Capstone Project Report

on

IAP Evaluation Portal (Industrial Attachment Program)

submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Computer Science and Engineering Department

by

Abhinav Garg
(101303004)
Akshit Arora
(101303012)

Arush Nagpal
(101303034)
Chahak Gupta
(101303041)

Under the supervision of

Dr. Prashant Singh Rana, Assistant Professor, CSED
Dr. Ajay Batish, Professor, MED

Thapar University,
Patiala, Punjab, India - 147004.
December 2016

Abstract

In every college/university, students undergo 6-months of industrial training in their pre-final or final year of Bachelors of Engineering. All the records need to be maintained by the university about the students, companies in which they are undergoing training and also the feedback from the industry process. All of this work is done manually. As the number of students in the university are increasing per year, this work is becoming cumbersome. Also the juniors face difficulty in getting prior information about the companies their seniors joined which may help them securing internship in good companies. Departments are not able to provide them full help as records are sometimes misplaced. The primary objective of the portal is to give the university management a tool to help them manage the student training data. The automation of records of training semester will simplify the arduous task of manual maintenance. It aims at providing the students and teachers with the facility of submission, guiding, feedback and grading them on-line. Also the industry mentors would be able to provide feedback of the intern's performance.

Table of Contents

List of Figures

List of Tables

1 Introduction

Industrial Attachment Programme is a part of curriculum at Thapar University. Students undergo 6-months of industrial training in their pre-final or final year of Bachelors of Engineering. All the records need to be maintained by the university about the students, companies in which they are undergoing training and also the feedback from the industry process. Currently, this work is done manually in mechanical department.

The primary objective of the portal is to give the university management a tool to help them manage the student training data. The automation of records of training semester will simplify the arduous task of manual maintenance. It aims at providing the students and teachers with the facility of submission, guiding, feedback and grading them on-line. Also the industry mentors would be able to provide feedback of the intern's performance. The primary audience include the students going out for 6 month industrial training, faculty members and industry mentors allotted to those students. The personal data of all the students, their performance report during training period can be accessed from anywhere at any time giving the user the convenience of not going through the manual procedure of accessing records. The faculty and industry mentors will be able to evaluate the students (based on rubrics provided in the knowledge base of the website) and the students in turn, would be able to submit joining, intermediate and final reports online through the website. There will be individual record maintained for each student on the server will be kept for easy access to faculty members and mentors. Record of all the pass outs will also be available to all the students for consultation. The students who have completed their training are obliged to comment about their industry experience and living conditions that would serve as a reference for the students that would be going the forthcoming year. Students may contact their seniors too before applying for training in industry, hence making the experience smooth and easy.

2 Literature Survey

Presently, Computer Engineering Department is using IAP. This serves as the blueprints for IAP for mechanical department. As the requirements of both the departments are different so new portal had to be devised. Other departments are still manually doing the work. For developing this software, a reference was taken from the university's online hostel allotment system to account for the technology stack and handling of a large number of concurrent users.

3 Motivation

The primary motivation behind developing this software was to apply our technical knowledge to create a software that would help a person or a body of people in reducing the manual workload. In every college/university, students undergo 6-months of industrial training in their pre-final or final year of Bachelors of Engineering. All the records need to be maintained by the university about the students, companies in which they are undergoing training and also the feedback from the industry process. All of this work is done manually. As the number of students in the university and hence the training records to be managed are increasing per year, this work is becoming cumbersome. Also the juniors face difficulty in getting prior information about the companies their seniors joined which may help them securing internship in good companies. Departments are not able to provide them full help as records are sometimes misplaced. The primary objective of the portal is to give the university management a tool to help them manage the student training data as well as give university students access to past training records along with contact information of their seniors. The automation of records of training semester will simplify the arduous task of manual maintenance. It aims at providing the students and teachers with the facility of submission, guiding, feedback and grading them on-line. It would provide a transparency in the industrial training management system, where deadlines could be easily achieved and the users of the system are timely reminded of the documents that need to be submitted to the university.

4 Gaps and Objectives

4.1 Gaps

1. Students undergo 6-months of industrial training in their pre-final or final year of Bachelors of Engineering. All the records need to be maintained by the university about the students, companies in which they are undergoing training and also the feedback from the industry process. All of this work is done manually.As the number of students in the university are increasing per year, this work is becoming cumbersome.
2. Also the juniors face difficulty in getting prior information about the companies their seniors joined which may help them securing internship in good companies.
3. Departments are not able to provide them full help as records are sometimes misplaced.

4. This software can help to automate the current manual system and be extended to all the departments of the university
5. A reminder and notification system might be incorporated to promote submissions at expected time.

4.2 Objectives

1. The primary objective of the portal is to give the university management a tool to help them manage the student training data.

2. The automation of records of training semester will simplify the arduous task of manual maintenance.

3. It aims at providing the students and teachers with the facility of submission, guiding, feedback and grading them on-line.

4. The industry mentors would be able to provide feedback of the intern's performance.

5. Administrator would be able to download, maintain and manage mappings and records in an efficient way.

5 Scope of the project

The website is currently being prepared for Thapar University's Mechanical Department but it can be extended to any number of departments in any college or university that works under the same structure. Thus, it can be scaled up to serve as the default training portal for the whole university. It would ease the process of maintaining and accessing performance records of students undertaking 6 month industrial training. It will also ease the process of evaluating student's performance by the faculty members and mentors and help in reducing human error. The proposed system will keep all the data in place avoiding the risk of loss of any kind and act as future reference for the students. Since the system will link the administrator, student, faculty and the industry mentor, there will be no need for manually keeping records.

6 Methodology Used

In this web-app we have developed responsive web interface using bootstrap framework (Bootstrap 4) (1). PHP (CodeIgniter (5) framework) is used as a back-end scripting language. Relational Database is used using MySQL(4). Website is currently hosted at hostinger http://iapthapar.esy.es/ . To monitor the system online, google analytics will be used. The responsive framework enables the website to be ported into mobile applications very easily. The modular framework will enable us to extend the framework easily with easy documentation.

6.1 Role of Team Members

1. **Abhinav Garg**: Developed modules for industry mentor section.
2. **Akshit Arora**: Developed modules for faculty advisor section.
3. **Arush Nagpal**: Developed modules for admin section.
4. **Chahak Gupta**: Developed modules for student section.

6.2 Tool/Technique Used

1. **HTML5 + CSS3**: Used for formatting and designing of web pages.
2. **Bootstrap + Kadmin Framework**: Framework used for responsive design layout.
3. **CodeIgniter**: Framework used for developing the back-end of the portal.
4. **PHP**: Used as a Backend scripting language.
5. **MYSQL**: Used for Database management.
6. **Google Analytics**: Used for monitoring the online transactions occurring in the system.

7 Project analysis and design

7.1 Software Requirements Specification (SRS)

Product Perspective:
The IAP Evaluation portal is a completely new self-contained product in the existing system of Thapar University. This will be developed from the scratch and all the website functions are currently handled manually.

Product Features:
The website will have four kinds of users which are the student, the administrator (IAP Coordinator), the faculty member and the industry mentor. Their major functions are described as under:

Administrator:

1. Validating student's training details.
2. Full access management.
3. Allocating cities to faculty.
4. Allotting faculty mentors to students.
5. Start or end the grading procedure.
6. Change details of students or faculty to system, and maintain the system.
7. Log, report and download spreadsheets of the training system data.

Students:

1. Registering on website with their training details.
2. Updating their training details timely.
3. Uploading their joining report, goals reports, midway report, emergency contact details and final report timely.
4. Receive feedback from the students and faculty.

Mentor:

1. Registering on website.
2. Uploading Final assessment timely.
3. Updating details of students working on them.
4. Communicate with the faculty mentor of the student.

Faculty:

1. Registering on website.
2. Uploading assessment and grade the students timely.
3. Choosing city preferences.
4. Communicate with the industry mentors

General Audience:

1. View the trainings undergone by students in the previous years.
2. View the past companies for training purpose.

User Classes and Characteristics

1. Students undertaking 6 months summer training: Role of students in the system involves maintaining the training records corresponding to their ID.
2. Faculty: Managing, evaluating students' performance on the basis of the reports and industry mentor's feedback.
3. Industry Mentor: Evaluating student on the basis of his performance in the training period and providing feedback for the same.
4. Administrator: Linking student to faculty members and managing the website. Manipulating and cross verification of student records.

Operating Environment:
The system will work on Internet thus making it accessible from any platform. A browser is needed to access the Internet which in turn will allow the access to the particular website. The website will be made responsive so that it works on any device.

Design and Implementation Constraints:

Number of hits on website: Since the website will be deployed on a server, the processing power of the server will act as a constraint on the number of simultaneous hits that the website can take. However there is a safe assumption that at a particular time, even at peak usage, no more than 100 people will visit the site.

Operating Platform:

The portal is a web service. It does not matter which device or operating system the user uses to access the website. The user only needs a web browser to access it. Also any front-end framework requirements are being delivered by content delivery networks. Hence there are no constraints on the operating platform.

