****

**Pakistan Air Force**

**Karachi Institute of Economics & Technology**

**ACADEMIC REVIEWS**

**Submitted by:   
Naifish Nizar Ali (57036)**

**Muhammad Faiz (55903)**

**Supervisor: Amanullah**

College of Computing & Information Sciences

A final year project report presented to the College of Computing & Information Sciences,  
Pakistan Air Force – Karachi Institute of Economics & Technology in partial fulfillment of the requirements of the degree of Bachelor of Science

**2016**

Contents

[Abstract](#_Toc401811362)

[Acknowledgements](#_Toc401811363)

[Acronyms](#_Toc401811364)

[List of Figures](#_Toc401811365)

[1. Introduction 1](#_Toc401811366)

[1.1. Motivations](#_Toc401811367)

[1.2. Objectives and Scope](#_Toc401811368)

[2. Literature Review 2](#_Toc401811369)

[2.1. Universities and faculties ratings](#_Toc401811370)

[3. Domain and Methodology 3-9](#_Toc401811371)

[3.1. Dataset](#_Toc401811372)

[1.1.1 Data Sources / Information Gathering](#_Toc401811373)

[3.2. Model / Development Algorithms](#_Toc401811374)

[4. Design and Implementation 10-22](#_Toc401811375)

[4.1. Design](#_Toc401811376)

[USE CASES](#_Toc401811377)  
Use Case Glossary

[ACTIVITY DIAGRAMS](#_Toc401811378)

[SEQUENCE DIAGRAMS](#_Toc401811379)

[5. Result 23](#_Toc401811381)

[6. Conclusion 24](#_Toc401811381)

[References 25](#_Toc401811384)

[Appendix A – Development 26-30](#_Toc401811385)

[Environment 26-30](#_Toc401811386)

[ Operating Systems(s)](#_Toc401811387)

[o Windows 7](#_Toc401811388)

[ DataBase](#_Toc401811389)

[ Programming Language(s)](#_Toc401811390)

[o PHP](#_Toc401811391)

[o HTML5](#_Toc401811392)

[o CSS](#_Toc401811393)

[o JAVASCRIPT](#_Toc401811394)

[o JQUERY](#_Toc401811395)

[o AJAX](#_Toc401811396)

[ Programming Paradigm](#_Toc401811397)

[o Object-oriented programming](#_Toc401811398)

[ Communication and Leadership](#_Toc401811399)

[o Democratic Decentralized](#_Toc401811400)

[Appendix B – Tools and Softwares 31-32](#_Toc401811401)

[ Xampp](#_Toc401811402)

[ Mozilla Firefox](#_Toc401811403)

[ Sublime Text](#_Toc401811404)

[Appendix C – Code 33-55](#_Toc401811405)

[addProf.php](#_Toc401811406)

[addUni.php](#_Toc401811407)

[rateProf.php](#_Toc401811408)

[rateUni.css](#_Toc401811409)

[flagTeacherRating.php](#_Toc401811410)

[shareProfStats.php](#_Toc401811411)

[fetchData.php](#_Toc401811412)

[Appendix D – Snapshots of project 56-63](#_Toc401811405)

[Search Professors](#_Toc401811406)

[Search Universities](#_Toc401811406)

[Add Professor](#_Toc401811406)

[Add University](#_Toc401811406)

[Professor's Search Result](#_Toc401811406)

[University’s Search Result  
Show Professor’s Rating  
Show University’s Rating  
Rate Professor  
Rate University  
Submit Corrections  
Report Rating](#_Toc401811406)

[Appendix E – Team Members 64](#_Toc401811422)

[Annexure II – Software Planning & Costing 65](#_Toc401811425)

[Annexure III – Test Case 66-69](#_Toc401811426)

[Test Case 1.1: User Login to the website](#_Toc401811427)

[Test Case 1.2: rate a professor](#_Toc401811428)

[Test Case 1.3: rate a university](#_Toc401811429)

[Test Case 1.4: flag student rating](#_Toc401811430)

[Test Case 1.5: share professor or university stats](#_Toc401811431)

[Test Case 1.6: Correct university info](#_Toc401811432)

[Test Case 1.7: contact us](#_Toc401811433)

[End](#_Toc401811436)

# Abstract

This will be an open Website to all who are in search of reviews and the feedback of teachers and the universities, the core features will be the review of teaching faculties who are teaching in different educational institution given by respective students. This will help new students to check their new upcoming faculties and to analyze their teaching methodologies. Later on they can decide to take that course or not, in addition Student can rate different aspects of teaching pattern and interaction with student, and overall rating would be calculated on that profile. Second feature is giving reviews to particular educational institution similar to teaching faculty.

# 

# Acknowledgements

We would like to acknowledge the great support of our instructor for helping us throughout making of our final year project. We would also like to show the deepest appreciation to the committee members who showed their positive attitude towards the project, and being a source of guidance, and for explaining all the queries we came across while preparing this project. Without their support or guidance, it would not be possible. We are thankful to them for their constant help while developing our project

Naifish Ali

Muhammad Faiz

# Acronyms

**Alt** Alternative

**AJAX** Asynchronous JavaScript and XML

**B** Bold

**BR** Break

**CSS** Cascading Style Sheets

**Div** Division

**Dir** Directory

**DOM** Document Object Model

**GUI** Graphical User Interface

**h** Header

**H6** Header 6

**HR** Horizontal Ruler

**HREF** Hypertext Reference

**HTTP** Hypertext Transfer Protocol

**I** Italic

**IMG** Image

**LI** List Item

**P** Paragraph

**PHP** Hypertext Preprocessor

**SEO** Search Engine Optimization

**src** Source

**TD** Table Data

**TH** Table Header

**TR** Table Row

**U** Underline

**UL** Unsorted List

**URL** Uniform Resource Locator

**Var** Variable

**WWW** World Wide Web

# 

# List of Figures

[Figure 4.1.1 Entity Relationship Diagram 10](#_Toc401812080)

[Figure 4.1.2 Flow Chart Diagram 11](#_Toc401812081)

[Figure 4.1.3 Use Case Diagrams 12](#_Toc401812083)

[Figure 4.1.4 Class Diagram 13](#_Toc401812084)

[Figure 4.1.5 Activity Diagrams 14](#_Toc401812085)-15

[Figure 4.1.6 Sequence Diagrams 16-17](#_Toc401812087)

# Introduction

Academic review is the web application through which Students can find the review and the feedbacks on different faculties and universities. This application will help students to rate different faculties and the universities by visiting this web application students who have recently completed their inter level education can easily find out the best universities throughout the world.

## Motivations

By developing this website of Academic Reviews we believe that our project will help all those students and their parents who are in searching of the best universities for their children because as we can see that After completing intermediate students face problems in choosing a university from where they can get best education and even the students of same university face problems after completion of semester that what course they should take and from which teacher so our website will help them to take a right decision

## Objectives and Scope

By developing this website of Academic Reviews we believe that our project will help all those students and their parents who are in searching of the best universities for their children from where they can get best education and the students of same university can decide that in which course they should enroll and from which teacher. So our website will help them to take a right decision.

# Literature Review

In Academic Review few problems we encountered was that how we find out that the reviews which are given by the students are valid or not for a faculty or university and does such type of university and faculty really exist for that we have been through the website like uloop.com and myedu.com and it help us. The problems which we found in their website are less user friendly and this type of website do not require even an account to rate faculties and universities so any unauthorized person can give wrong rating. So in our project we let other students that they can report any student’s rating on any faculty or university and our website will also require a student account in order to rate university or faculty.

## Universities and faculties ratings

We have performed deep researched on the websites who work on faculties and universities ratings but we did not find a solution to validate the student feedback then we decided to let other student to validate a particular feedback. If the feedback is not valid then they can report that feedback and the number of reports will help admin to take action on that feedback. One more thing is might be possible that the ratings are not according to feedback so for this we will fetch comments and posts regarding faculties and universities from the Facebook and rate accordingly.

# Domain and Methodology

## Dataset

### Data Sources / Information Gathering

In our application Academic Review our data is Students, faculties, universities, countries, states and cities information and the feedback of faculties and universities.

## Model / Development Algorithms

**PROBLEM**:

To identify that the user who is registering him/her self as a professor is actually a professor or not.

**SOLUTION**:

We make a script that login a user through Facebook and using Facebook graph API we grab the user work history. And at the registration time we check that user is actually a professor or not and to which university the professor belongs.

**Algorithm**:

<?php

if ($ProfileKeys[$i]=="work") {

# code...

$Work\_KEYS=array\_keys($profile[$ProfileKeys[$i]]);

foreach ($profile[$ProfileKeys[$i]] as $K => $V) {

# code...

//echo count($profile[$ProfileKeys[$i]]);

//$multi\_wrk=array\_keys($profile[$ProfileKeys[$i]][$Work\_KEYS[$wrkNumber]]);

$employerArr[$wrkNumber]="undefined";

$positionArr[$wrkNumber]="undefined";

$startDateArr[$wrkNumber]="undefined";

$endDateArr[$wrkNumber]="undefined";

foreach ($profile[$ProfileKeys[$i]][$Work\_KEYS[$wrkNumber]] as $KK => $VV) {

# code...

if ($KK=="employer") {

# code...

foreach ($profile[$ProfileKeys[$i]][$Work\_KEYS[$wrkNumber]][$KK] as $key => $value) {

# code...

if($key=="name"){ $employerArr[$wrkNumber]=$value; }

}

}

if ($KK=="position") {

# code...

foreach ($profile[$ProfileKeys[$i]][$Work\_KEYS[$wrkNumber]][$KK] as $key => $value) {

# code...

if($key=="name"){ $positionArr[$wrkNumber]=$value; }

}

}

if ($KK=="start\_date") {

# code...

$startDateArr[$wrkNumber]=$VV;

}

if ($KK=="end\_date") {

# code...

$endDateArr[$wrkNumber]=$VV;

}

}

//print\_r($multi\_edu\_wrk);

$wrkNumber++;

}

}

//Registration Validation

if ($\_SESSION["lecturerPlaces"]!="") {

# code...

$userIsProfessor=true;

$lecturerAt=explode("\*\*\*", $\_SESSION["lecturerPlaces"]);

}else{/\*echo $showMessage="You do not have any rights to add any professor.";\*/}

}

if ($prof\_exist!=true) {

# code...

if ($\_SESSION["lecturerPlaces"]!="") {

$stu = new Student();

$stu->addProf($firstName,$middleName,$lastName,$university,$department,$directory,$country,$email,$start\_year,$end\_year,$contact,$av\_status);

}else{

/\*echo "<script>if (window.confirm('You do not have any rights to add any professor')){document.location = 'index.php'}else{document.location = 'index.php'}</script>";\*/

header('location:index.php#invalidProf');

}

}

?>

**PROBLEM:**

To check the user who is rating any university is really the student of that university or not.

**SOLUTION:**

We grab the user education history using Facebook graph API and validate that if user is actually belonging to the university or not.

**Algorithm:**

if ($ProfileKeys[$i]=="education") {

# code...

$isStudent=true;

$Edu\_KEYS=array\_keys($profile[$ProfileKeys[$i]]);

foreach ($profile[$ProfileKeys[$i]] as $K => $V) {

# code...

//echo count($profile[$ProfileKeys[$i]]);

//$multi\_edu=array\_keys($profile[$ProfileKeys[$i]][$Edu\_KEYS[$eduNumber]]);

$schoolArr[$eduNumber]="undefined";

$yearArr[$eduNumber]="undefined";

$typeArr[$eduNumber]="undefined";

foreach ($profile[$ProfileKeys[$i]][$Edu\_KEYS[$eduNumber]] as $KK => $VV) {

# code...

if ($KK=="school") {

# code...

foreach ($profile[$ProfileKeys[$i]][$Edu\_KEYS[$eduNumber]][$KK] as $key => $value) {

# code...

if($key=="name"){ $schoolArr[$eduNumber]=$value; }

}

}

if ($KK=="year") {

# code...

foreach ($profile[$ProfileKeys[$i]][$Edu\_KEYS[$eduNumber]][$KK] as $key => $value) {

# code...

if($key=="name"){ $yearArr[$eduNumber]=$value; }

}

}

if ($KK=="type") {

# code...

$typeArr[$eduNumber]=$VV;

}

}

//print\_r($multi\_edu\_wrk);

$eduNumber++;

}

}

//validating user

if ($\_SESSION["educationPlaces"]!="") {

# code...

$userIsStudent=true;

$studentAt=explode("\*\*\*", $\_SESSION["educationPlaces"]);

}

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($\_GET['univName']),strtolower($studentAt[$i]),$nameMchTemp);

if ($nameMchTemp>$nameMch) {

# code...

$nameMch=$nameMchTemp;

}

}

//echo $nameMch."<br>";

/\*matching the university nick name with the universitie's name of student attended\*/

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($uniNickName),strtolower($studentAt[$i]),$nickNameMchTemp);

if ($nickNameMchTemp>$nickNameMch) {

# code...

$nickNameMch=$nickNameMchTemp;

}

}

//echo $nickNameMch."<br>";

if ($nameMch>80 || $nickNameMch>80) {

# code...

$belongToUni=true;

}

**PROBLEM:**

To check the user who is rating any Professor is really belongs to that professor or not.

**SOLUTION:** We grab the user education history using Facebook graph API and compare the professor teaching period to that university and student studies period to that university and validate that if user is actually belonging to that Professor or not .

**Algorithm:**

//$start\_date=$\_POST['startDate'];

if ($\_POST['startDate']!=null) {

# code...

$start\_date=$\_POST['startDate'];

}

else{

$start\_date=null;

}

//$start\_date=$\_POST['startDate'];

if ($\_POST['endDate']!=null) {

# code...

$end\_date=$\_POST['endDate'];

$stuEndYear=$\_POST['endDate'];

}

else{

$end\_date=null;

$stuEndYear=date("Y");

}

$av\_status="pending";

$start\_date."-".$stuEndYear."<br>";

$profStartYear."-".$profEndYear."<br>";

/\*if user belongs to professor or not\*/

if ($profEndYear<$start\_date || $stuEndYear<$profStartYear) {

# code...

$belongToProf=false;

}

**PROBLEM:**

Some confidential info of professor was visible to the user through URL.

**SOLUTION:** To hide that confidential data we write a algorithms to encrypt and decrypt that data.

**Algorithm:**

<?php

function encrypt\_url($string) {

$key = "KIET\_1984"; //key to encrypt and decrypts.

$result = '';

$test = "";

for($i=0; $i<strlen($string); $i++) {

$char = substr($string, $i, 1);

$keychar = substr($key, ($i % strlen($key))-1, 1);

$char = chr(ord($char)+ord($keychar));

$test[$char]= ord($char)+ord($keychar);

$result.=$char;

}

return urlencode(base64\_encode($result));

}

function decrypt\_url($string) {

$key = "KIET\_1984"; //key to encrypt and decrypts.

