="col-sm-4"></div>

<div class="col-sm-8">

<input type="submit" value="Login" name="save" class="btn btn-info"/>

</div>

</div>

</form>

</div>

</div>

**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 verifying each and every data entered by user [student] while registration, while login and while giving feedback.
* We are also validating and verifying 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 html entities () 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.

* 1. If any user wants 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.
  2. 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](http://php.net/manual/en/ref.filter.php) will allow the input data to the php script to be [sanitized](http://php.net/manual/en/filter.filters.sanitize.php) or [validated](http://php.net/manual/en/filter.filters.validate.php) in [many ways.](http://php.net/manual/en/filter.filters.php) 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 is 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 taking 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.