User Documentation:

The software will be accompanied with a user help section integrated in the system itself to make it easy for new users to understand the functioning of the system. An online video tutorial shall also be made available at the time of system delivery to further ease the process of system learning.

Assumptions and Dependencies:

The project is based on the assumption that the student regularly inputs valid information into the system.

System Features:

1. Student, Faculty and Industry Mentor Registration

Description and Priority:

The task of registration of the three main users of the system is dealt in this feature. Every user will have their own username and password. If this feature fails, we might end up getting the user into wrong situation altogether.

Stimulus/Response Sequences:

Stimulus: The user is unregistered and submits personal details.

Response: User is registered and can sign in using password provided through the registered email.

Stimulus: The user is unregistered and enters incorrect confirmation password while registering.

Response: Display an error message.

Functional Requirements:

REQ-1: For student, the required details are: Roll Number, Branch, Full Name, E-mail, Phone number, Company and City of internship. Also an emergency contact form will be filled by the student. An example form is mentioned in appendix 3 at the end of the SRS.

REQ-2: For faculty, the required details are: Initials, Full Name, and Designation

REQ-3: For industry mentor, the required details are: Code provided by student, username, E-mail

REQ-4: After any of the above users have signed up, a random password will be generated and sent to their respective e-mail. This will serve as e-mail validation and clicking on the link in the e-mail, the user will be redirected to password panel where they can set their password themselves. After that, dashboard will be accessible.

2. Student, Faculty and Industry Mentor Login

Description and Priority:

This feature is required to give all the registered users to their respective dashboards, where they can perform relevant tasks (Example for student: updating emergency contact details, for faculty: update city preferences, for industry mentor: update student evaluation marks, for admin: approving student details for further processing).

Stimulus/Response Sequences:

Stimulus: The user is logged out and dashboard is inaccessible. They fill-in their username and correct password (used in the sign-up phase).

Response: User redirected to their respective dashboard.

Stimulus: The user is unregistered and tries to sign in. Or the user is registered and puts incorrect username / password.

Response: Display an error message.

Functional Requirements:

REQ-1: For student, the tasks they can do with their dashboard are: edit personal details, submit their respective company and city details, upload their reports timely (joining report, goals report, emergency contact details, midway report and final report) and edit password.

REQ-2: For faculty, the tasks they can do with their dashboard are: upload assessment of every student assigned to them, edit personal details and set city preferences.

REQ-3: For industry mentor, the tasks they can do with their dashboard: upload assessment details of every student working under them.

REQ-4: After any major change is made by student, industry or faculty a notification is generated for relevant users.

3. Student Assessment Upload

Description and Priority:

This feature is required to compute the final assessment of student from industry mentor and faculty.

Stimulus/Response Sequences:

Stimulus: The reports and company details are submitted by student

Response: Faculty / Industry mentors submit the evaluation details.

Functional Requirements:

REQ-1: For faculty, the evaluation form needs to be submitted. And example report has been mentioned in Appendix 1 at the end of the report.

REQ-2: For industry mentor, the evaluation form needs to be submitted. And example report has been mentioned in Appendix 2 at the end of the report.

REQ-3: When final student evaluation forms have been submitted by both faculty and industry mentor, admin is notified and student's final evaluation can be locked and approved.

External Interface Requirements:

1. User Interfaces:
The interface provides a variety of options to the users. It allows the following features:

2. Student Interface: It opens up after student logs in from General Audience Interface. It will contain the following major panels:

1. Login Page: Meant for opening up administrator, faculty, mentor and student interface
2. Training details: options for editing the details of the training undertaken by the student.
3. Documents: options for uploading the joining report, emergency details, goals report
4. Publications: options for addition / deletion of research publication records.
5. Profile: options editing the profile page of the student.
6. Settings: Access to various student settings
7. Administrator interface: It opens up after administrator logs in from General Audience Interface
8. Student records: Access to student information.

9. Faculty Records: access to faculty information.
10. Mentor panel: evaluating students performance and providing feedback for the same
11. Settings: Administrator settings manipulation.

3. Hardware Interfaces

IAP portal is a website that can be accessed from any device be it a computer desktop, laptop, tablets, phones or any other device having a NIC (Network Interface Card).

4. Software Interfaces

As mentioned earlier the project is a website hence can be accessed from any device. It does not require any particular operating system to execute. All it needs is just a modern web browser (like Safari, Firefox, and Google Chrome etc.)

5. Communications Interfaces

The students are required to send request through email to the administrators for editing their training details. Mentors and faculty members can send email to each other to discuss about students performance.

Other Nonfunctional Requirements:

Performance Requirements:

1. Reliability: Auxiliary storage devices must be available for backing up the data. The internet connection in the school must be stable most of the time.
2. User Interface and Human Factors: The training for using the admin panel is required to be provided so as to prevent him/her from making errors.
3. Availability: The website shall be available, up and running for 24*7 throughout the year except due to the routine maintenance activities.

Safety Requirements:

There are no specific safety requirements associated with the proposed system. The portal executes on well-known and commonly used hardware which does not cause any safety hazards.

Security Requirements:

Security is one of the points of concern in this system.

1. A validation link will be send to the registered users.

2. The password details that are stored in the database should be in encrypted format so that the data cannot be read directly.
3. Only a registered user should be able to access the account and perform various tasks such as updating the training details, editing user profiles etc.
4. PHP functions should be used to prevent the users from entering malicious queries into the database.
5. Various security attacks like CSRF, session forgery and cross server scripting attacks will be prevented using secure PHP coding practices.

7.2 Data Flow Diagram

This data flow diagram (DFD) is a graphical representation of the flow of data through IAP, modelling its process aspects.

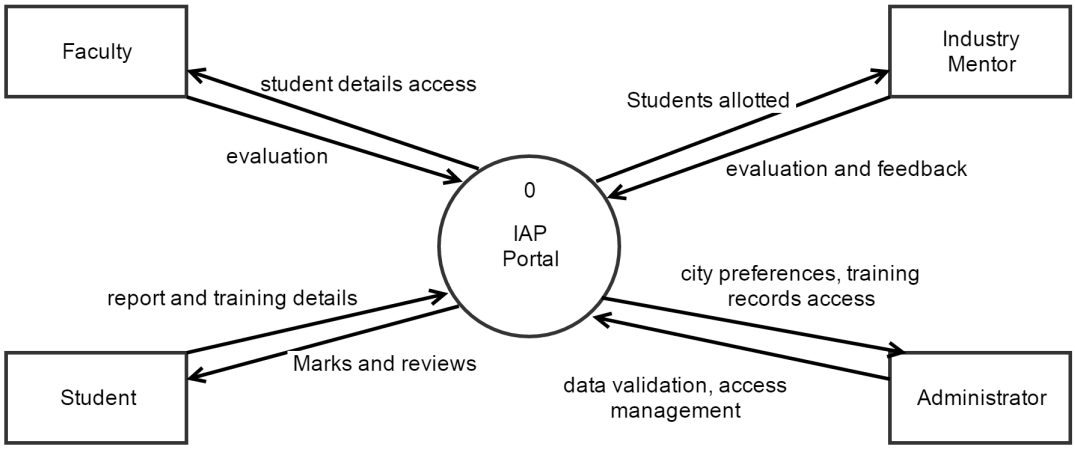

Figure 1: Level 0 DFD

DFD level 1 shows the process of registration, evaluation in detail. This figure shows the Admin operations' detail.