$result = '';

$string = base64\_decode(urldecode($string));

for($i=0; $i<strlen($string); $i++) {

$char = substr($string, $i, 1);

$keychar = substr($key, ($i % strlen($key))-1, 1);

$char = chr(ord($char)-ord($keychar));

$result.=$char;

}

return $result;

}

function parse\_Myurl(){

$actual\_link = "$\_SERVER[REQUEST\_URI]";

$whatIWant = substr($actual\_link, strpos($actual\_link, "?") + 1);

$get\_string=decrypt\_url($whatIWant);

parse\_str($get\_string, $get\_array);

foreach ($get\_array as $key => $value) {

# code...

$\_GET[$key]=$value;

}

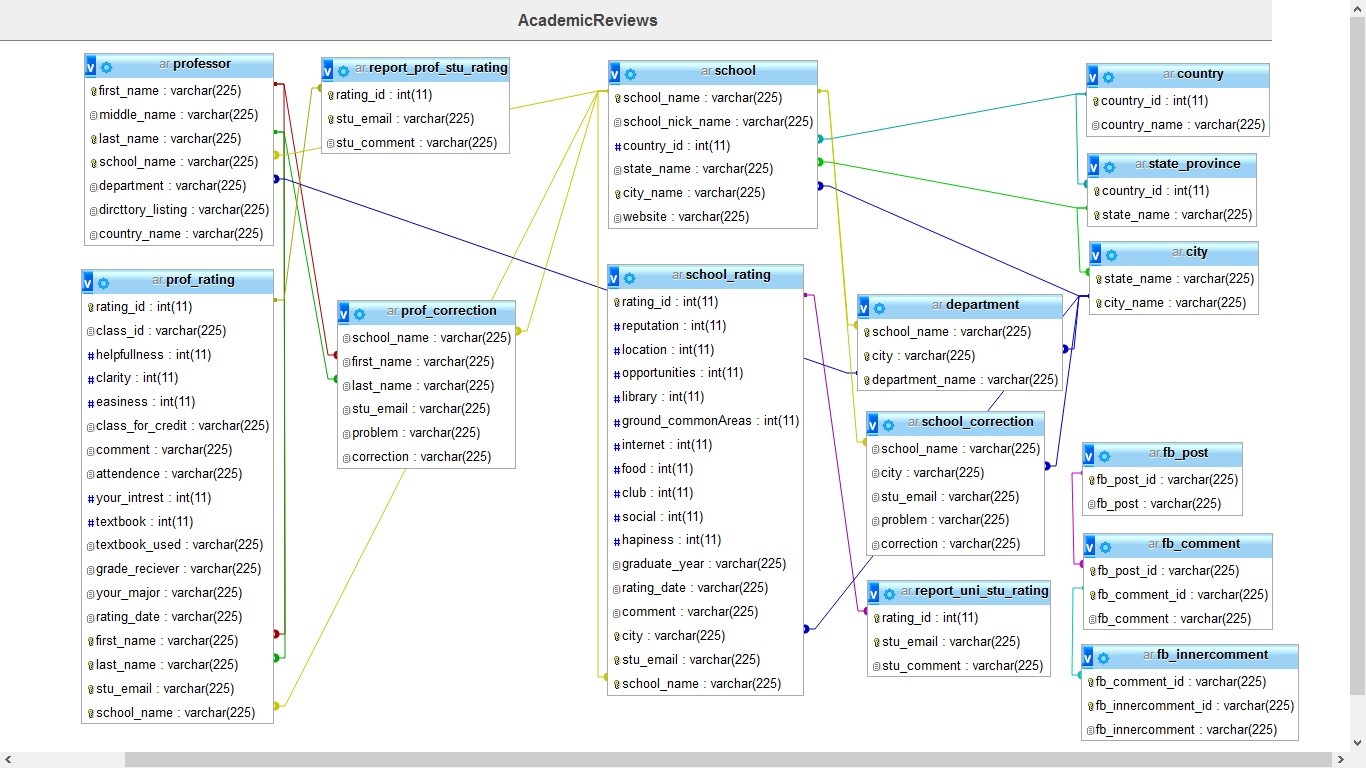
}

?>

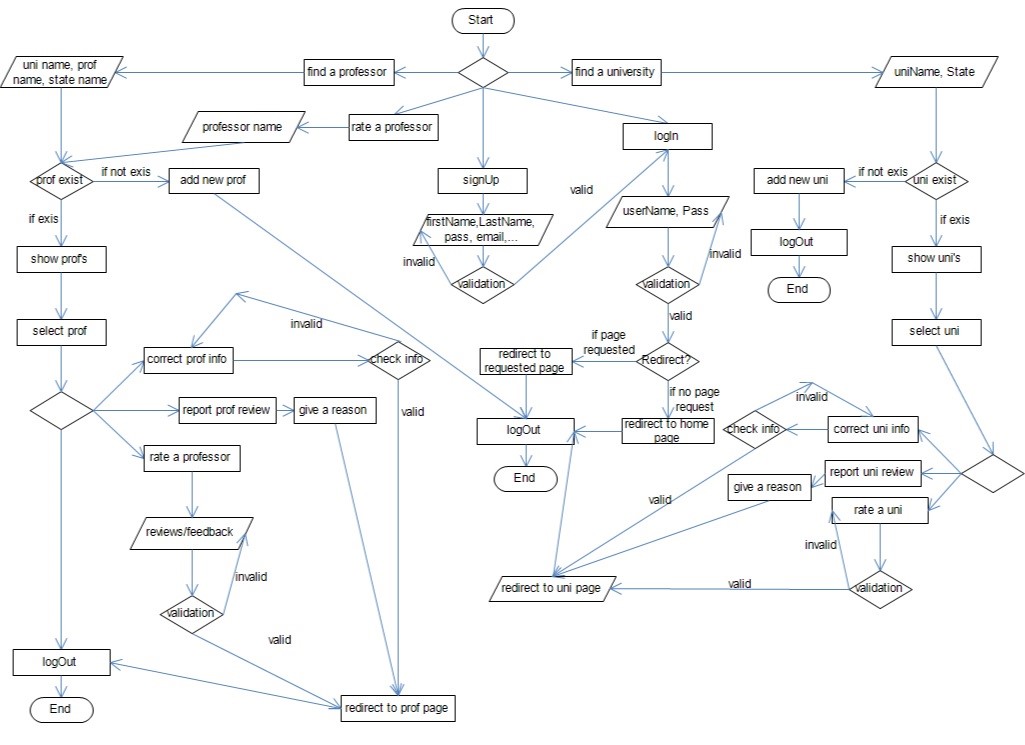
# Design and Implementation

## Design

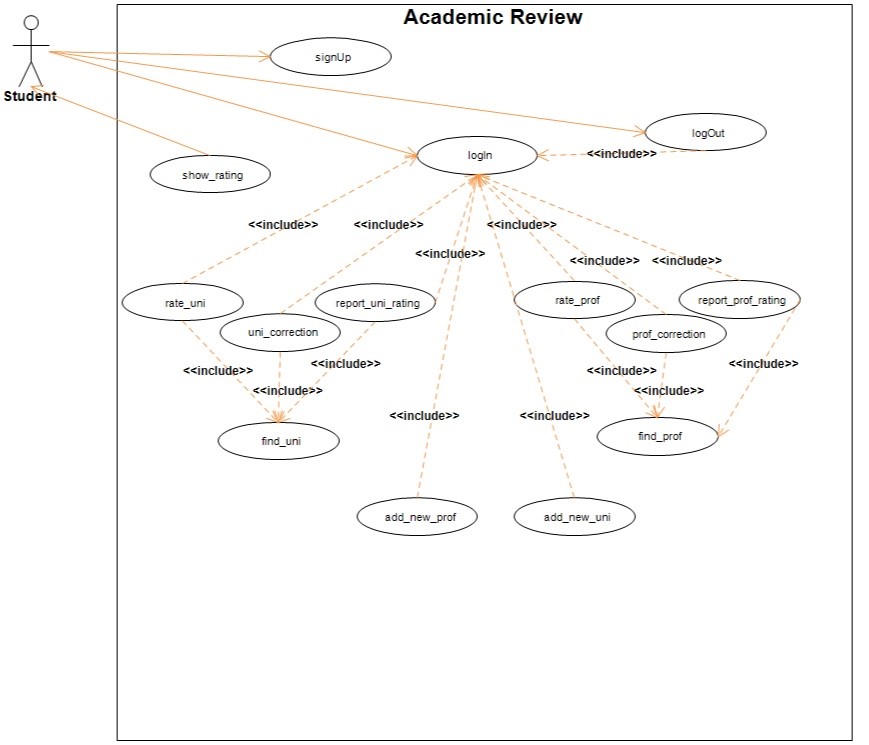
##### Figure 4.1.1 Entity Relationship Diagram:

****

##### Figure 4.1.2 Flow Chart Diagram:



##### Figure 4.1.3 Use Case Diagram:

****

##### 

## Use Case Glossary

|  |  |
| --- | --- |
| Use Case ID: | **101** |
| Use Case Name: | **Login** |
| Actors: | Student/Professor |
| Priority | High |
| Description: | In order to perform some action on website then user will have to login through Facebook. |
| Preconditions: | None |
| Post conditions: | User will have to give access to the share permissions on Facebook including education and work history. |
| Extends | None |
| Uses | None |
| Normal Flow | User will be redirected to the requested page. |
| Alternate Flow | User will be redirected to the Home page. |
| Assumptions: | User must have Facebook account. |

|  |  |
| --- | --- |
| Use Case ID: | **102** |
| Use Case Name: | **Search Professor** |
| Actors: | Student/Professor |
| Priority | Normal |
| Description: | User will provide professor and university name or state name of university name in order to search any professor. |
| Preconditions: | None. |
| Post conditions: | User will select an appropriate professor from the search results. |
| Extends | None |
| Uses | None |
| Normal Flow | All the matching professor will be shown to the user. |
| Alternate Flow | If there is no matching result, then it will be asked by the professor to register him/herself. |
| Assumptions: | None |

|  |  |
| --- | --- |
| Use Case ID: | **103** |
| Use Case Name: | **Search University** |
| Actors: | Student/Professor |
| Priority | Normal |
| Description: | User will provide university name or state name in order to search any University. |
| Preconditions: | None. |
| Post conditions: | User will select an appropriate University from the search results. |
| Extends | None |
| Uses | None |
| Normal Flow | All the matching universities will be shown to the user. |
| Alternate Flow | If there is no matching result, then it will be asked by the professor to register his/her university. |
| Assumptions: | None |

|  |  |
| --- | --- |
| Use Case ID: | **104** |
| Use Case Name: | **Rate Professor** |
| Actors: | Student |
| Priority | Normal |
| Description: | Student will have to provide the required info in order to rate the professor. |
| Preconditions: | Student must be login. |
| Post conditions: | None. |
| Extends | Search Professor |
| Uses | Login |
| Normal Flow | Student will provide the required info and the redirected to the home page with the success message. |
| Alternate Flow | Student might redirect to the same page if student is not applicable or missed any required field. |
| Assumptions: | Professor must be registered on the website. |

|  |  |
| --- | --- |
| Use Case ID: | **105** |
| Use Case Name: | **Rate University** |
| Actors: | Student |
| Priority | Normal |
| Description: | Student will have to provide the required info in order to rate the University. |
| Preconditions: | Student must be login. |
| Post conditions: | None. |
| Extends | Search University |
| Uses | Login |
| Normal Flow | Student will provide the required info and the redirected to the home page with the success message. |
| Alternate Flow | Student might redirect to the same page if student is not applicable or missed any required field. |
| Assumptions: | University must be registered on the website. |

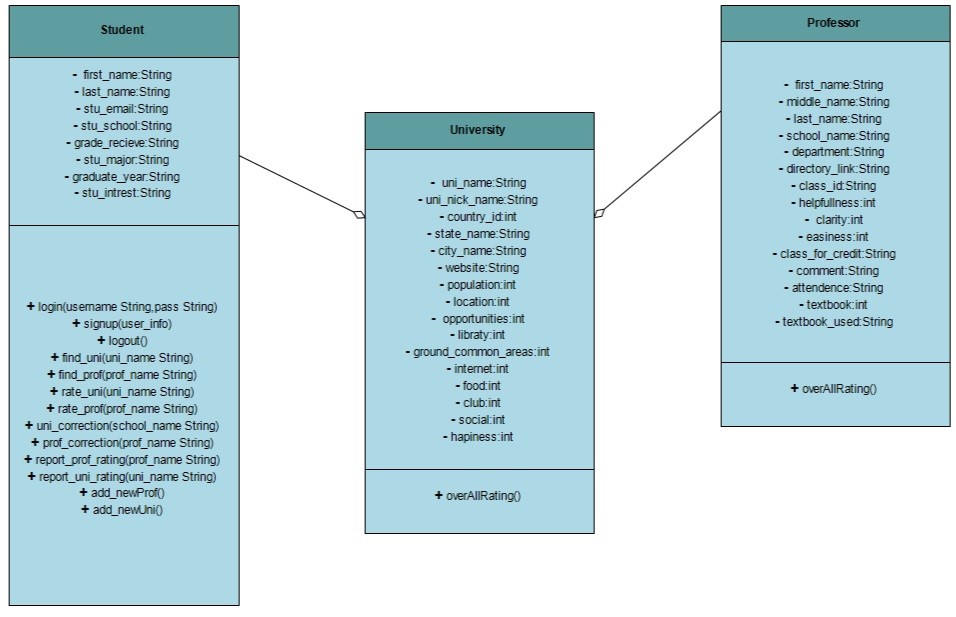
|  |  |
| --- | --- |
| Use Case ID: | **106** |
| Use Case Name: | **Add Professor** |
| Actors: | Professor |
| Priority | Normal |
| Description: | Professor will provide the required info in order to register his/herself to the website. |
| Preconditions: | Professor must be login. |
| Post conditions: | None. |
| Extends | None |
| Uses | Login |
| Normal Flow | Professor will provide the required info and the redirected to the home page with the success message. |
| Alternate Flow | Professor might redirect to the same page if professor is not applicable or missed any required field. |
| Assumptions: | Professor must have Facebook account. |

|  |  |
| --- | --- |
| Use Case ID: | **107** |
| Use Case Name: | **Add University** |
| Actors: | Professor |
| Priority | Normal |
| Description: | Professor will provide the required info in order to register his/her university to the website. |
| Preconditions: | Professor must be login and university must not already been registered by another Professor. |
| Post conditions: | None. |
| Extends | None |
| Uses | Login |
| Normal Flow | Professor will provide the required info and the redirected to the home page with the success message. |
| Alternate Flow | Professor might redirect to the same page if user is not applicable or missed any required field. |
| Assumptions: | Professor must have Facebook account. |

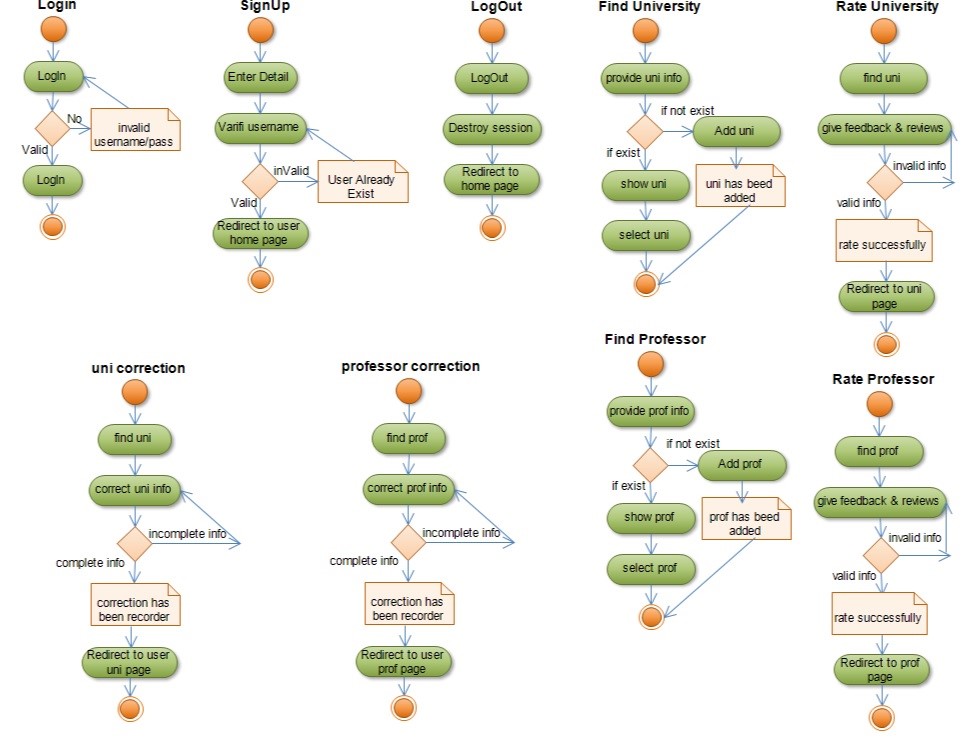
|  |  |
| --- | --- |
| Use Case ID: | **108** |
| Use Case Name: | **Report Professor or University rating** |
| Actors: | Professor/Student |
| Priority | Normal |
| Description: | User will select any rating if he/she thinks that it is not a true rating then he/she can report that rating. |
| Preconditions: | user must be login and there must be a review that can be report. |
| Post conditions: | None. |
| Extends | View Prof’s rating |
| Uses | Login |
| Normal Flow | User will provide the required info and the redirected to the home page with the Thanking message. |
| Alternate Flow | User might redirect to the same page if user is not applicable or missed any required field. |
| Assumptions: | User must have Facebook account. |

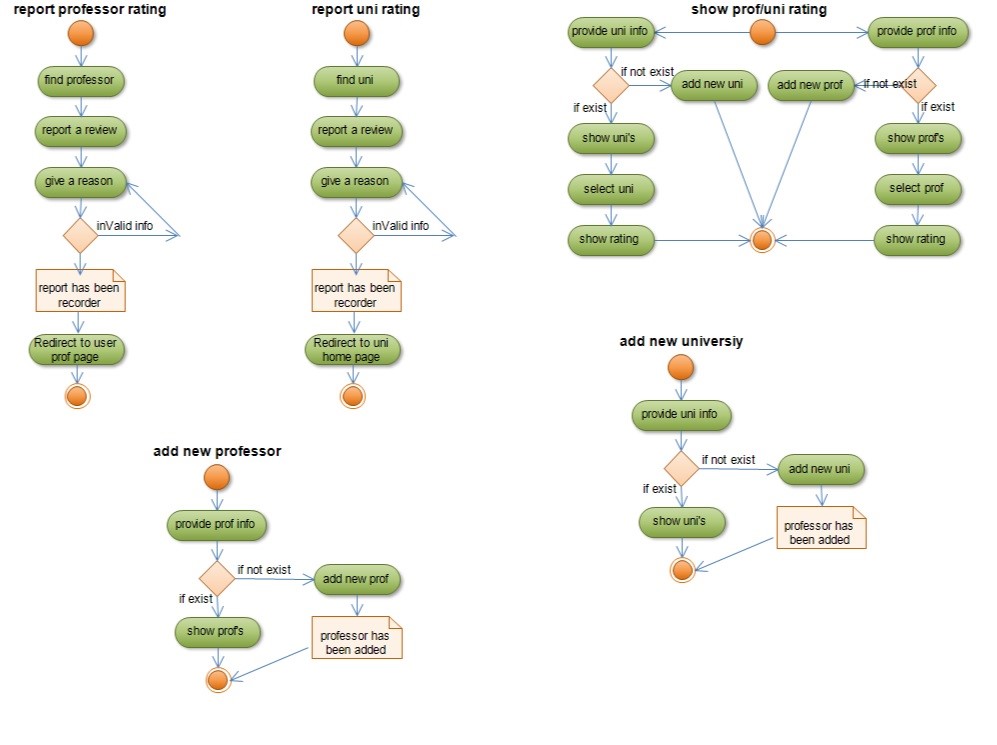
|  |  |
| --- | --- |
| Use Case ID: | **109** |
| Use Case Name: | **Logout** |
| Actors: | Student/Professor |
| Priority | Normal |
| Description: | If user does not want to perform any other activity on the website, then user can logout. |
| Preconditions: | User must be login. |
| Post conditions: | None |
| Extends | None |
| Uses | Login |
| Normal Flow | After pressing logout button user will be redirected to the home page. |
| Alternate Flow | None. |
| Assumptions: | User must have Facebook account. |

##### Figure 4.1.4 Class Diagram:

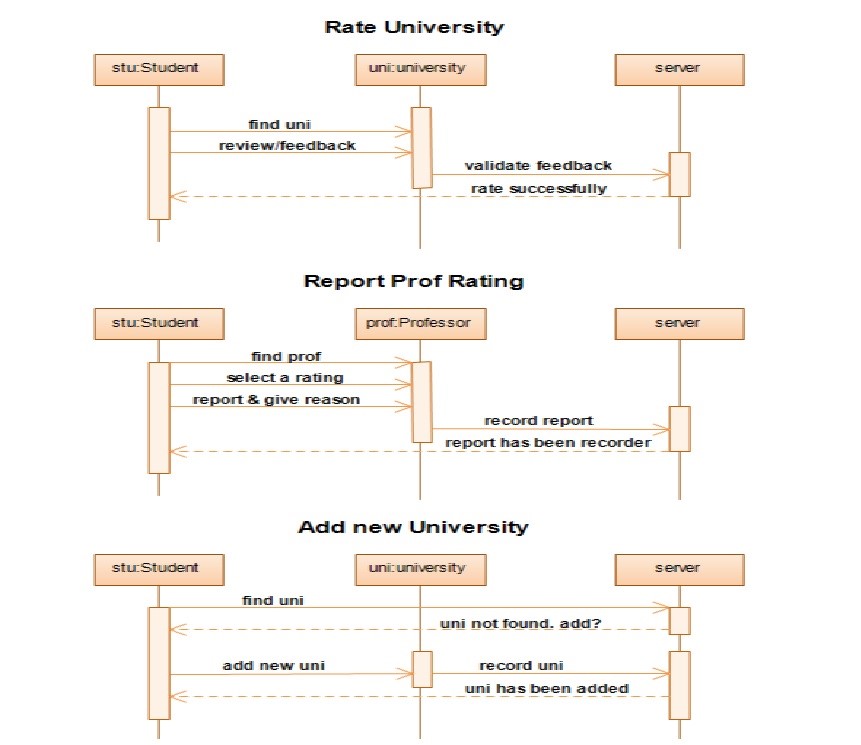


##### Figure 4.1.5 Activity Diagram:

****

****

##### Figure 4.1.6 Sequence Diagram:

****

#### 

## Results

Academic Reviews will help all those students and their parents who are in searching of the best universities for their children from where they can get best education and the students of same university can decide that in which course they should have enrolled and from which teacher. So our website will help them to take a right decision.

# Conclusion

We are developing a platform where graduate or current studying students can give their feedback on universities from where they graduate and on professors by whom they taught. This feedback will help other students and their parents who are in searching of best universities.

# References

1. <http://www.uloop.com/professors>
2. <https://www.myedu.com>

# Appendix A – Development

# Environment

## Operating Systems(s)

### Windows 7

Windows 7 is a personal computer [operating system](http://en.wikipedia.org/wiki/Operating_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). It was initially targeted mainly at the home, education, and creative professional markets.  *Developer* – Microsoft

*Preceded by* – Windows Vista (2007)

*Succeeded by* – Windows 8 (2012)

*Platforms* – IA-32 and x86-64

## Database

MYSQL is a RDBMS (Relational Database Managements System) which is use create and manage databases it actually uses the SQL language in order to perform an activity on MYQSL RDBMS. It is used by many languages in order to communicate with the database one of the most common language which is being used by the developers is PHP.

*Original Authors* – Oracle

*Developer* – Oracle Corporation

*Type* – RDBMS

*Operating System* – Windows, Linux, Solaris, OS X , FreeBSD

*License* – GPL

*Website* – <http://mysql.com>

## Programming Language(s)

### PHP

PHP is a server side scripting language which is used to develop a different sort of web applications. PHP uses the Apache server in order to run. Using PHP we can develop different sort of CRM, CMS like WordPress and other stuff. PHP uses the HTML and CSS for the attractive frontend view.

*Developer* – The PHP Group

*Appeared in – 1995*

*OS- Cross Platform*

*Website* - <https://www.php.net/>

### HTML5

HTML (Hyper Text Markup language) is a scripting language which is purely based on tags. HTML is used to define the structure of the web page that how our web app will be structured. The latest version of the HTML is HTML5 which have introduced some new tags like Header, Footer, Article, videos and others.

*Developed by* – World wide Web Consortium and WHATWG

*Open format – Yes*

*Type of Format- Markup Language*

### CSS

CSS (Cascading Style Sheet) is actually used to styling our web application. It uses the class and id’s to recognize the HTLM tag uniquely and then perform some styles on those tags to make it more beautiful.

*Developed by* – World wide Web Consortium, Bert Bos and HakonWium Lie

*Initial Release – December 17,1996*

*Type of Format- Style Sheet Language*

### JAVASCRIPT

JavaScript is a [dynamic](http://en.wikipedia.org/wiki/Dynamic_programming_language) computer [programming language](http://en.wikipedia.org/wiki/Programming_language). It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate [asynchronously](http://en.wikipedia.org/wiki/Ajax_(programming)), and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed. It is also being used in server-side network programming (with [Node.js](http://en.wikipedia.org/wiki/Node.js)), game development and the creation of desktop and mobile applications.

*Designed by* – Brendan Eich

*Developed by* – Netscape communications Corporations, Mozilla Foundation

*Stable Release – March 22,2011*

*Type of Format – Scripting Language*

### JQUERY

JQuery is the famous JavaScript library which has reduces the efforts of the developer. JQuery uses the JavaScript code to facilitate their users. JQuery can be defined as “Do Less And Get More”. It is use for rapid application development it is concise and fastest library of JavaScript it also make it easy to make a AJAX call to the server.

*Initial Release – August 26 ,2006*

*Type – JavaScript Library*

*Written in - JavaScript*

*License – MIT*

*Website –* [*www.jquery.com*](http://www.jquery.com)

### AJAX

Ajax is a programing language which is used to develop asynchronous Web Application, using Ajax we can call any server side function asynchronously to make the Web application more effective. Asynchronous call hit the server without refreshing the web page. Using Ajax we can call different sort of web services asynchronously so we can send and receive the data without refreshing the page.

## Programming Paradigm

### Object-oriented programming

Object-oriented programming (OOP) is a programming paradigm that represents the concept of "objects" that have data fields (attributes that describe the object) and associated procedures known as methods. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

## Communication and Leadership

### Democratic Decentralized

Decentralization (or decentralization) is the process of redistributing or dispersing functions, powers, people or things away from a central location or authority. Idea with the majority is accepted. There is no prime leader in-group.

# Appendix B – Tools and Softwares

## Xampp

XAMPP is a [free and open source](http://en.wikipedia.org/wiki/Free_software) [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [web server](http://en.wikipedia.org/wiki/Web_server) [solution stack](http://en.wikipedia.org/wiki/Solution_stack) package, consisting mainly of the [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), [MySQL](http://en.wikipedia.org/wiki/MySQL) [database](http://en.wikipedia.org/wiki/Database), and [interpreters](http://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](http://en.wikipedia.org/wiki/PHP) and [Perl](http://en.wikipedia.org/wiki/Perl) [programming languages](http://en.wikipedia.org/wiki/Programming_language)

*Developer –* Apache Friends

*Stable Release –* 1.8.3-5 /August 22,2014

*Environment –* Cross-Platform (Linux, Windows, Max OS X)

*Type –* WAMP, MAMP, SAMP, LAMP.

*License –* GNU GPL.  
*Website* - <https://apachefriends.org/index.html>

## Mozilla Firefox

Mozilla Fire Fox is actually a web browser which is used for surfing and it also provide some tools for the ease of the developers so they can easily track the low level of vulnerabilities.  
*Version – 31.0* released on July 22, 2014  
*Environment –* Mac OS X (10.10)   
*License –* MPL  
*Website* – [**http://mozilla.org/firefox**](http://mozilla.org/firefox)

## Sublime Text

Sublime Text is actually a cross platform text editor. Mainly use by python developers, Sublime Text support many markup languages as well as Programing languages we can extend its functionality as well by using different sort of plugins provided by third party companies.

*Original authors – Macromedia*

*Developers –* Adobe systems

*Operating System –* Windows, OS X

*Stable Release –* CC(2014) / June 18 2014  
*License –* Proprietary  
*Website* – <http://www.adobe.com/products/dreamweaver/>

# Appendix C – Code

Our Website has main feature in which professor can register him/herself or his/her university to the website, students can rate university/Professor or report any other student rating. Students can share their professor/University stats to the social websites. This web application fetches the students posts and comments on their professor from Facebook and perform data mining on that data and then show the result to the professor’s rating page. And below is the code that how this all things get done:

* Add Professor
* Add University
* Rate Professor
* Rate University
* Report Student rating
* Share Professor/University stats on social websites.
* Fetch the posts and comments from Facebook and Perform data mining on that data.

## AddProf.php:

<?php

//if user search from main search bar

if (isset($\_GET['uniName']) && isset($\_GET['profName']) && isset($\_GET['profNameln'])) {

$qry=encrypt\_url("uniName=".$\_GET['uniName']."&profName=".$\_GET['profName']."&profNameln=".$\_GET['profNameln']);

header("location:showProfs.php?".$qry);

}

include('classes/student\_class.php');

$prof\_exist=false;$userIsProfessor=false;$showMessage="";

//agr user login nai hai to use login karwao

session\_start();

if(!isset($\_SESSION['user\_status'])){

$request=$\_SERVER['REQUEST\_URI'];

header('location:member.php?req='.$request);

}

if (isset($\_SESSION["lecturerPlaces"])) {

# code...

//echo "lecturerPlaces set";

if ($\_SESSION["lecturerPlaces"]!="") {

# code...

$userIsProfessor=true;

$lecturerAt=explode("\*\*\*", $\_SESSION["lecturerPlaces"]);

}else{/\*echo $showMessage="You do not have any rights to add any professor.";\*/}

}

//if submit button pressed to add a professor

if (isset($\_POST['\_action\_save'])) {

# code...

$firstName=$\_POST['fname'];

$middleName=null;

if ($\_POST['mname']!="") {

# code...

$middleName=$\_POST['mname'];

}

$lastName=$\_POST['lname'];

$country=$\_POST['country'];

$university=$\_POST['uniV'];

$department=$\_POST['dept'];

$start\_year=$\_POST['startDate'];

$end\_year=0;

if ($\_POST['endDate']!=null) {

# code...

$end\_year=$\_POST['endDate'];

}

$directory=null;

if ($\_POST['url']!=null) {

# code...

$directory=$\_POST['url'];

}

$contact=$\_POST['contact'];

$email=$\_SESSION['user\_email'];

$av\_status="pending";

//if professor is unique then allow(insert) else not allow.

$chckqru="SELECT \* FROM professor";

$myqry=mysql\_query($chckqru);

while ($val\_of=mysql\_fetch\_array($myqry)) {

if ($firstName==$val\_of['first\_name'] && $lastName==$val\_of['last\_name'] && $university==$val\_of['school\_name']) {

$prof\_exist=true;

}else{}

}

if ($prof\_exist!=true) {

# code...

if ($\_SESSION["lecturerPlaces"]!="") {

$stu = new Student();

$stu->addProf($firstName,$middleName,$lastName,$university,$department,$directory,$country,$email,$start\_year,$end\_year,$contact,$av\_status);

}else{

/\*echo "<script>if (window.confirm('You do not have any rights to add any professor')){document.location = 'index.php'}else{document.location = 'index.php'}</script>";\*/

header('location:index.php#invalidProf');

}

}

}//end of submit button

?>

## AddUni.php:

<?php

//if user search from main search bar

if (isset($\_GET['uniName']) && isset($\_GET['profName']) && isset($\_GET['profNameln'])) {

# code...

$qry=encrypt\_url("uniName=".$\_GET['uniName']."&profName=".$\_GET['profName']."&profNameln=".$\_GET['profNameln']);

header("location:showProfs.php?".$qry);

}

include('classes/student\_class.php');

$uni\_exist=false;$userIsProfessor=false;$showMessage="";

//agr user login nai hai to use login karwao

session\_start();

if(!isset($\_SESSION['user\_status'])){

$request=$\_SERVER['REQUEST\_URI'];

header('location:member.php?req='.$request);

}

if (isset($\_SESSION["lecturerPlaces"])) {

# code...

//echo "lecturerPlaces set";

if ($\_SESSION["lecturerPlaces"]!="") {

# code...

$userIsProfessor=true;

$lecturerAt=explode("\*\*\*", $\_SESSION["lecturerPlaces"]);

}else{/\*echo $showMessage="You do not have any rights to add any University.";\*/}

}

//if submit button pressed to add University

if (isset($\_POST['\_action\_saveSchool'])) {

# code...

$uniName=$\_POST['school'];

$nickName=$\_POST['school\_n'];

$country=$\_POST['country'];

$countryID=null;

$stateProv=$\_POST['statess'];

$city=$\_POST['city'];

$Website=null;

if ($\_POST['website']!=null) {

# code...

$Website=$\_POST['website'];

}

$av\_status="pending";

//just to get country ID

$cntryInfoQry="SELECT \* FROM country WHERE country\_name='$country'";

$myqry=mysql\_query($cntryInfoQry);

while ($val\_of=mysql\_fetch\_array($myqry)) {

$countryID=$val\_of['country\_id'];

}

//end of geting countryID

//if school is unique then allow(insert) else not allow.