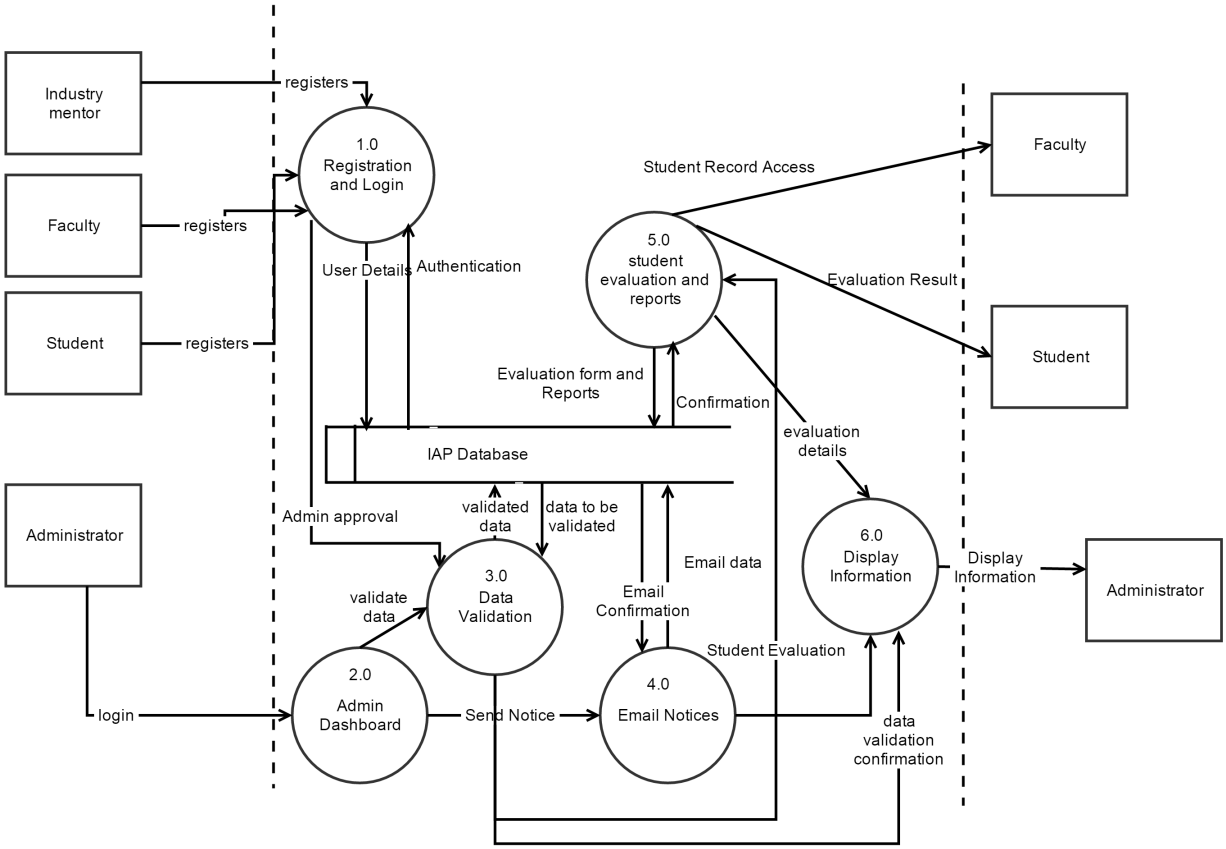

Figure 2: Level 1 DFD

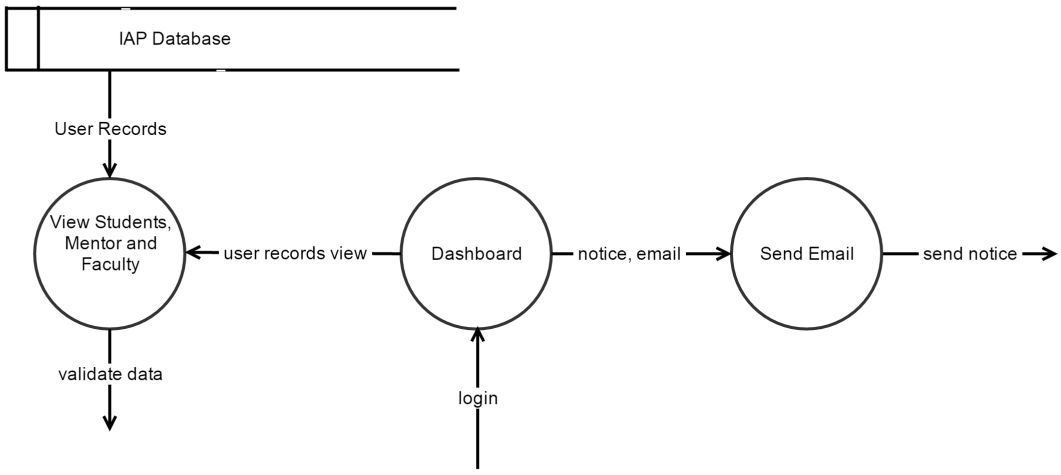

Figure 3: Level 2 DFD

7.3 UML Diagram

Class Diagram

This class diagram is describing the attributes and operations of a class and also the constraints imposed on the system.

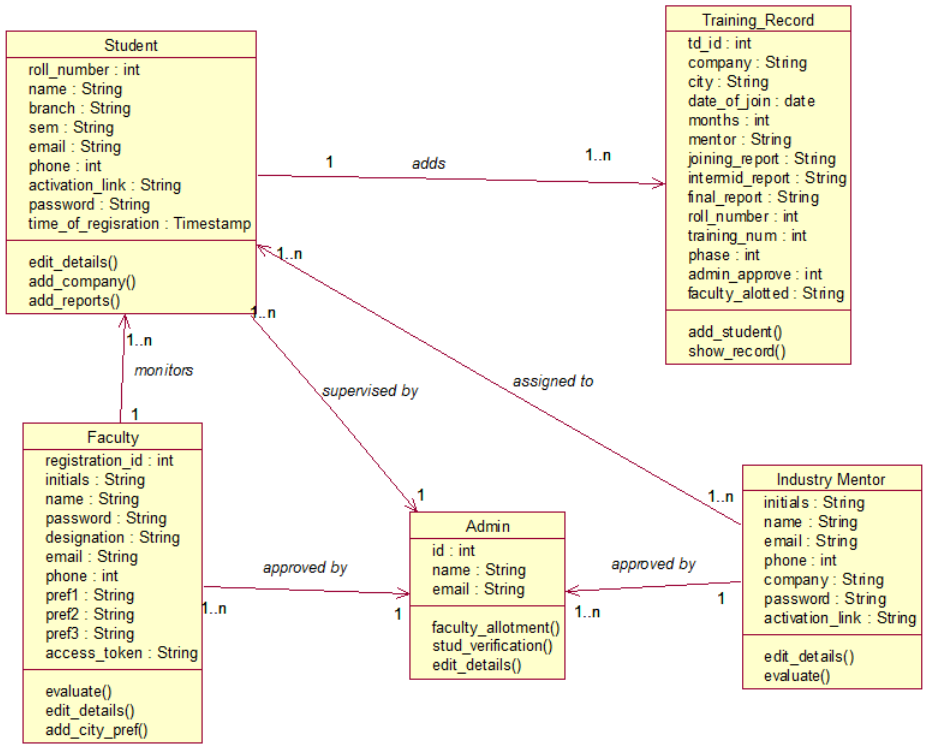

Figure 4: Class Diagram

Activity Diagram

It is a flow chart to represent the flow form one activity to another activity in the IAP.

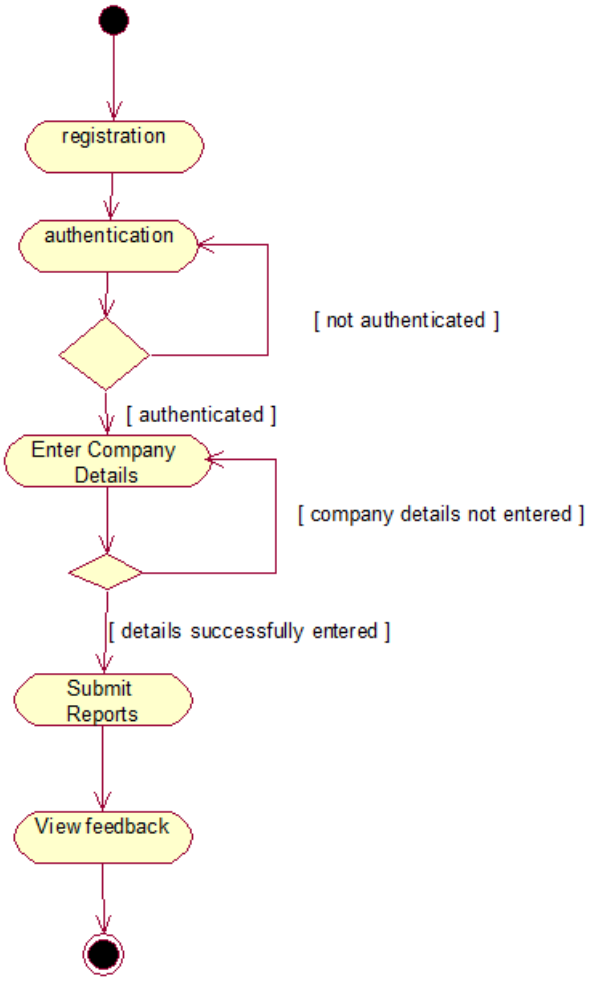

Figure 5: Activity Diagram For Student actions on the website

Swimlane Diagram

This swim lane is a visual element used in flowcharts, that visually distinguishes job sharing and responsibilities for sub-processes of IAP.

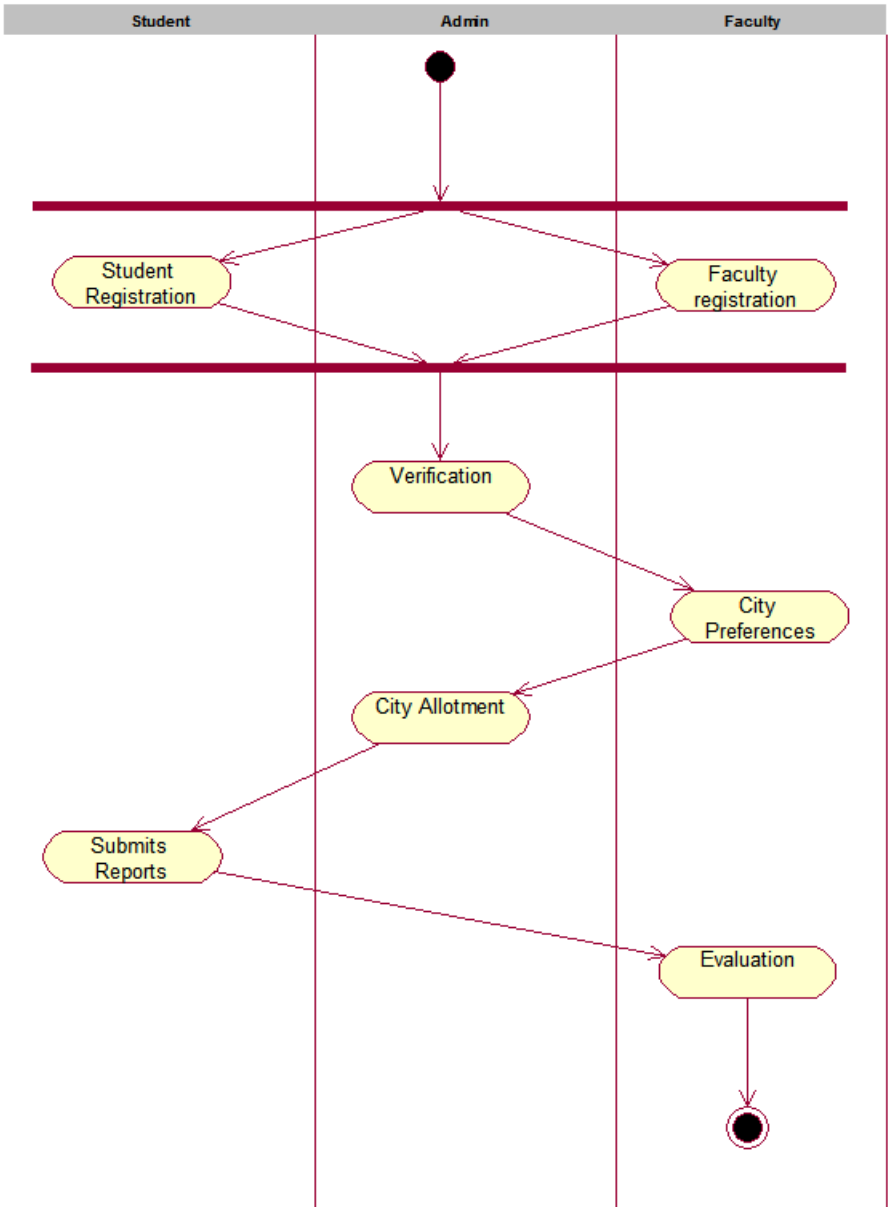

Figure 6: Swimlane Diagram depicting communication between the various users

Collaboration Diagram

This portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.

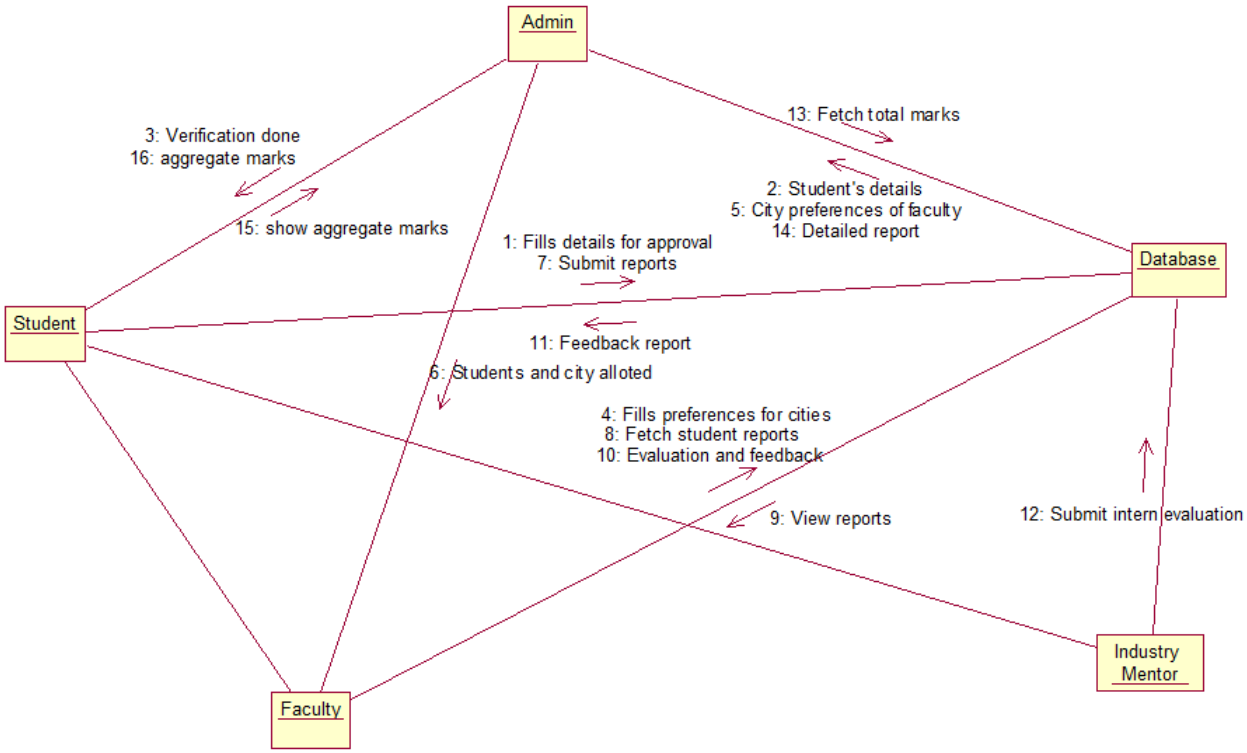

Figure 7: Collaboration Diagram

State Chart Diagram

State chart diagram describes the flow of control from one state to another state.

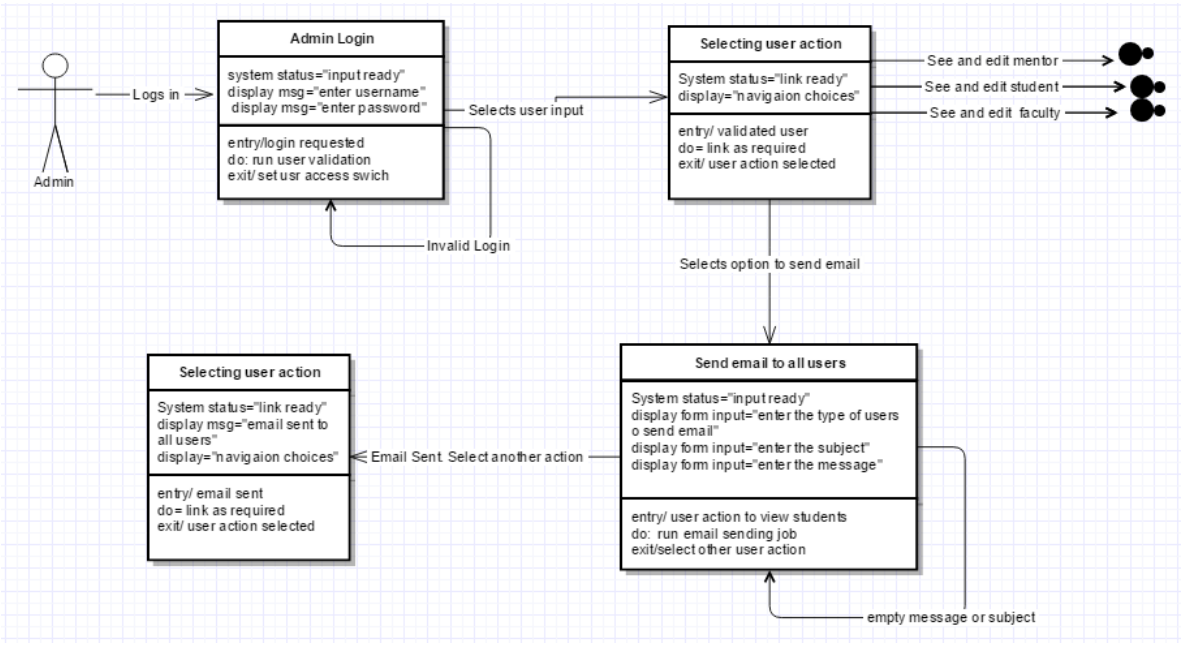

Figure 8: State Chart Diagram

Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment(distribution) of software artifacts to deployment targets.