$chckqru="SELECT \* FROM school";

$myqry=mysql\_query($chckqru);

while ($val\_of=mysql\_fetch\_array($myqry)) {

if ($uniName==$val\_of['school\_name'] && $city==$val\_of['city\_name']) {

$uni\_exist=true;

}else{}

}

if ($uni\_exist!=true) {

# code...

if ($\_SESSION["lecturerPlaces"]!="") {

$stu = new Student();

$stu->addUni($uniName,$nickName,$countryID,$stateProv,$city,$Website,$av\_status);

}else{

//echo $showMessage="You do not have any rights to add any University.";

/\*echo "<script>if (window.confirm('You do not have any rights to add any University')){document.location = 'index.php'}else{document.location = 'index.php'}</script>";\*/

header('location:index.php#invalidProf');

}

}

}//end of submit button

?>

## rateProf.php:

<?php

//if user search from main search bar

if (isset($\_GET['uniName']) && isset($\_GET['profName']) && isset($\_GET['profNameln'])) {

# code...

$qry=encrypt\_url("uniName=".$\_GET['uniName']."&profName=".$\_GET['profName']."&profNameln=".$\_GET['profNameln']);

header("location:showProfs.php?".$qry);

}

parse\_Myurl();

include('classes/professor\_class.php');

include('classes/student\_class.php');

$rating\_exist=false;$userIsStudent=false;$showMessage="";$uniNickName="";$uniMatched=null;$belongToUni=false;$profStartYear=null;$profEndYear=null;$profEmail=null;$belongToProf=true;$selfRating=false;

$nameMch=0;$nickNameMch=0;

//agr user login nai hai to use login karwao

session\_start();

if(!isset($\_SESSION['user\_status'])){

$request=$\_SERVER['REQUEST\_URI'];

header('location:member.php?req='.$request);

}

if (isset($\_SESSION["educationPlaces"])) {

# code...

//echo "lecturerPlaces set";

if ($\_SESSION["educationPlaces"]!="") {

# code...

$userIsStudent=true;

$studentAt=explode("\*\*\*", $\_SESSION["educationPlaces"]);

/\*getting the nick name of the university,prof start year and end year\*/

$chckqru="SELECT p.first\_name,p.last\_name,p.school\_name,p.start\_year,p.end\_year,p.email,p.available\_status,s.school\_nick\_name

FROM professor p,school s

WHERE p.first\_name='$\_GET[fname]' AND p.last\_name='$\_GET[lname]' AND p.school\_name=s.school\_name AND p.school\_name='$\_GET[univName]' AND p.available\_status='approved'";

$exQry=mysql\_query($chckqru);

while ($valOf=mysql\_fetch\_array($exQry)) {

# code...

$uniNickName=$valOf['school\_nick\_name'];

$profStartYear=$valOf['start\_year'];

$profEndYear=$valOf['end\_year'];

$profEmail=$valOf['email'];

}

/\*end of getting the nick name of the university,prof start year and end year\*/

if ($profEndYear==0) {

# code...

$profEndYear=date("Y");

}

//agr valid user ko b restrict kar raha hoga rate karne se to phr me apna algo likhunga for matching uni's

//print\_r($studentAt);

/\*matching the university name with the universitie's name of student attended\*/

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($\_GET['univName']),strtolower($studentAt[$i]),$nameMchTemp);

if ($nameMchTemp>$nameMch) {

# code...

$nameMch=$nameMchTemp;

}

}

//echo $nameMch."<br>";

/\*matching the university nick name with the universitie's name of student attended\*/

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($uniNickName),strtolower($studentAt[$i]),$nickNameMchTemp);

if ($nickNameMchTemp>$nickNameMch) {

# code...

$nickNameMch=$nickNameMchTemp;

}

}

//echo $nickNameMch."<br>";

if ($nameMch>80 || $nickNameMch>80) {

# code...

$belongToUni=true;

}

}else{}

}

//using hiddenfield of helprating(required by JS) instead of submit button because JS is unseting submit button.

if(isset($\_POST['helprating'])){

$prof=new Professor();

$start\_date=null;$stuEndYear=null;

$prof->\_set('first\_name',$\_GET['fname']);

$prof->\_set('last\_name',$\_GET['lname']);

$prof->\_set('school\_name',$\_GET['univName']);

$courseCode=$\_POST['courseName'];

$helpfullness=$\_POST['helprating'];

$clarity=$\_POST['clarityrating'];

$easiness=$\_POST['easinessrating'];

$course\_credit=$\_POST['course\_credit'];

$comments=$\_POST['comments'];

$your\_interest=$\_POST['interestrating'];

$textbookUse=$\_POST['textbookrating'];

$fName=$prof->\_get('first\_name');

$lName=$prof->\_get('last\_name');

$uniName=$prof->\_get('school\_name');

$stu\_email=$\_SESSION['user\_email'];

$rating\_date=date("m/d/Y");

//$attendance=$\_POST['attendance'];

if ($\_POST['attendance']!=null) {

# code...

$attendance=$\_POST['attendance'];

}

else{

$attendance="";

}

//$textbook\_used=$\_POST['rTextBookUsed'];

if ($\_POST['rTextBookUsed']!=null) {

# code...

$textbook\_used=$\_POST['rTextBookUsed'];

}

else{

$textbook\_used=null;

}

//$gradeRecieved=$\_POST['grade'];

if ($\_POST['grade']!=null) {

# code...

$gradeRecieved=$\_POST['grade'];

}

else{

$gradeRecieved=null;

}

//$start\_date=$\_POST['startDate'];

if ($\_POST['startDate']!=null) {

# code...

$start\_date=$\_POST['startDate'];

}

else{

$start\_date=null;

}

//$start\_date=$\_POST['startDate'];

if ($\_POST['endDate']!=null) {

# code...

$end\_date=$\_POST['endDate'];

$stuEndYear=$\_POST['endDate'];

}

else{

$end\_date=null;

$stuEndYear=date("Y");

}

$av\_status="pending";

$start\_date."-".$stuEndYear."<br>";

$profStartYear."-".$profEndYear."<br>";

/\*if user belongs to professor or not\*/

if ($profEndYear<$start\_date || $stuEndYear<$profStartYear) {

# code...

$belongToProf=false;

}

//if rating is unique then allow(insert) else not allow.

$chckqru="SELECT \* FROM prof\_rating";

$myqry=mysql\_query($chckqru);

while ($val\_of=mysql\_fetch\_array($myqry)) {

//uncomment the below condition if student want to rate same prof for different courses(course id)

/\*if ($fName==$val\_of['first\_name'] && $lName==$val\_of['last\_name'] && $uniName==$val\_of['school\_name'] && $stu\_email==$val\_of['stu\_email'] && $courseCode==$val\_of['class\_id'])\*/

if ($fName==$val\_of['first\_name'] && $lName==$val\_of['last\_name'] && $uniName==$val\_of['school\_name'] && $stu\_email==$val\_of['stu\_email']) {

$rating\_exist=true;

}else{}

if ($stu\_email==$profEmail) {

# code...

$selfRating=true;

}

}

if ($rating\_exist!=true && $userIsStudent==true && $belongToUni==true && $belongToProf==true && $selfRating==false) {

# code...

//INSERT INTO student(first\_name, last\_name, stu\_email, stu\_pass, stu\_school) VALUES(first\_name,last\_name,stu\_email,stu\_pass,stu\_school);

$stu = new Student();

$stu->rateProf($courseCode,$helpfullness,$clarity,$easiness,$course\_credit,$comments,$attendance,$your\_interest,$textbookUse,$textbook\_used,$gradeRecieved,$rating\_date,$fName,$lName,$stu\_email,$uniName,$start\_date,$end\_date,$av\_status);

}

else{

//echo "you have already rate this professor for this course";

}

}

?>

## rateUni.php:

<?php

//if user search from main search bar

if (isset($\_GET['uniName']) && isset($\_GET['profName']) && isset($\_GET['profNameln'])) {

# code...

$qry=encrypt\_url("uniName=".$\_GET['uniName']."&profName=".$\_GET['profName']."&profNameln=".$\_GET['profNameln']);

header("location:showProfs.php?".$qry);

}

parse\_Myurl();

include('classes/school\_class.php');

include('classes/student\_class.php');

$rating\_exist=false;$userIsStudent=false;$showMessage="";$uniNickName="";$uniMatched=null;$belongToUni=false;

$nameMch=0;$nickNameMch=0;

//agr user login nai hai to use login karwao

session\_start();

if(!isset($\_SESSION['user\_status'])){

$request=$\_SERVER['REQUEST\_URI'];

header('location:member.php?req='.$request);

}

if (isset($\_SESSION["educationPlaces"])) {

# code...

//echo "lecturerPlaces set";

if ($\_SESSION["educationPlaces"]!="") {

# code...

$userIsStudent=true;

$studentAt=explode("\*\*\*", $\_SESSION["educationPlaces"]);

/\*getting the nick name of the university\*/

$chckqru="SELECT \* FROM school WHERE school\_name='$\_GET[univName]' AND state\_name='$\_GET[stateName]' AND city\_name='$\_GET[citName]'";

$exQry=mysql\_query($chckqru);

while ($valOf=mysql\_fetch\_array($exQry)) {

# code...

$uniNickName=$valOf['school\_nick\_name'];

}

/\*end of getting the nick name of the university\*/

//agr valid user ko b restrict kar raha hoga rrate karne se to pht me apna algo likhunga for matching uni's

//print\_r($studentAt);

/\*matching the university name with the universitie's name of student attended\*/

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($\_GET['univName']),strtolower($studentAt[$i]),$nameMchTemp);

if ($nameMchTemp>$nameMch) {

# code...

$nameMch=$nameMchTemp;

}

}

//echo $nameMch."<br>";

/\*matching the university nick name with the universitie's name of student attended\*/

for ($i=0; $i <count($studentAt) ; $i++) {

# code...

similar\_text(strtolower($uniNickName),strtolower($studentAt[$i]),$nickNameMchTemp);

if ($nickNameMchTemp>$nickNameMch) {

# code...

$nickNameMch=$nickNameMchTemp;

}

}

//echo $nickNameMch."<br>";

if ($nameMch>80 || $nickNameMch>80) {

# code...

$belongToUni=true;

}

}else{}

}

//using hiddenfield of rep\_rating(required by JS) instead of submit button because JS is unseting submit button.

if(isset($\_POST['rep\_rating'])){

$schl=new School();

$schl->\_set('reputation',$\_POST['rep\_rating']);

$schl->\_set('location',$\_POST['location\_rating']);

$schl->\_set('opportunities',$\_POST['opt\_rating']);

$schl->\_set('library',$\_POST['library\_rating']);

$schl->\_set('ground\_common\_areas',$\_POST['campus\_rating']);

$schl->\_set('internet',$\_POST['internet\_rating']);

$schl->\_set('food',$\_POST['food\_rating']);

$schl->\_set('club',$\_POST['club\_rating']);

$schl->\_set('social',$\_POST['social\_rating']);

$schl->\_set('hapiness',$\_POST['happy\_rating']);

$schl->\_set('school\_name',$\_GET['univName']);

$schl->\_set('city\_name',$\_GET['citName']);

$graduate\_year=$\_POST['crYearGraduating'];

$comment=$\_POST['crComments'];

$stu\_email=$\_SESSION['user\_email'];

$rating\_date=date("m/d/Y");

$av\_status="pending";

$city\_name=$schl->\_get('city\_name');

$uniName=$schl->\_get('school\_name');

//if rating is unique then allow(insert) else not allow.

$chckqru="SELECT \* FROM school\_rating";

$myqry=mysql\_query($chckqru);

while ($val\_of=mysql\_fetch\_array($myqry)) {

if ($uniName==$val\_of['school\_name'] && $city\_name==$val\_of['city'] && $stu\_email==$val\_of['stu\_email']) {

$rating\_exist=true;

}else{}

}

if ($rating\_exist!=true && $userIsStudent==true && $belongToUni==true) {

# code...

$reputation=$schl->\_get('reputation');

$location=$schl->\_get('location');

$opportunities=$schl->\_get('opportunities');

$library=$schl->\_get('library');

$ground\_common\_areas=$schl->\_get('ground\_common\_areas');

$internet=$schl->\_get('internet');

$food=$schl->\_get('food');

$club=$schl->\_get('club');

$social=$schl->\_get('social');

$hapiness=$schl->\_get('hapiness');

$stu = new Student();

$stu->rateUni($reputation,$location,$opportunities,$library,$ground\_common\_areas,$internet,$food,$club,$social,$hapiness,$graduate\_year,$rating\_date,$comment,$city\_name,$stu\_email,$uniName,$av\_status);

}

else{

//echo "you have already rate this University";

}

}

?>

## flagTeacherRating.php:

<?php

//if user search from main search bar

if (isset($\_GET['uniName']) && isset($\_GET['profName']) && isset($\_GET['profNameln'])) {

# code...

$qry=encrypt\_url("uniName=".$\_GET['uniName']."&profName=".$\_GET['profName']."&profNameln=".$\_GET['profNameln']);

header("location:showProfs.php?".$qry);

}

parse\_Myurl();

include('classes/student\_class.php');

$rating\_id=null;$comment=null;$report\_exist=false;$action\_result=null;

//agr user login nai hai to use login karwao

session\_start();

if(!isset($\_SESSION['user\_status'])){

$request=$\_SERVER['REQUEST\_URI'];

header('location:member.php?req='.$request);

}

if (isset($\_GET['rid'])) {

# code...

$rating\_id=$\_GET['rid'];

$comment=$\_GET['comment'];

}

if (isset($\_POST['save'])) {

# code...

$report=$\_POST['rErrorMsg'];

$stu\_email=$\_SESSION['user\_email'];

//check if stu have already report this rating

$checkreportQry="SELECT \* FROM report\_prof\_stu\_rating";

$exeQry=mysql\_query($checkreportQry);

while ($valOf=mysql\_fetch\_array($exeQry)) {

# code...

if ($valOf['rating\_id']==$rating\_id && $valOf['stu\_email']==$stu\_email) {

# code...

$report\_exist=true;

}

}

if ($report\_exist!=true) {

# code...

$stu = new Student();

$action\_result=$stu->reportProfRating($rating\_id,$stu\_email,$report);

}

}//end of submit button

?>

## shareProfStats.php:

<?php

if (isset($accessToken) && isset($\_GET['fbShare'])) {

# code...

//echo 'user is logedIn and share';

if (isset($\_SESSION['facebook\_access\_token'])) {

$fb->setDefaultAccessToken($\_SESSION['facebook\_access\_token']);

} else {

// getting short-lived access token

$\_SESSION['facebook\_access\_token'] = (string) $accessToken;

// OAuth 2.0 client handler

$oAuth2Client = $fb->getOAuth2Client();

// Exchanges a short-lived access token for a long-lived one

$longLivedAccessToken = $oAuth2Client->getLongLivedAccessToken($\_SESSION['facebook\_access\_token']);

$\_SESSION['facebook\_access\_token'] = (string) $longLivedAccessToken;

// setting default access token to be used in script

$fb->setDefaultAccessToken($\_SESSION['facebook\_access\_token']);

}

// redirect the user back to the same page if it has "code" GET variable

/\*if (isset($\_GET['code'])) {

header('location:index.php');

}\*/

try {

$request = $fb->get('/me');

} catch(Facebook\Exceptions\FacebookResponseException $e) {

// When Graph returns an error

if ($e->getCode() == 190) {

unset($\_SESSION['facebook\_access\_token']);

$helper = $fb->getRedirectLoginHelper();

/\*$loginUrl = $helper->getLoginUrl('https://apps.facebook.com/APP\_NAMESPACE/', $permissions);\*/

$loginUrl = $helper->getLoginUrl('http://localhost:8080/AR/member.php', $permissions);

echo "<script>window.top.location.href='".$loginUrl."'</script>";

exit;

}

} catch(Facebook\Exceptions\FacebookSDKException $e) {

// When validation fails or other local issues

echo 'Facebook SDK returned an error: ' . $e->getMessage();

exit;

}

// getting basic info about user

try {

// message must come from the user-end

//$current\_link = $\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI'];

$encrpt\_Qry=encrypt\_url("fname=".$\_GET['fname']."&lname=".$\_GET['lname']."&univName=".str\_replace(" ", "%20", $\_GET['univName'])."&dept=".str\_replace(" ", "%20", $\_GET['dept']));

$data = ['link' => 'http://academicreviews.azurewebsites.net/profRating.php?'.$encrpt\_Qry,'title'=>'All the students of '.$\_GET['univName'].' can review and rate Professor '.$\_GET['fname'].' '.$\_GET['lname'], 'description'=>'All the students of '.$\_GET['univName'].' can review and rate Professor '.$\_GET['fname'].' '.$\_GET['lname']];