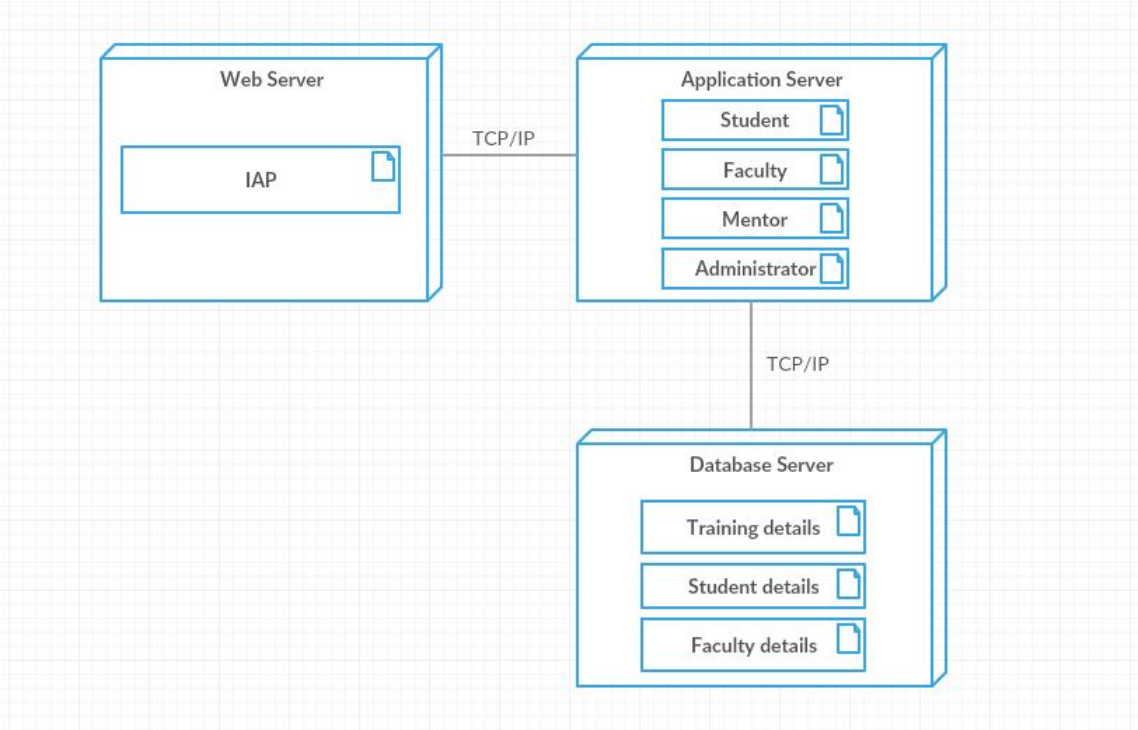

Figure 9: Deployment Diagram

Component Diagram

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc.

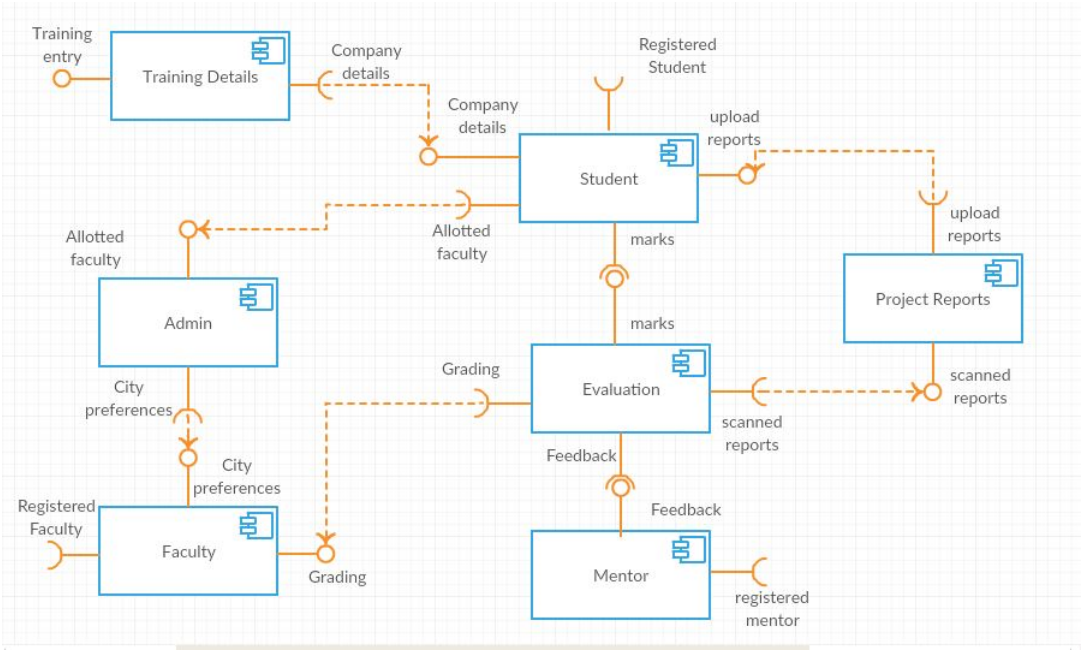

Figure 10: Component Diagram

Structure Chart

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them.

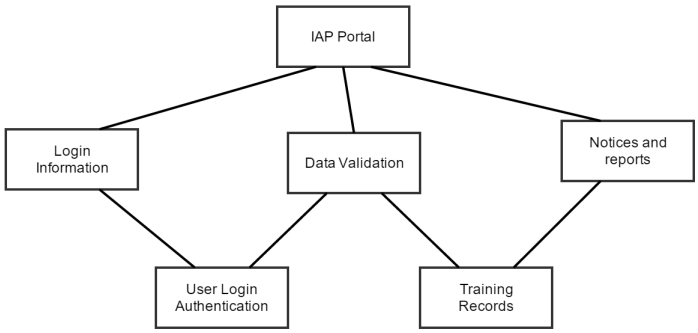

Figure 11: Structure Chart

7.4 ER Diagram

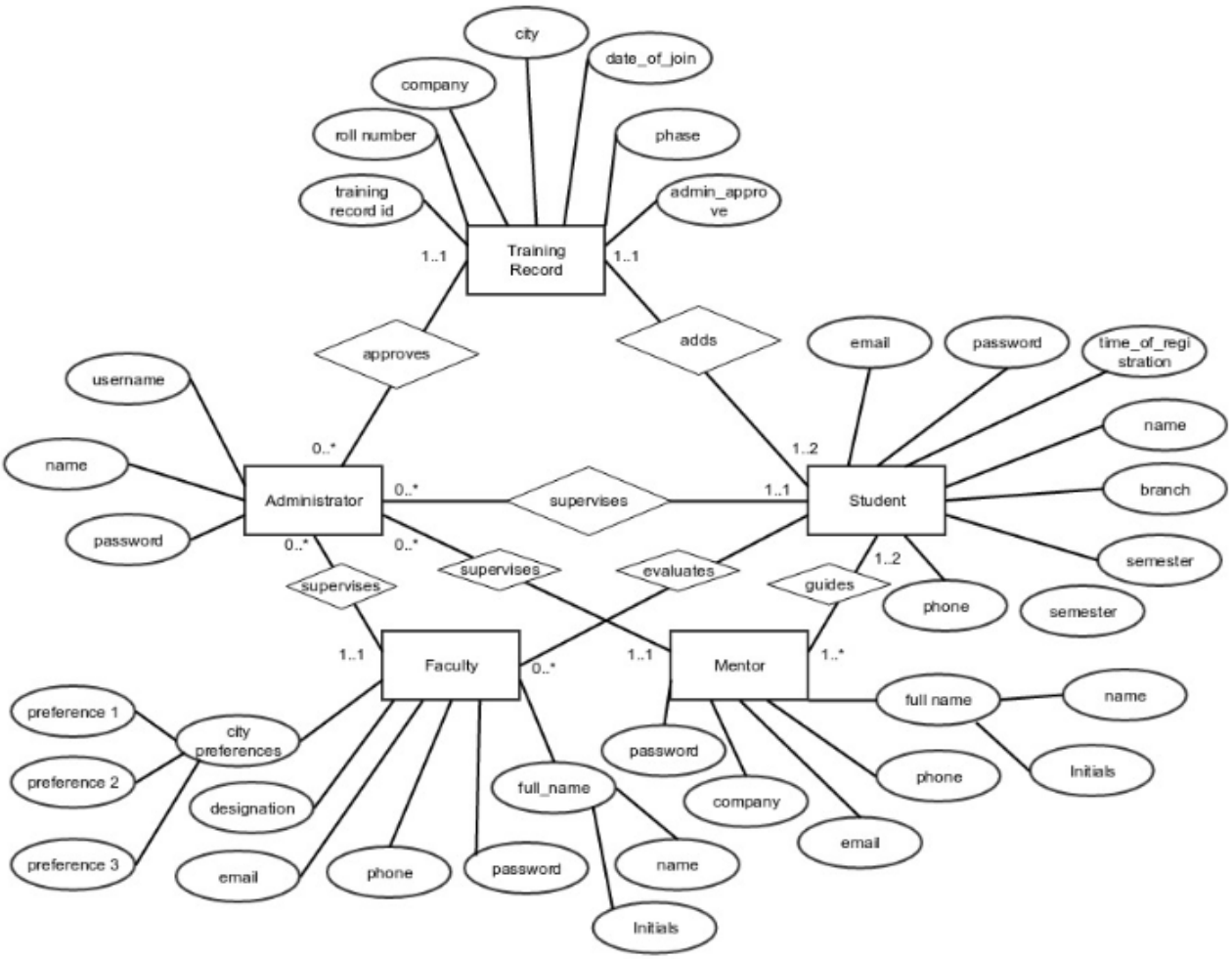

Figure 12: ER Diagram

8 Implementation

This web-app is implemented in PHP (2) as back-end and Bootstrap as frontend. MYSQL database is used in this project. Currently web-app is hosted on hostinger.com and e-mail client of hostinger is being used for sending email for donation request to donors. HTML and CSS (3) is used in designing of the web-app. On local computer website is designed and maintained on XAMPP server which uses Apache as a server and PHPMYADMIN for handling MYSQL queries. Text Editor used in the implementation is Sublime Editor (Free Version).

8.1 Snapshots of Application

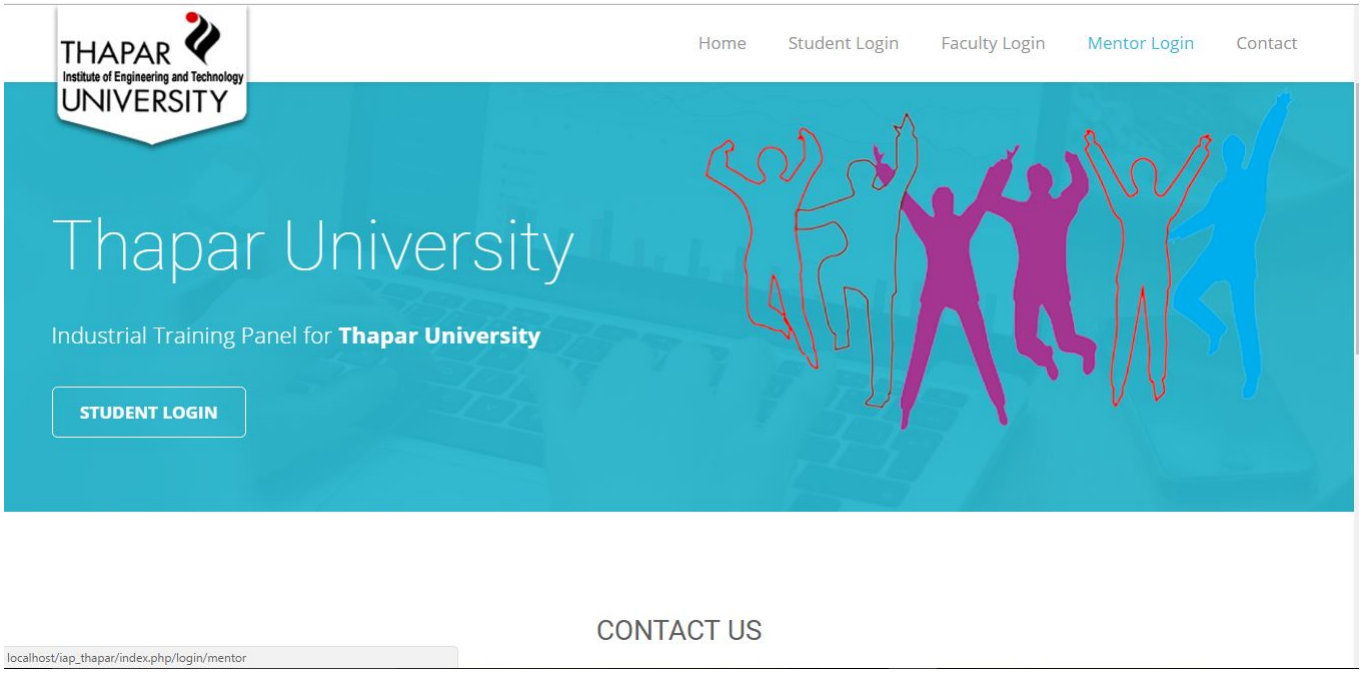

Figure 13: IAP Portal Home (Screenshot)

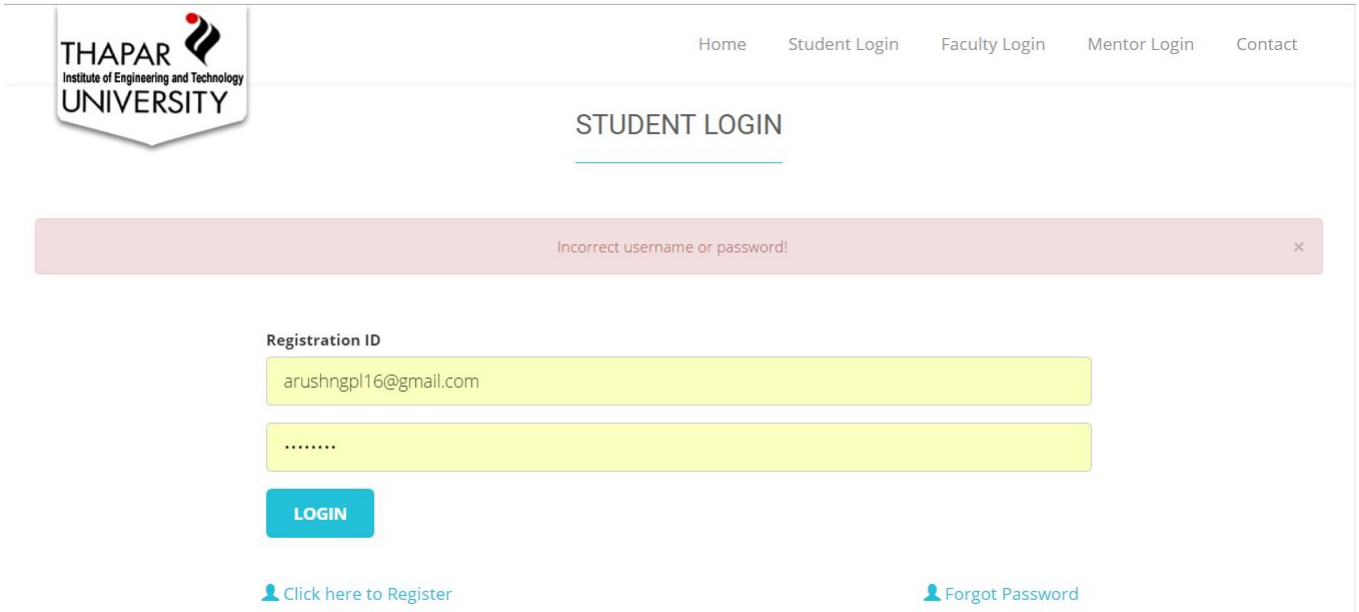

Figure 14: Student - Login Page (Screenshot)

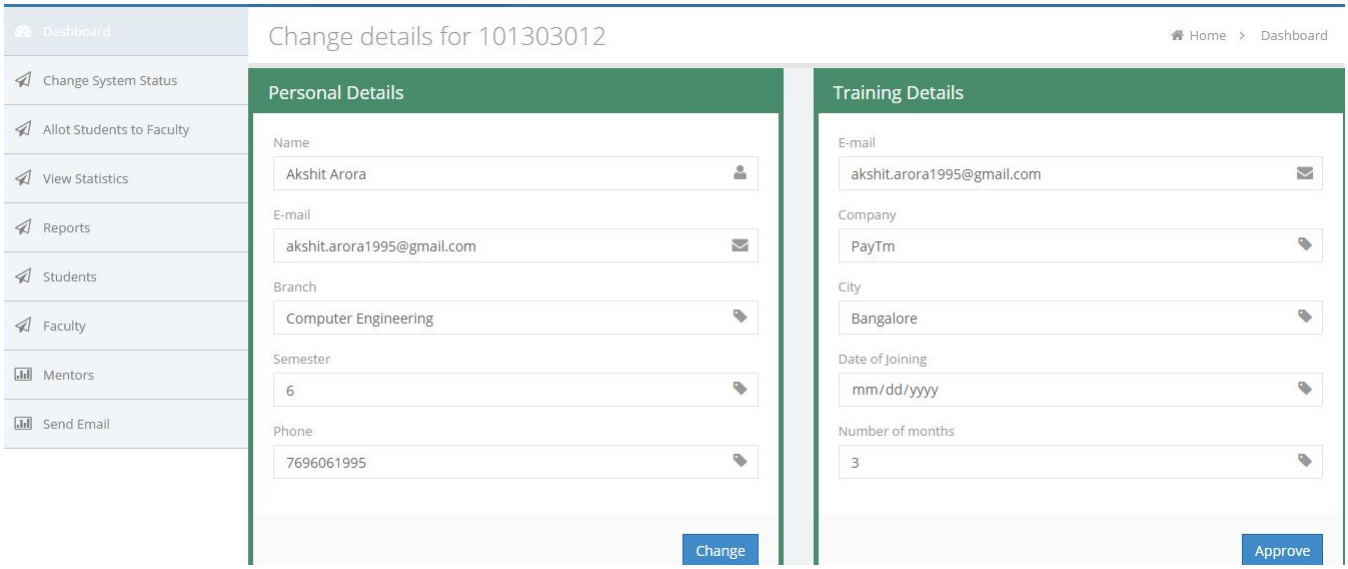

Figure 15: Admin - Edit and Approve Students (Screenshot)