$request = $fb->post('/me/feed', $data);

$response = $request->getGraphEdge()->asArray;

} catch(Facebook\Exceptions\FacebookResponseException $e) {

// When Graph returns an error

echo 'Graph returned an error: ' . $e->getMessage();

session\_destroy();

// redirecting user back to app login page

header("Location: ./");

exit;

} catch(Facebook\Exceptions\FacebookSDKException $e) {

// When validation fails or other local issues

//echo 'Facebook SDK returned an error: ' . $e->getMessage();

/\*agr error ka msg ka solution mil jata hai to upr wali line uncomment karni hai and niche wali line(jugar) remove karni hai\*/

//SDK se error aa raha tha to jugari kam kia hai. it can be risky

header('location:profRating.php?'.encrypt\_url("fname=".$\_GET['fname']."&lname=".$\_GET['lname']."&univName=".str\_replace(" ", "%20", $\_GET['univName'])."&dept=".str\_replace(" ", "%20", $\_GET['dept'])."&share\_success=true"));

exit;

}?>

## fetchData.php:

<?php

include('imp\_files/con\_file.php');

$post\_exist=false;$comment\_exisst=false;$innerComment\_exist=false;

ob\_start();

//include\_once \_\_DIR\_\_ .'/src/Facebook/autoload.php';

require\_once('assets/facebook-php-sdk-for-pagePosts/src/facebook.php');

$arr = [];

$arr['appId'] = '503201259853633';

$arr['secret'] = 'b0e4fc6e3ccda3g390ba638288s33262';

$arr['retunUrl'] = 'http://localhost/fb/';

$fb = new Facebook($arr);

$page\_id = '1673011501952917';

$pagefeed = $fb->api("/" . $page\_id . "/feed");

$allPost = $pagefeed;

echo '<pre>';

//delete all records from all FB tables

$exeQry=mysql\_query('delete from fb\_innercomment');

$exeQry=mysql\_query('delete from fb\_comment');

$exeQry=mysql\_query('delete from fb\_post');

foreach ($allPost['data'] as $allPost1) {

$postID=null;$postSelf=null;

echo '<pre>';

if(isset($allPost1['message'])){

print\_r('<b>'.$allPost1['message'].'</b><br>');

print\_r('<b>'.$allPost1['id'].'</b><br>');

$postID=$allPost1['id'];

$postSelf=$allPost1['message'];

$insrtQry=mysql\_query("INSERT INTO fb\_post(fb\_post\_id,fb\_post) VALUES ('$postID','$postSelf')");

}

//print\_r($allPost1['id']);

$comments = $fb->api($allPost1['id'] . '/comments');

//print\_r($comments);

foreach ($comments['data'] as $value) {

//print\_r($commentsz);

$commentID=null;$commentSelf=null;

if (isset($value['message'])) {

# code...

print\_r('<br><b>Comments:</b>'.$value['message'].'<br>');

print\_r('<br><b>ID:</b>'.$value['id'].'<br>');

$commentID=$value['id'];

$commentSelf=$value['message'];

$insrtQry=mysql\_query("INSERT INTO fb\_comment(fb\_post\_id,fb\_comment\_id,fb\_comment) VALUES ('$postID','$commentID','$commentSelf')");

}

//print\_r('<br>Profile Name: </b>'.$value['from']['name']);

$commentsz = $fb->api($value['id'] . '/comments');

foreach ($commentsz['data'] as $innerComments) {

$innerCommentID=null;$innerCommentSelf=null;

if (isset($innerComments['message'])) {

# code...

print\_r('<br><b>Inner Comments:</b>'.$innerComments['message']);

print\_r('<br><b>Inner ID:</b>'.$innerComments['id']);

$innerCommentID=$innerComments['id'];

$innerCommentSelf=$innerComments['message'];

$insrtQry=mysql\_query("INSERT INTO fb\_innercomment(fb\_comment\_id,fb\_innercomment\_id,fb\_innercomment) VALUES ('$commentID','$innerCommentID','$innerCommentSelf')");

}

//print\_r('<br><b>Profile Name: </b> </b>'.$innerComments['from']['name']);

}

}

}

?>

<?php

$data=array();$Profdata=array();$posRating=array();$negRating=array();$overallRating=array();$positiveRating=0;$negativeRating=0;$pRating=0;$nRating=0;

$getPosts="SELECT \* FROM fb\_post";

$exQry=mysql\_query($getPosts);

while ($valOf=mysql\_fetch\_array($exQry)) {

# code...

$data[]=$valOf['fb\_post'];

}

$getComments="SELECT \* FROM fb\_comment";

$exQry=mysql\_query($getComments);

while ($valOf=mysql\_fetch\_array($exQry)) {

# code...

$data[]=$valOf['fb\_comment'];

}

/\*$data[]="professor Rafiullah afridi is one of the best professor in karachi unIverSity";

$data[]="Ku me sir rafiullah afridi parhate hai!!! or acha parhate hai :).";\*/

$getInnerComments="SELECT \* FROM fb\_innercomment";

$exQry=mysql\_query($getInnerComments);

while ($valOf=mysql\_fetch\_array($exQry)) {

# code...

$data[]=$valOf['fb\_innercomment'];

}

//check if data(posts,comments,innercomments) belongs to current professor or not.

foreach ($data as $value) {

# code...

$gotFname=false;$gotLname=false;$gotUniname=false;

//echo $value."<br>";

$arr = explode(' ',$value);

foreach ($arr as $word) {

# code...

//echo $word."<br>";

//values of firstname,lastname and uniname will be fetched from the file in which this file will include

if(strtolower($word)==strtolower($first\_name)){ $gotFname=true; }

if(strtolower($word)==strtolower($last\_name)){ $gotLname=true; }

/\*if(strtolower($word)==strtolower($uni\_name)){ $gotUniname=true; }\*/

}

//checking the uni name from whole sentence

if(strpos(strtolower($value), strtolower($uni\_name)) !== false){ $gotUniname=true; }

//checking the uni nick name from whole sentence

if(strpos(strtolower($value), strtolower($uni\_nick\_name)) !== false){ $gotUniname=true; }

if ($gotFname==true && $gotLname==true && $gotUniname==true) {

# code...

$Profdata[]=$value;

}

}

//print\_r($Profdata);

//start Sentiment Analyzing

require\_once('Sentiment/src/SentimentAnalyzer.php');

$sat = new SentimentAnalyzerTest(new SentimentAnalyzer());

$sat->trainAnalyzer('Sentiment/trainingSet/data.neg', 'negative', 5000); //training with negative data

$sat->trainAnalyzer('Sentiment/trainingSet/data.pos', 'positive', 5000); //trainign with positive data

foreach ($Profdata as $Vl) {

# code...

$result = $sat->analyzeSentence(strtolower($Vl));

$posRating[]=$result['accuracy']['positivity'];

$negRating[]=$result['accuracy']['negativity'];

}

//showing both positive and negative ratings

for ($i=0; $i <count($Profdata) ; $i++) {

# code...

$positiveRating=$positiveRating+$posRating[$i];

}

for ($i=0; $i <count($Profdata) ; $i++) {

# code...

$negativeRating=$negativeRating+$negRating[$i];

}

if (count($Profdata)!=0) {

# code...

$pRating=($positiveRating/count($posRating)\*100);

$nRating=($negativeRating/count($negRating)\*100);

}

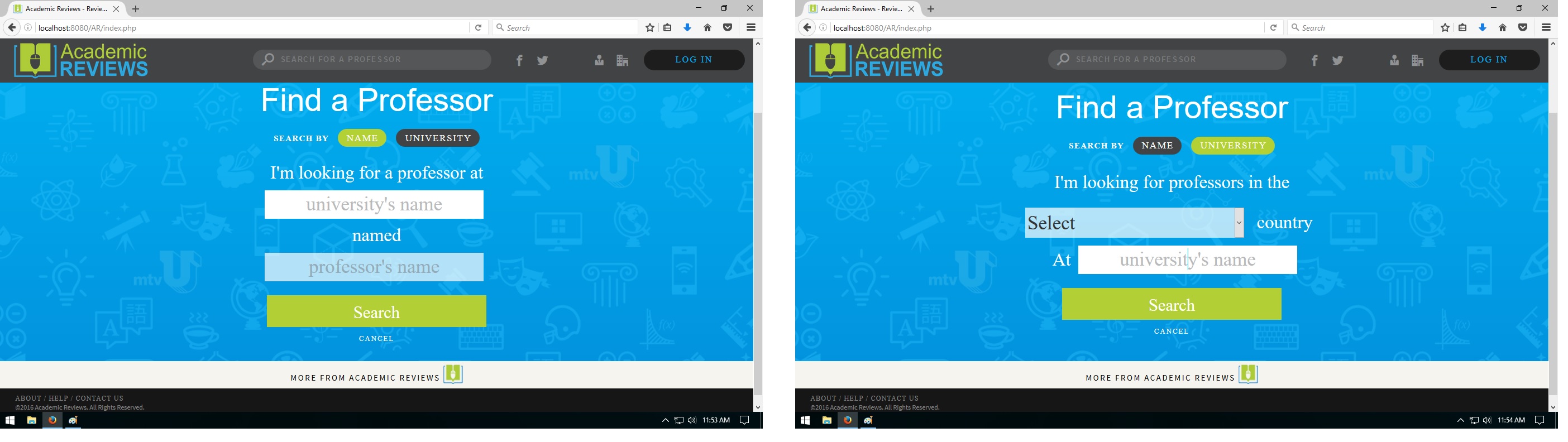
/\*echo number\_format($pRating,2,'.','');echo "<br>";

echo number\_format($nRating,2,'.','');echo "<br>";\*/

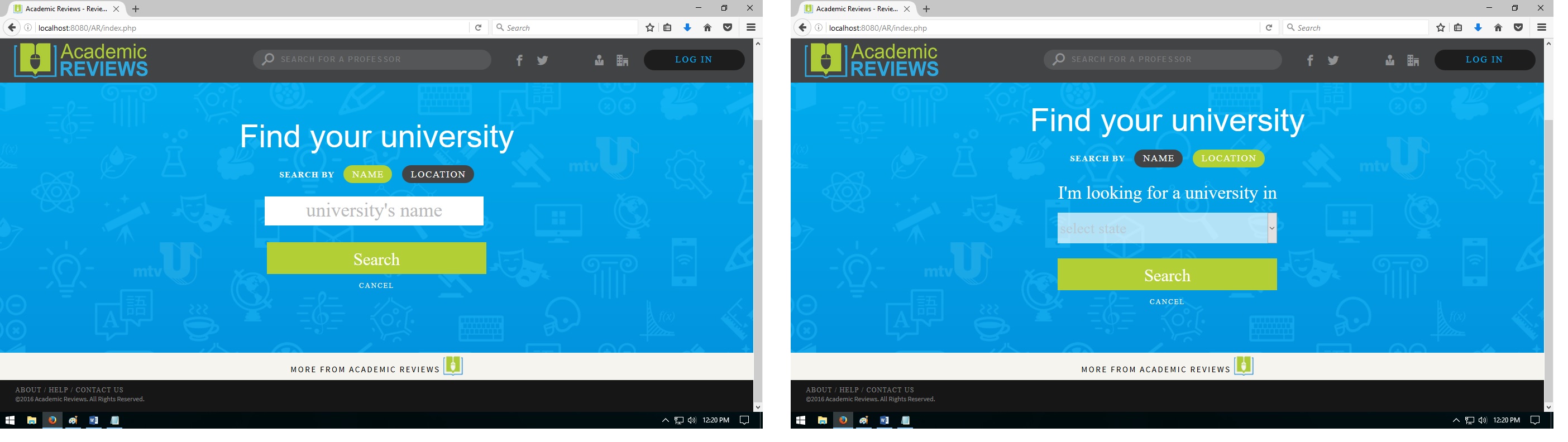
?>

Appendix D – Snapshots of Project

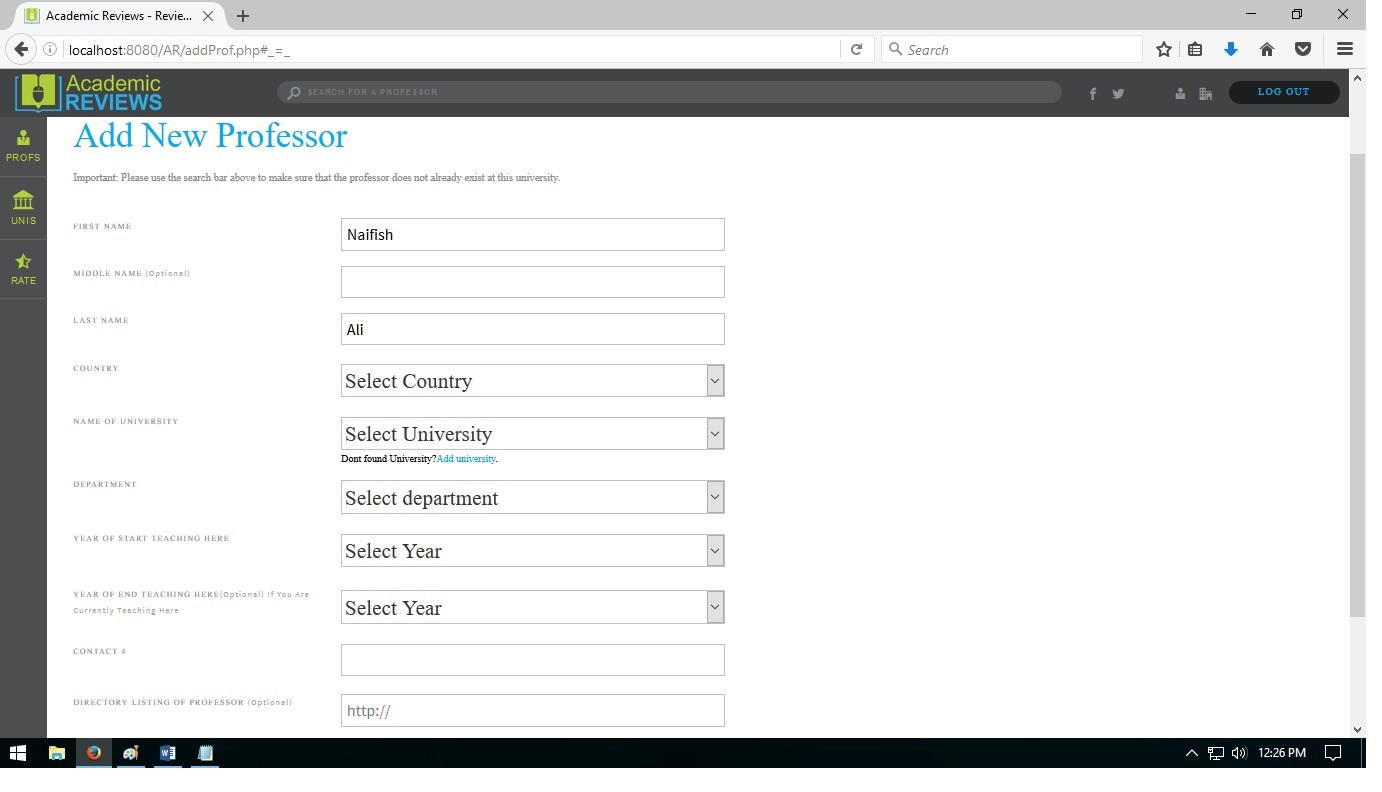
## Search Professors:



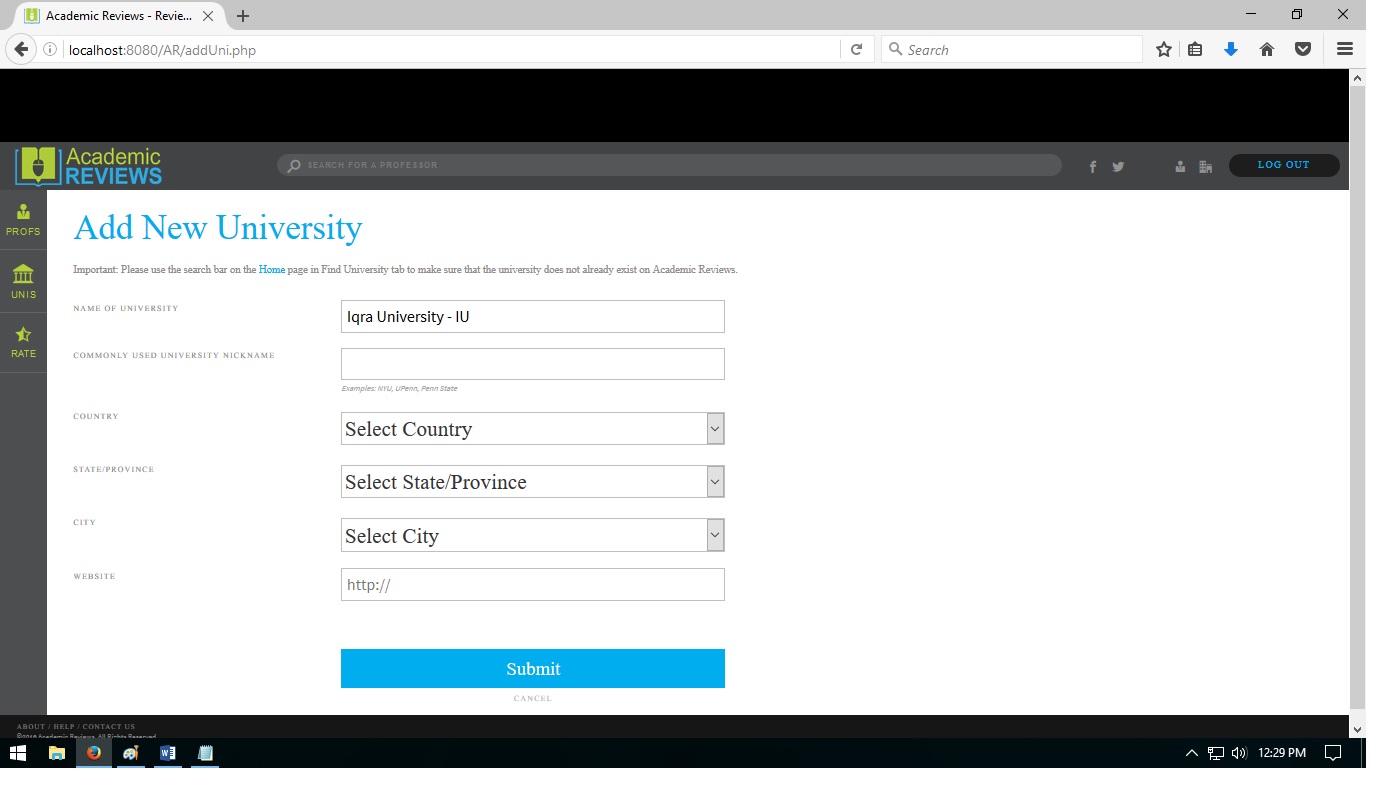
## Search Universities:



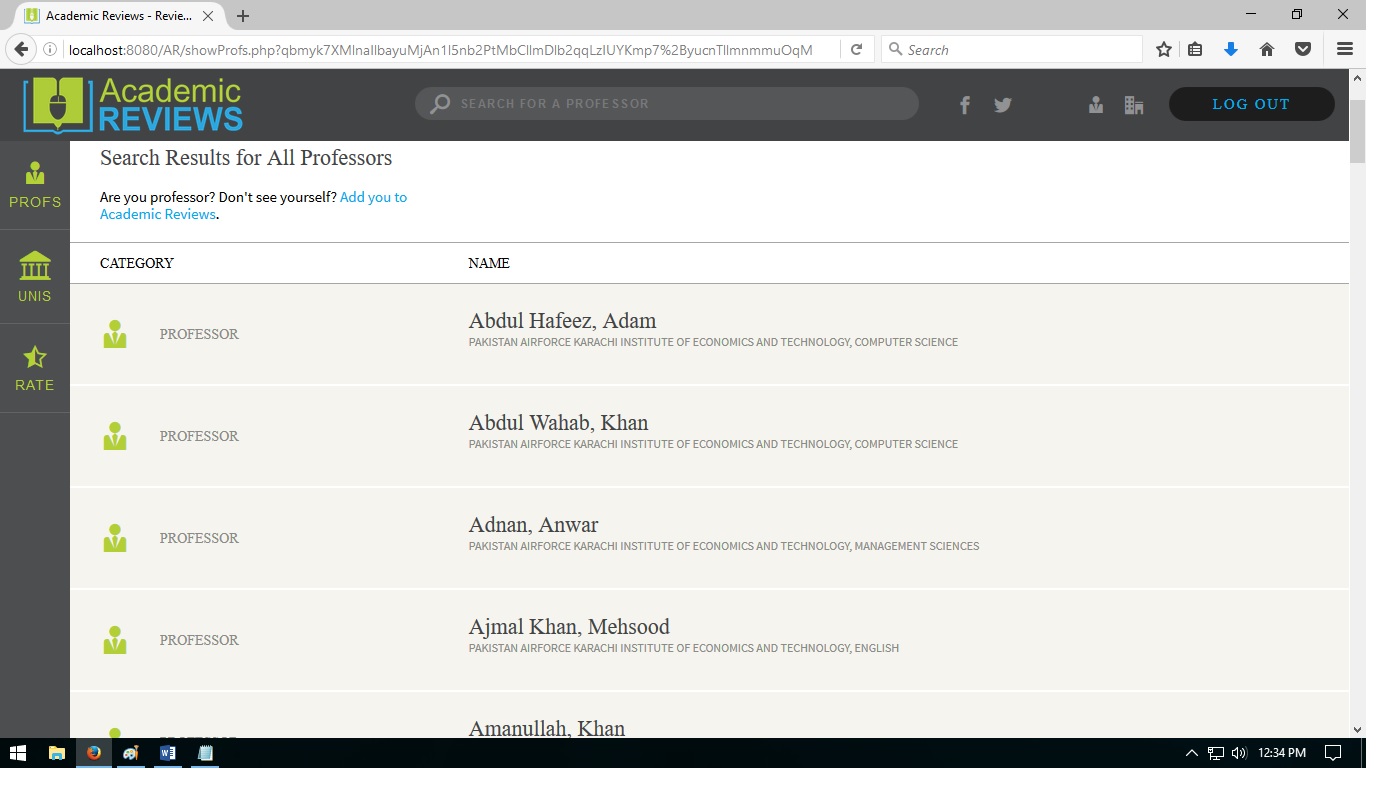
## Add Professor:



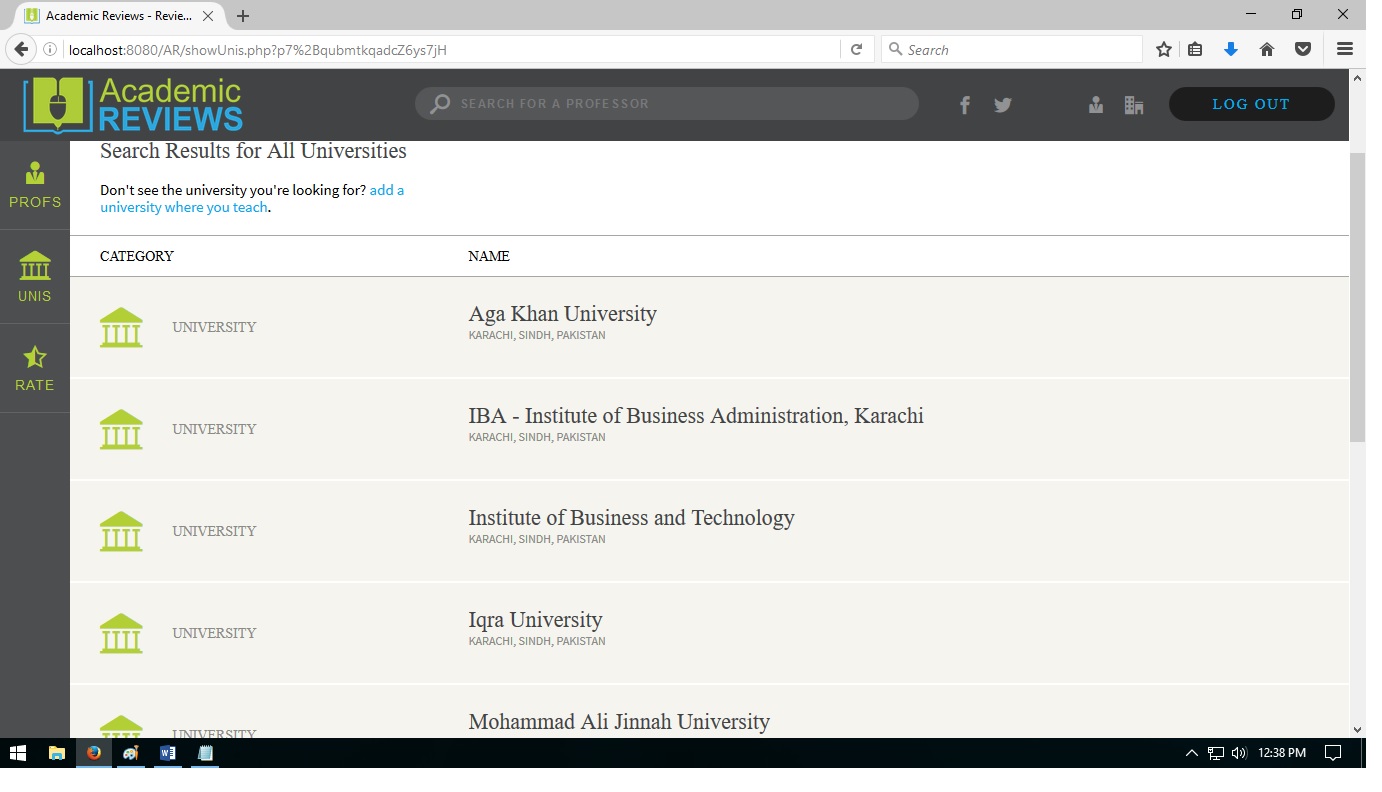
## Add University:



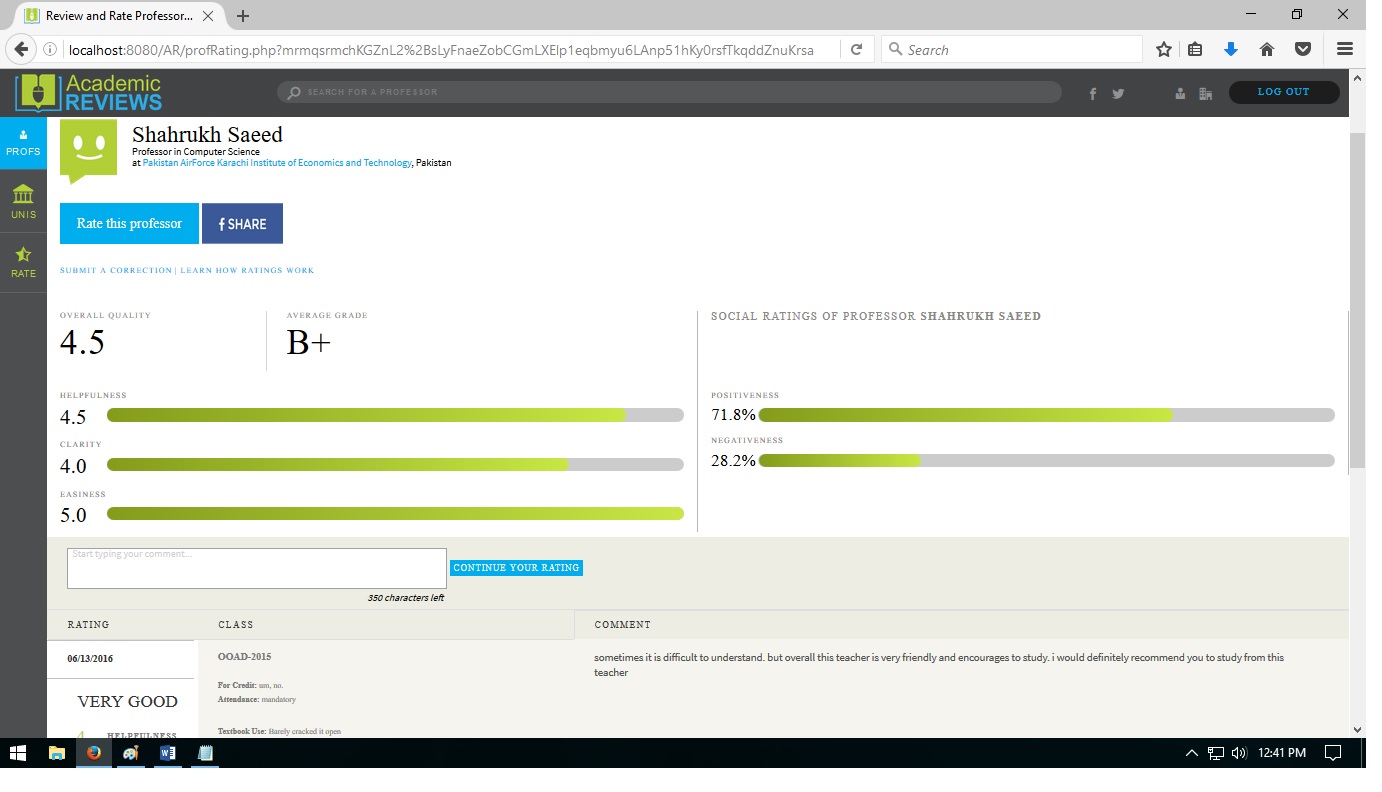
## Professor’s Search Result:



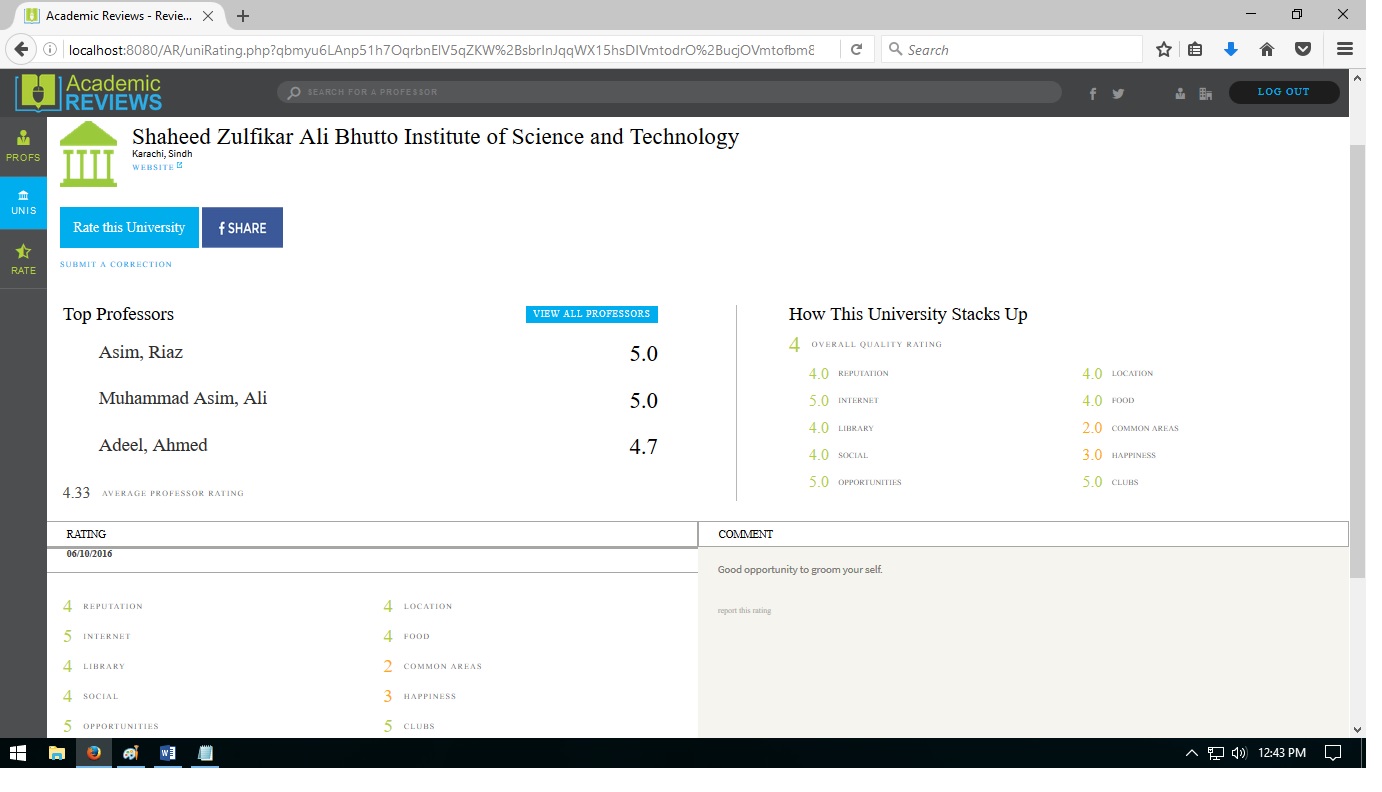
## University’s Search Result:



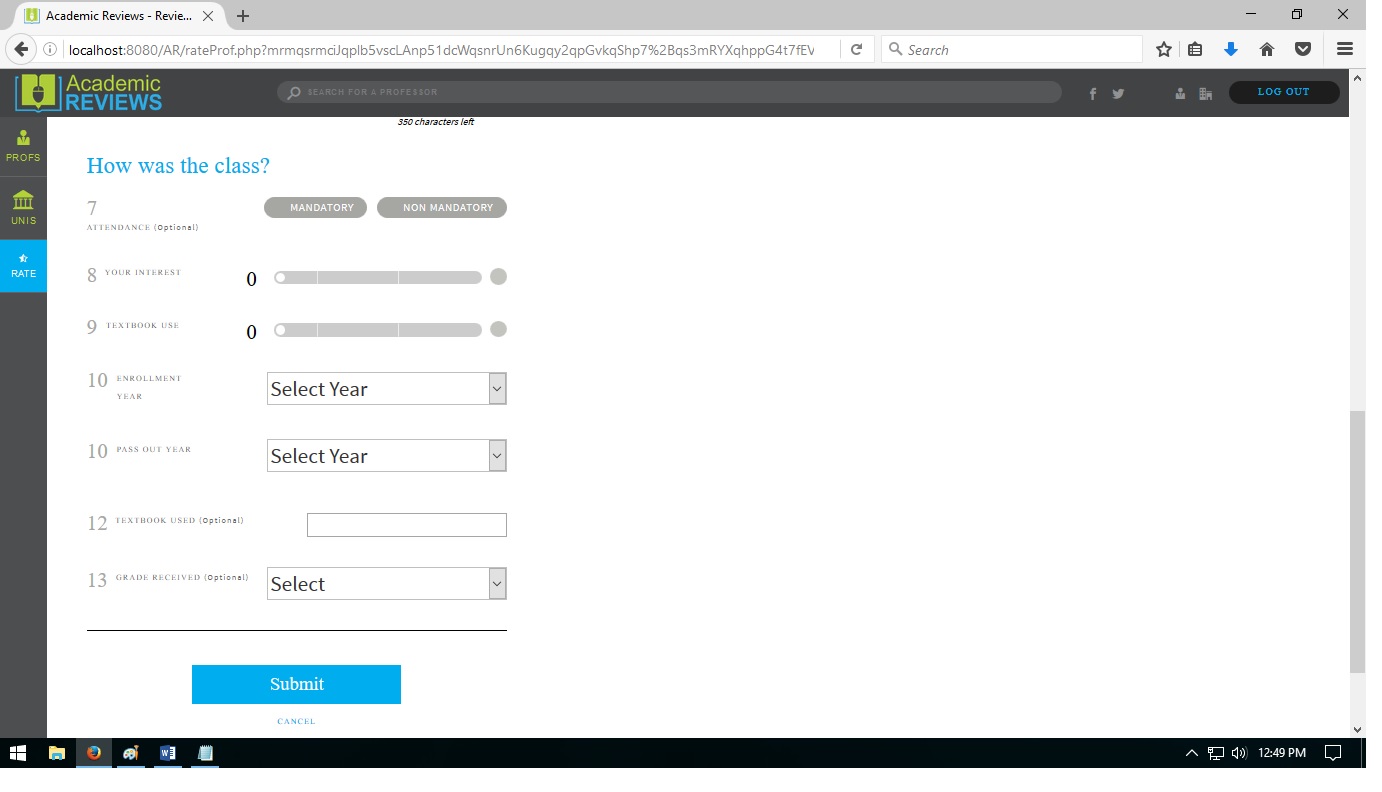
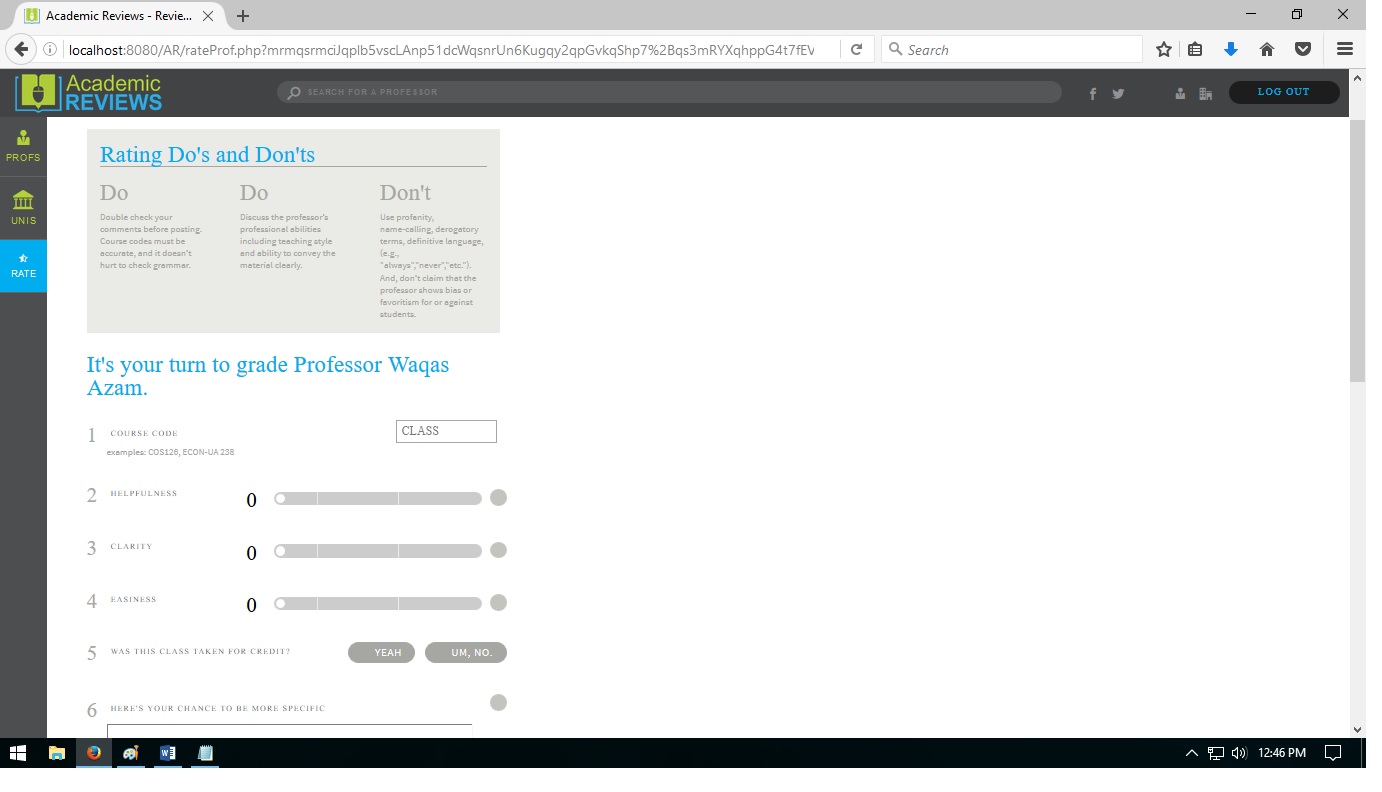
## Professor Rating:



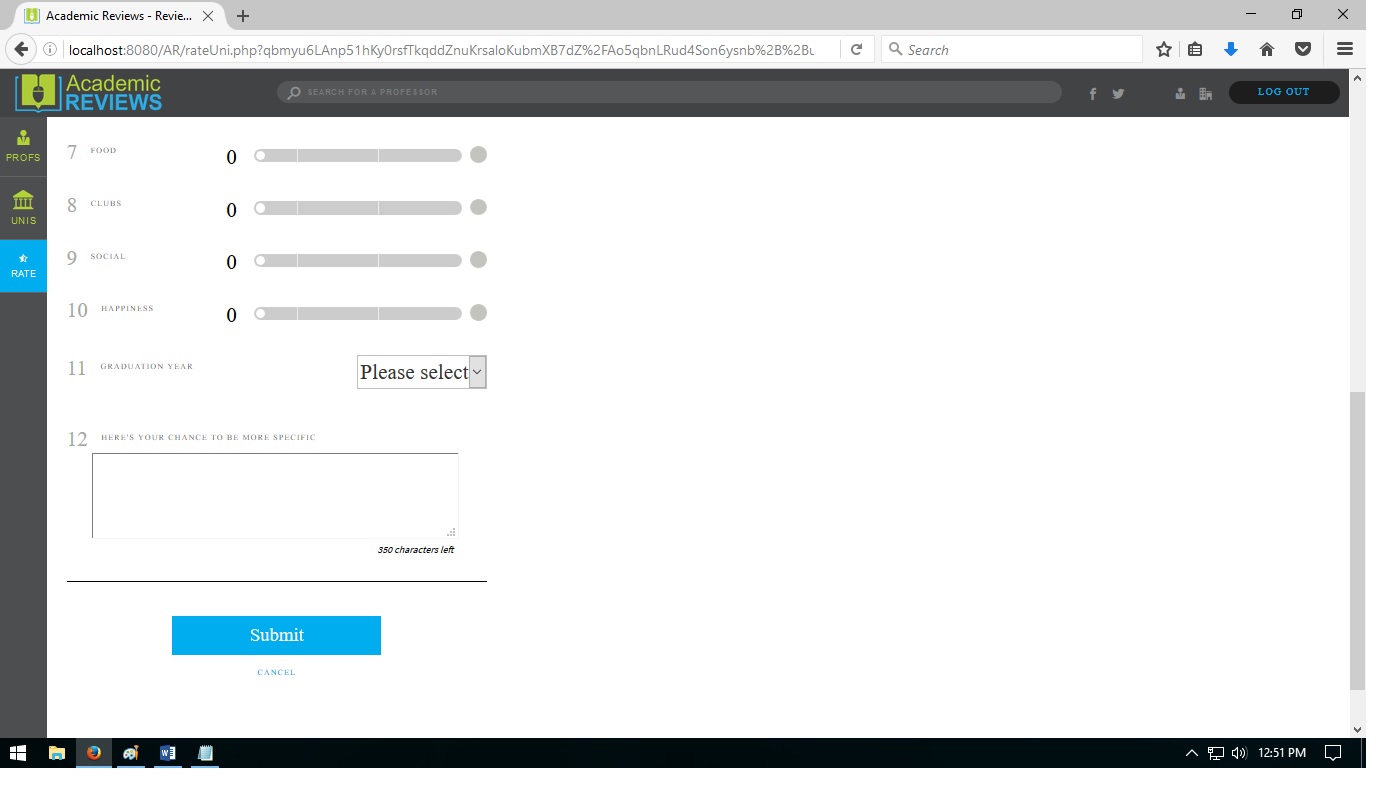
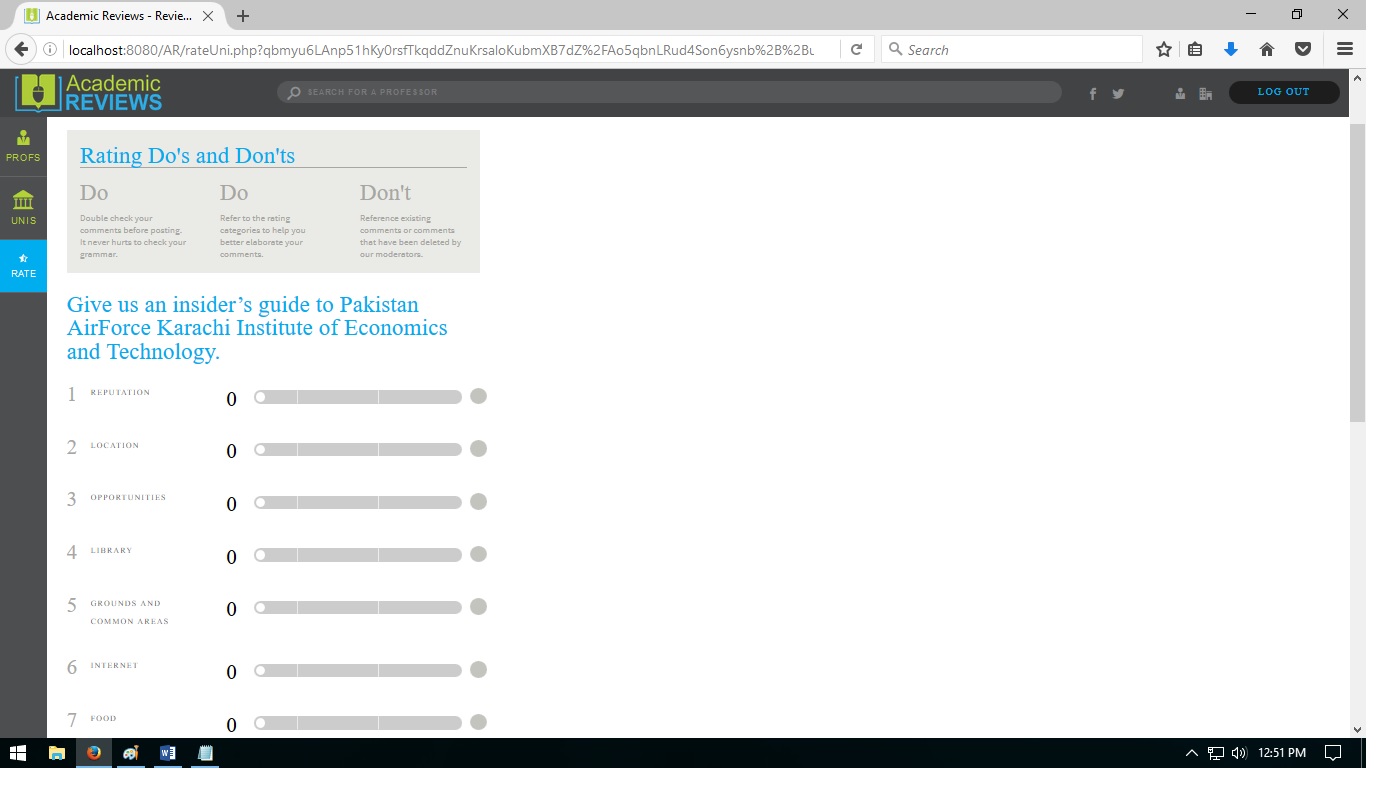
## University Rating:



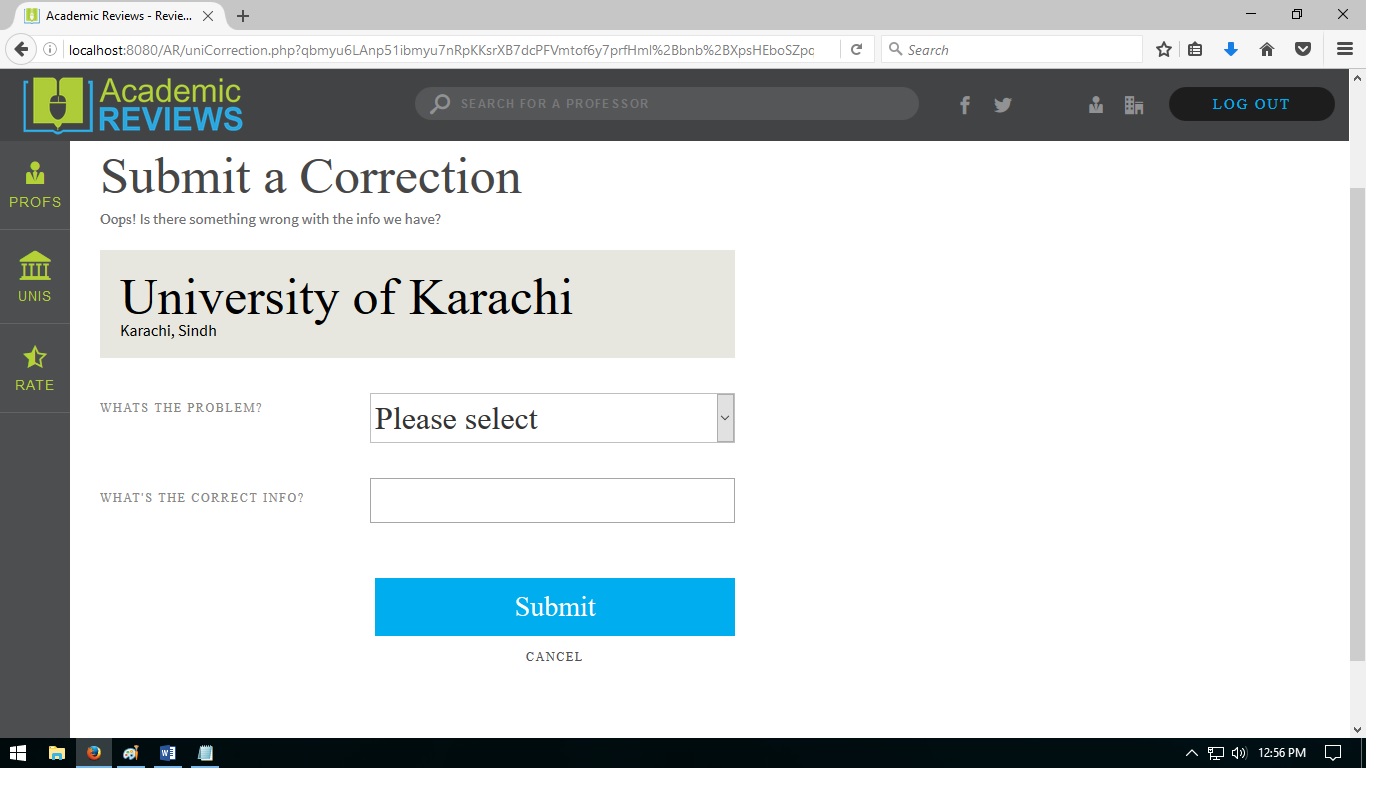
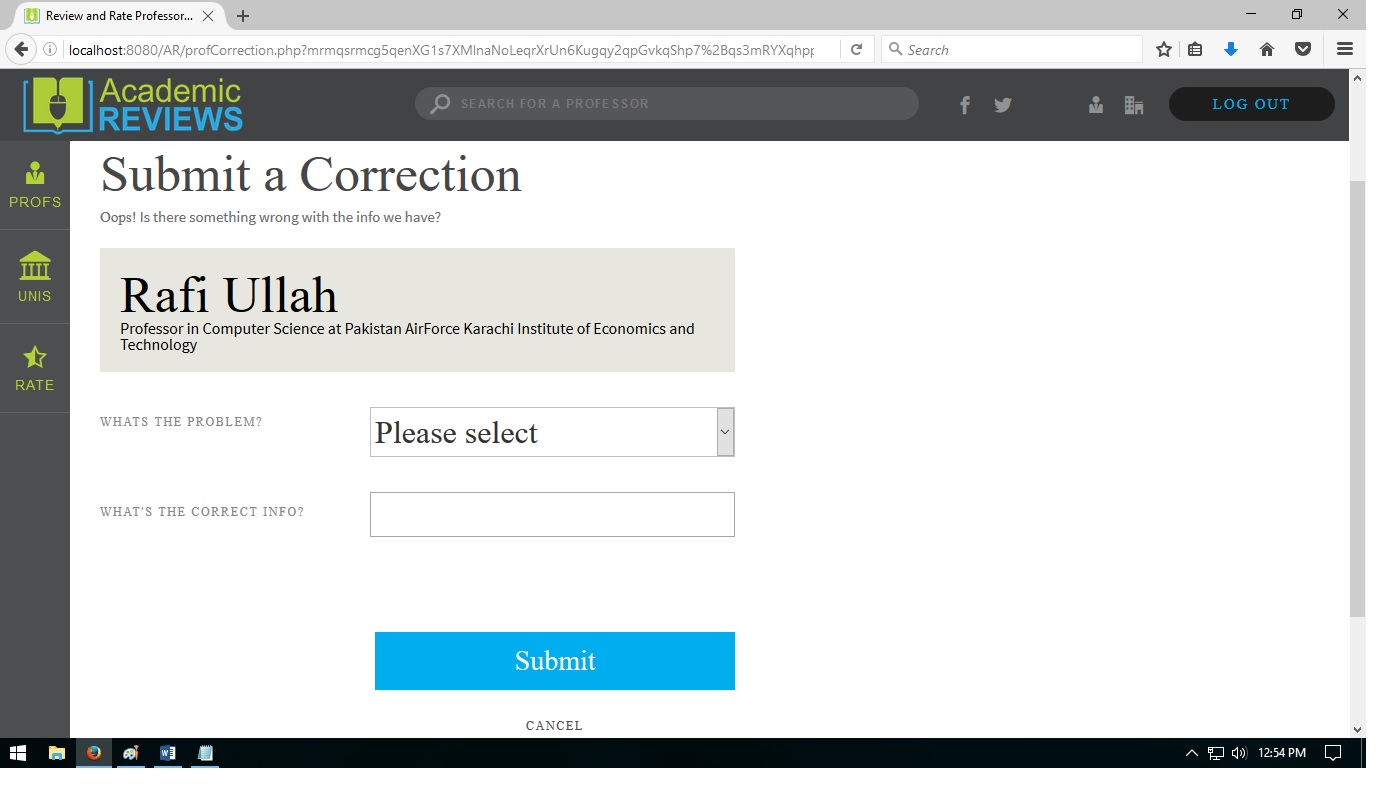
## Rate Professor:



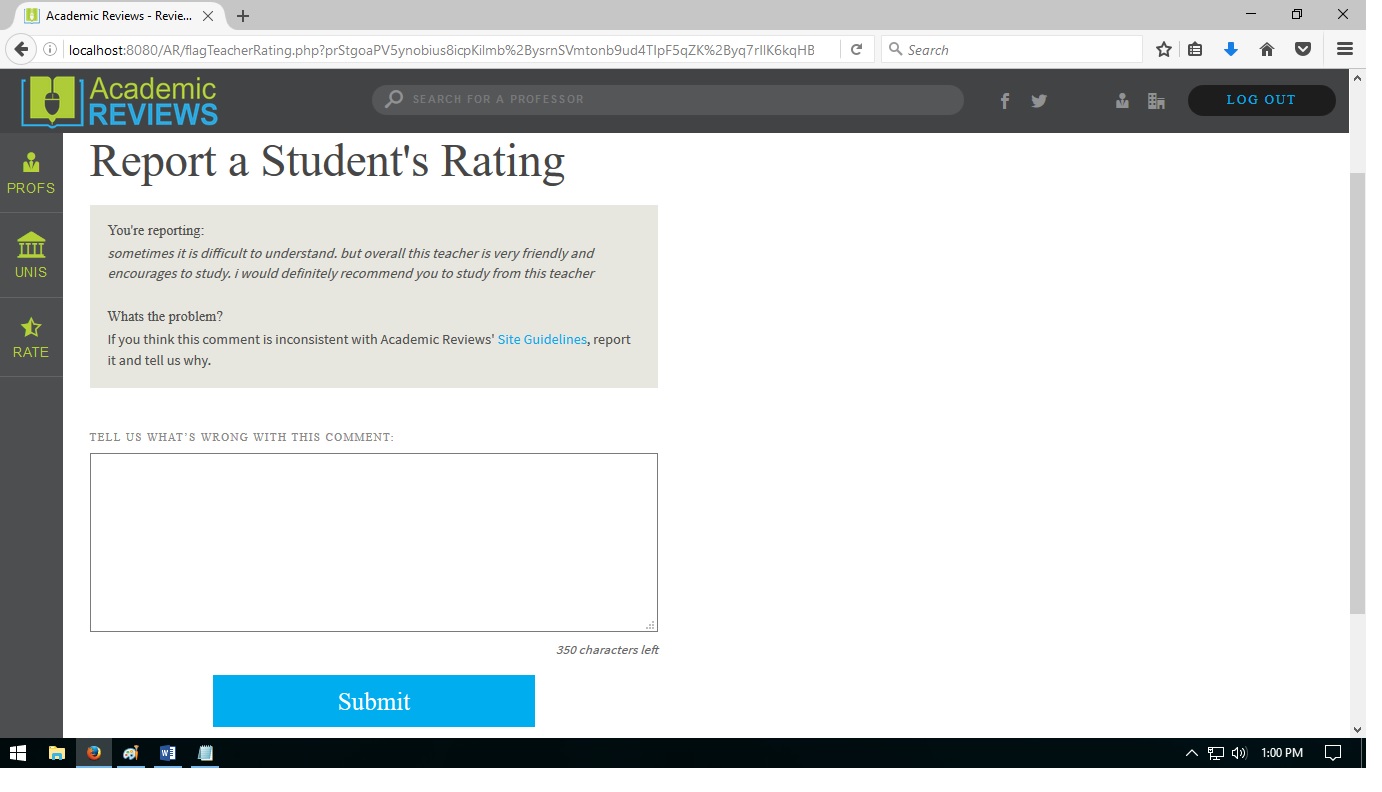
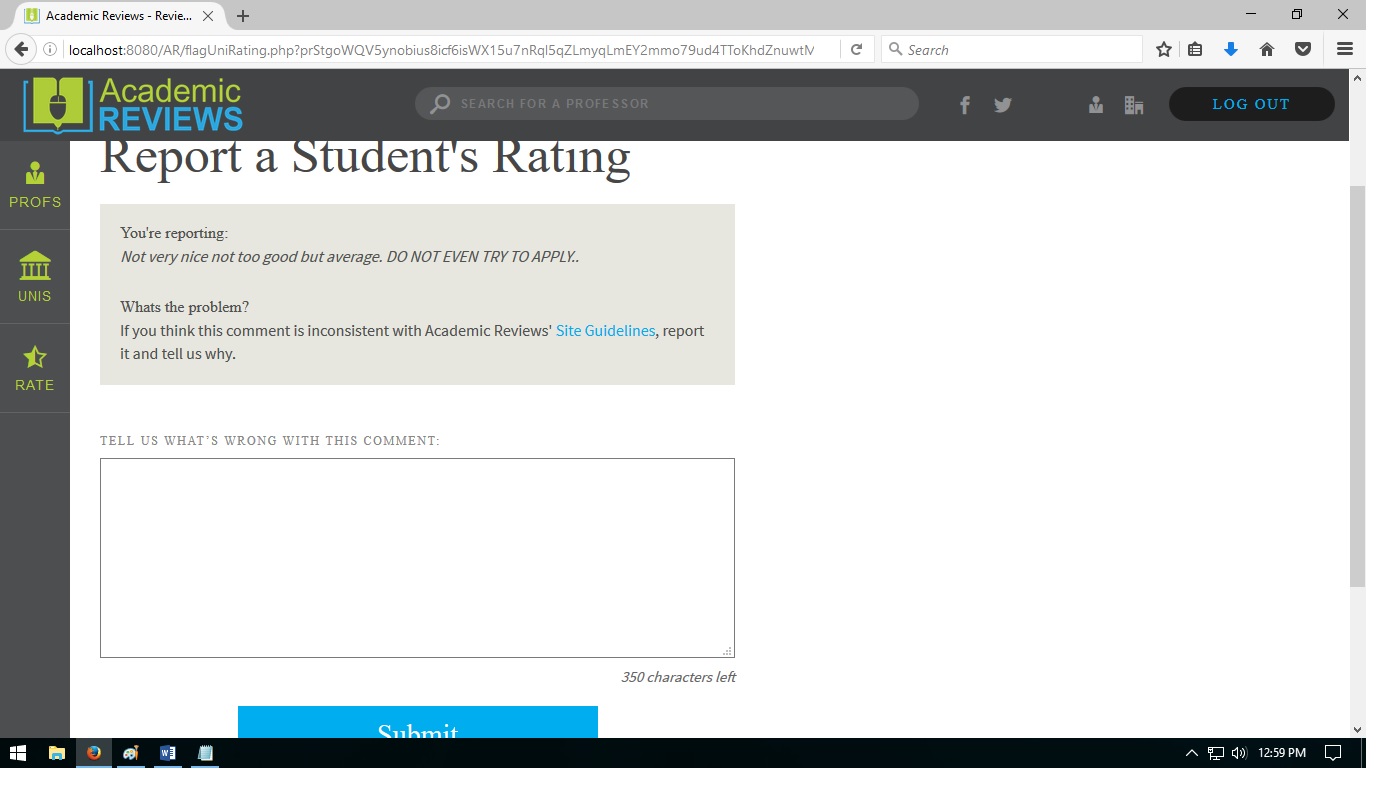
## Rate University:



## Submit Correction:



## Report Rating:



Appendix E – Team Members

**Naifish Nizar Ali** (September 9, 1992): [naifish\_momin@live.com](mailto:naifish_momin@live.com)

**Muhammad Faiz** (May 22, 1987): [faiz.yousuf@hotmail.com](mailto:naifish_momin@live.com)

Annexure II – Software Planning &Costing

We have implemented this through Constructive Cost Model (COCOMO) a method for evaluating and/or estimating the cost of software development.

# Annexure III – Test Case

# **Test Case 1.1: User Login to the website**

**Purpose**: Facebook integration working correctly

**Pre-requisite**: None

**Test Data**:

user\_email = {text}

user\_pass = {text}

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s work, education and share permissions.

**Note**: Users cannot login to our website until he/she provide a valid Facebook login credentials.

# **Test Case 1.2: Rate a professor**

**Purpose**: To check that only a authorized user can rate the professor.

**Pre-requisite**: User must be logged in

**Test Data**:

course\_code={text}

helpfullness={text}

clarity={text}

easiness={text}

comment={text}

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s work, education and share permissions.
4. Search the professor
5. Click on rate this professor button
6. Fill the required fields
7. Submit the rating

**Note**: If user does not belong to the professor then he/she will not able to rate that professor.

# **Test Case 1.3: Rate a university**

**Purpose**: To check that only a authorized user can rate the professor.

**Pre-requisite**: User must be logged in.

**Test Data**:

reputation={text}

location={text}

opportunities={text}

…….

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s work, education and share permissions.
4. Search the university
5. Click on rate this university button
6. Fill the required fields
7. Submit the rating

**Note**: If user does not belong to the university then he/she will not able to rate that university.

# **Test Case 1.4: Flag student rating**

**Purpose**: To check that functionality of reporting false rating is working or not.

**Pre-requisite**: user must be logged in and there must be a rating to report.

**Test Data**:

commnet={text}

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s work, education and share permissions.
4. Search the professor
5. Choose that which rating is to report
6. Fill the required field
7. Submit the report

**Note**: If user does not belong to that professor than he/she cannot report any rating on that professor.

# **Test Case 1.5: Share professor or university stats**

**Purpose**: To check if the shared stats is actually get visible on the social sites or not.

**Pre-requisite**: user must be logged in.

**Test Data**:

Click={Share on Facebook button }

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s share permissions.
4. Search the professor
5. Click on Share button

**Note**: If user is not logged in then that user will not able to share professor or university stats on social websites.

# **Test Case 1.6: Correct university info**

**Purpose**: To check that if the functionality of correcting university info is working correctly or not.

**Pre-requisite**: user must be logged in.

**Test Data**:

Correction\_type ={text}

Correction ={text}

**Steps**: Steps to carry out the test are following:

1. Visit Login page
2. Click on login with Facebook button
3. Allow access to your Facebook’s work, education and share permissions.
4. Search the university
5. Click on submit correction button
6. Fill the required fields
7. Submit the correction.

**Note**: If user does not belong to that university than that user will not able to provide the correct info of that university.

# **Test Case 1.7: Contact us**

**Purpose**: To check if we are getting user quarries or not.

**Pre-requisite**: None.

**Test Data**:

quarries\_type={text}

quarries={text}

**Steps**: Steps to carry out the test are following:

1. Visit Contact Us page.
2. Fill the required fields.
3. Click on send button

**Note**: User must have to provide a correct email address so we can discuss and solve their quarries.

# End