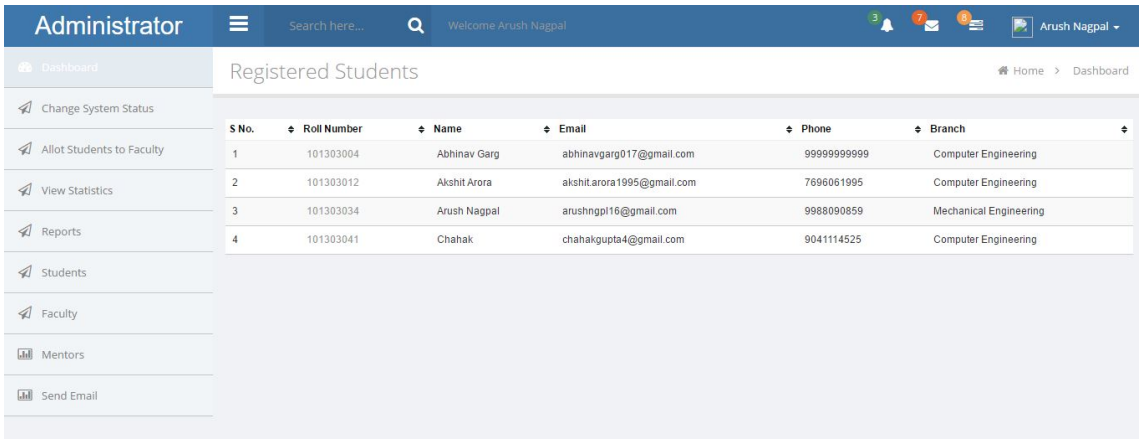

Figure 16: Admin - View all students (Screenshot)

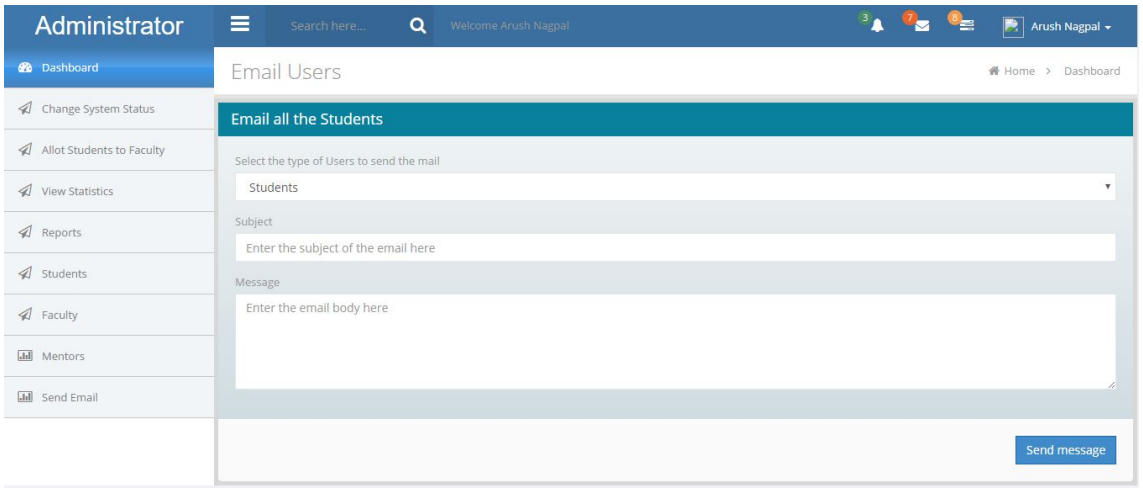

Figure 17: Admin - Sending notification to all the registered students (Screenshot)

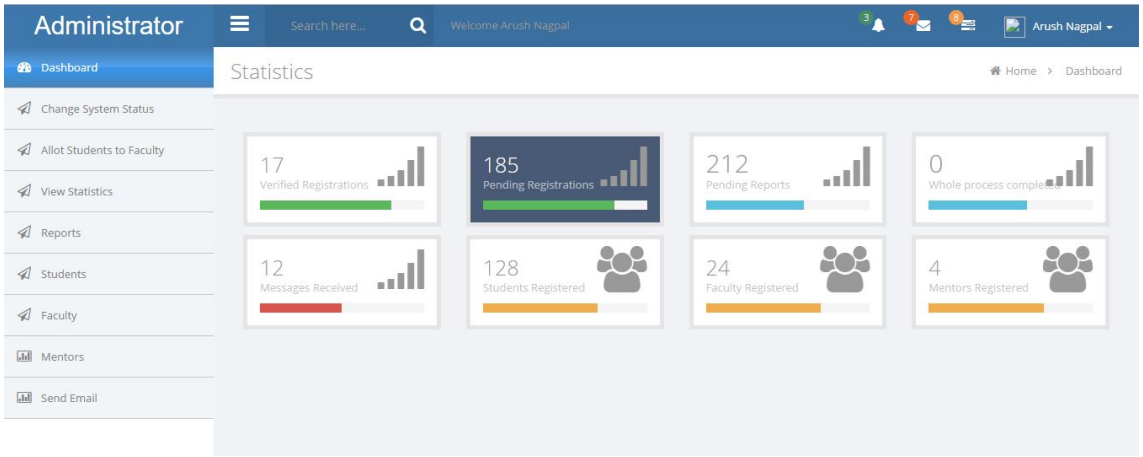

Figure 18: Admin - dashboard (Screenshot)

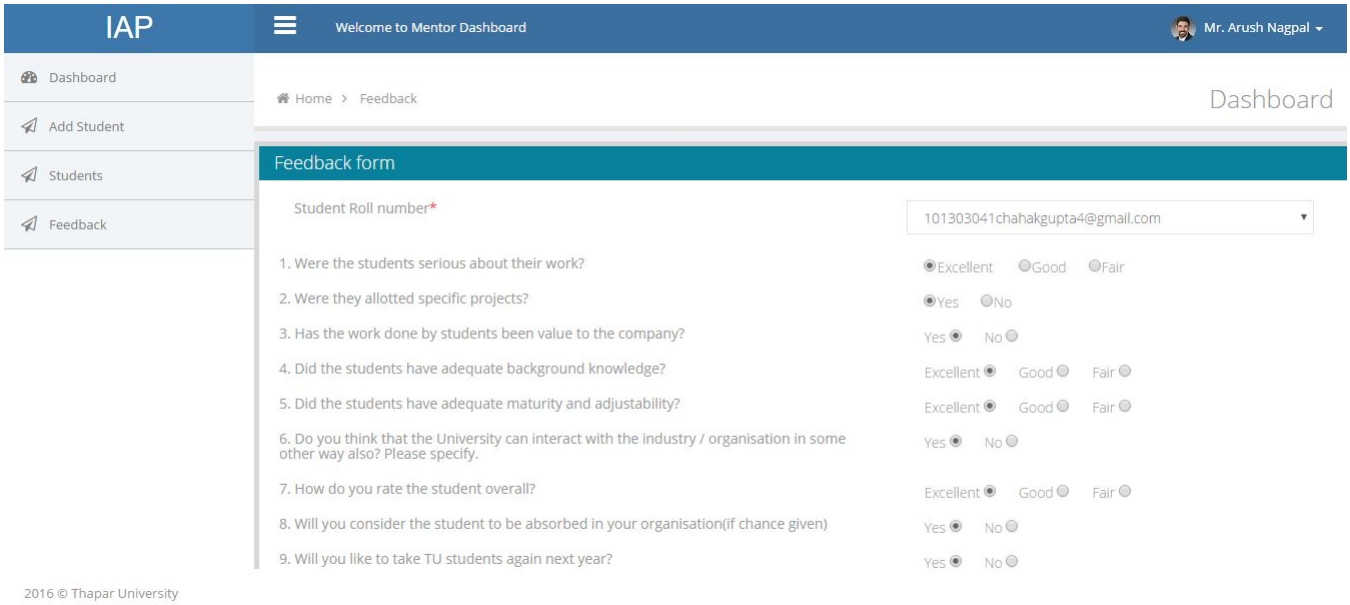

Figure 19: Mentor - Feedback form for student (Screenshot)

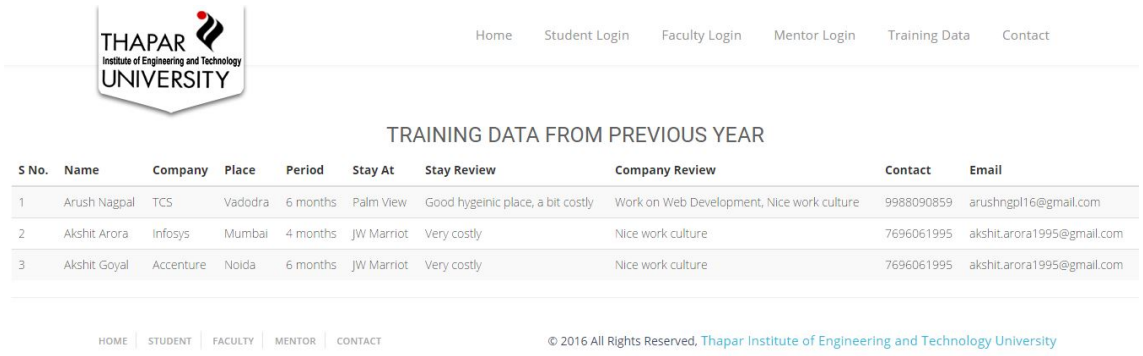

Figure 20: General Audience - Previous years' training data (Screenshot)

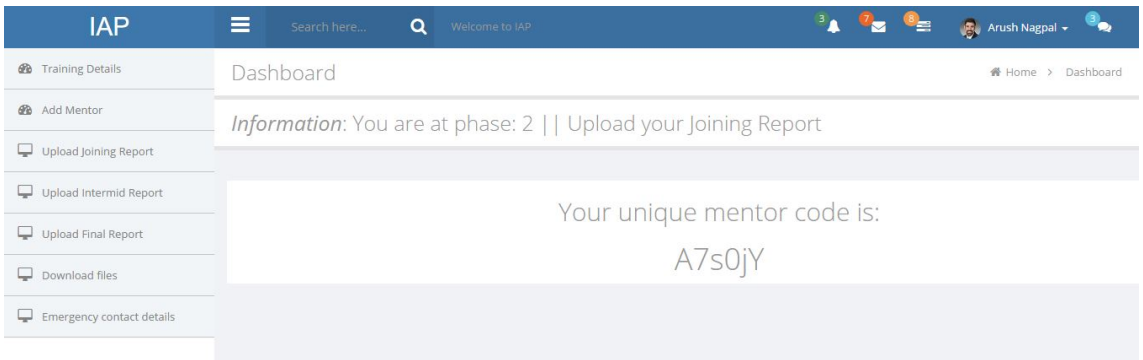

Figure 21: Student - Code for Linking Mentor (Screenshot)

9 Conclusion and Future Work

IAP for Mechanical can be extended to other departments as well by making necessary amendments in the modules in accordance with the requirements of the departments. Also, as the number of students are more in the junior years the software should be thought upon for scalability. Records of alumni and their placement and internship details along with their reviews can be made available. Every manual process going around us should be thought upon and considered for automation reducing the human work.

10 Appendices

10.1 Appendix A: Coding

1. **Module 1: Login module for Faculty**

```
public function faculty()
   {
      $data['title']="Login | Faculty";

      $this->load->model('Faculty_model');
      if($this->input->post())
      {
         $data['registration_id']=$this->input->post('registration');
         $data['password']=$this->input->post('pass');

         /*Now check for every error*/
         $data['error']=array();
         $this->load->helper('email');
         if($this->Default_model->isEmpty($data['registration_id']))
            array_push($data['error'], array('Please enter your
               registered email address!',0));
         if($this->Default_model->isEmpty($data['password']))
            array_push($data['error'], array('Please enter your
               password!',0));
         if(!valid_email($data['registration_id']))
            array_push($data['error'], array('Incorrect email!',0));
         if(isset($data['error'][0]))
         {
            $this->load->view('templates/front_header',$data);
            $this->load->view('templates/login/faculty',$data);
            $this->load->view('templates/front_footer',$data);
         }
         else
         {

            if($this->Faculty_model->auth($data))
            {
```

```
                $this->session->set_userdata('user_type', 'Faculty');
                $this->session->set_userdata('uid',
                    $data['registration_id']); //currently all Unique
                    IDentification are emails only
                $data_fetch = array();
                $data_fetch = $this->Faculty_model->details($data);
                $fname= $data_fetch['initials']." ".$data_fetch['fname'];
                    //fname = full name (initials + name)
                $this->session->set_userdata('full_name', $fname);
                redirect('faculty', 'refresh');
            }
            else
            {
                array_push($data['error'], array('Incorrect username or
                    password!',0));
                $this->load->view('templates/front_header',$data);
                $this->load->view('templates/login/faculty',$data);
                $this->load->view('templates/front_footer',$data);
            }
        }
    }
    else
    {

        $this->load->view('templates/front_header',$data);
        $this->load->view('templates/login/faculty',$data);
        $this->load->view('templates/front_footer',$data);
    }
}
```

2. **Module 2: Admin sending E-Mail**

```
public function send_email($page='send_email')
{
    $data['heading']="Email Users";
    if($this->input->post())
    {
        $data['usertype'] = $this->input->post('usertype');
        $data['subject'] = $this->input->post('subject');
```

```
            $data['message'] = $this->input->post('message');
            $emails=$this->Admin_model->getEmail($data['usertype']);
            $data['emails']=$emails;
            $res=true;
            $count=0;
            foreach($emails as $em)
            {
               if(!$this->Admin_model->send_mail($em['email'],$data['subject'],$data['messa
                  $res=false;
               else
                  $count++;
            }
            $data['error']=array();
            if($res)
               array_push($data['error'], array("Email sucessfully
                   sent",1));
            else
            {
               array_push($data['error'], array("There was an error sending
                   email. The email has been sent to $count candidates
                   instead of count($emails) candidates",0));
            }
            $this->load->view('admin/admin_header', $data);
              $this->load->view('admin/' . $page , $data);
              $this->load->view('admin/admin_footer');
         }
         else
         {
            $this->load->view('admin/admin_header', $data);
            $this->load->view('admin/' . $page , $data);
            $this->load->view('admin/admin_footer');
         }
      }
```

3. **Module 3: Student input training details**

```
function set_company_details($document){
      $company = $this->get_companies($document['student_email']);
      $data_user = $this->details($document['student_email']);
```

```
    $var = $data_user->roll_number;
       $data = array(
      'roll_number' => $var,
      'email' => $_SESSION['uid'],
      'company' => $document['name'],
      'city' => $document['city'],
      'date_of_join' => $document['doj'],
      'months' => $document['months'],
      'phase' => '0',
      'training_num'=>count($company)+1,
      'admin_approve' => '1'
      );

       $query1=$this->db->get_where('training_data', $data);

       if( $query1->num_rows()>0){

            return false;
       }
       else{
          if($this->db->insert('training_data', $data))
            return true;
          else
            return false;
       }
  }
```

10.2 Appendix B: Database Table Structure

Name	Type	Null	Default
t_id	int(11)	No	Auto-Increment
roll_number	bigint(20)	No	None
email	varchar(100)	No	None
company	varchar(100)	Yes	NULL
city	varchar(100)	Yes	NULL
date_of_join	date	No	None
months	int(11)	Yes	NULL
mentor	varchar(100)	Yes	NULL
joining_report	varchar(200)	Yes	NULL
intermediate_report	varchar(200)	Yes	NULL
final_report	varchar(200)	Yes	NULL
training_num	varchar(10)	Yes	NULL
phase	int(11)	No	0
admin_approve	int(2)	No	0
faculty_allotted	varchar(100)	Yes	NULL
feedback_done	int(2)	Yes	NULL

Table 1: Attributes for table: training_data

Name	Type	Null	Default
registration_id	varchar(200)	No	None
name	varchar(200)	No	None
password	varchar(200)	No	None

Table 2: Attributes for table: Administrator

Name	Type	Null	Default
id	bigint(20)	No	None
from_user	varchar(200)	No	None
to_user	varchar(200)	No	None
subject	varchar(300)	No	None
message	varchar(1000)	No	None
ip_address	varchar(100)	No	None
timestamp	timestamp	No	Current_timestamp

Table 3: Attributes for table: admin_messages

Name	Type	Null	Default
initials	varchar(10)	No	None
name	varchar(250)	No	None
phone	bigint(20)	No	None
email	varchar(200)	No	None
company	varchar(100)	No	None
password	varchar(250)	No	None
activation_link	varchar(250)	No	None

Table 4: Attributes for table: mentor

Name	Type	Null	Default
mentor	varchar(200)	No	None
roll_number	bigint(20)	No	None
email	varchar(200)	No	None
q1	varchar(4)	No	None
q2	varchar(4)	No	None
q3	varchar(4)	No	None
q4	varchar(4)	No	None
q5	varchar(4)	No	None
q6	varchar(4)	No	None
q7	varchar(4)	No	None
q8	varchar(4)	No	None
q9	varchar(4)	No	None

Table 5: Attributes for table: mentor_feedback

Bibliography

[1] http://getbootstrap.com/about/

[2] https://php.net/manual/en/index.php

[3] https://www.codecademy.com/learn/web

[4] https://dev.mysql.com/doc

[5] http://www.codeigniter.com/