**University of Central Missouri**

**Department of Mathematics & Computer Science**

**Fall 2015**

#### 

#### PROJECT

#### On

#### Course Registration System

#### Submitted

#### By

**Manoj Kumar Potlapalli [700635028]**

**Sandeep Reddy Aredla [700635578]**

**Ravali Gujja [700641004]**

**Himabindu Chittimalla [700641264]**

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER** | | | | | **NAME** | **PAGE NO.** |
| 1 | | | | | INTRODUCTION | 4 |
|  | 1.1 | | | | Problem Statement | 4 |
| 1.2 | | | | Existing System | 4 |
|  | 1.3 | | | | Proposed System | 4 |
|  | 1.4 | | | | Scope | 5 |
| 2 | | | | | REQUIREMENTS | 6 |
|  | | 2.1 | | | Introduction | 6 |
| 2.2 | | | System Features | 8 |
|  | | 2.3 | | | Functional Requirements | 8 |
|  | | 2.4 | | | Non Functional Requirements | 10 |
| 3 | | | | | ANALYSIS | 11 |
| 4 | | | | | SOFTWARE PROJECT PLAN | 15 |
| 5 | | | | | DESIGN | 17 |
|  | | | 5.1,5.2 | | Use case diagram | 19,20 |
|  | | | 5.3 | | Class Diagram | 22 |
|  | | | 5.4 | | Activity Diagram | 23 |
|  | | | 5.5 | | Sequence Diagram | 24 |
|  | | | 5.6 | | Collaboration Diagram | 25 |
| 6 | | | | | IMPLEMENTATION | 26 |
| 7 | | | | | TESTING | 31 |
| 8 | | | | | Screen Shots | 34 |
| 9 | | | | | CONTRIBUTIONS | 38 |
| 10 | | | |  | REFERENCES | 39 |
| 10.1 | Tools and Technologies |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **DESCRIPTION** | **PAGE NO.** |
| 5.1,5.2 | Use Case Diagrams | 19,20 |
| 5.3 | Class Diagram | 22 |
| 5.4 | Activity Diagram | 23 |
| 5.5 | Sequence Diagram | 24 |
| 5.6 | Collaboration Diagram | 25 |

**1. INTRODUCTION**

## 1.1Problem Statement:

Internet has become popular now-a-days and used in our daily life. The days of paper registration forms and advertising is declining day by day and the traditional way of managing courses is time consuming. Many colleges do not have a website which allows students to register for the courses provided by the college and some colleges have a comprehensive website which deals with every information related to the college and making it complicated , students often feel confused with heavy information and face server problems due to heavy traffic while accessing the website for registering the courses. In serving the solution to this problem the proposed system is developed.

## 1.2 Existing System:

Whenever Course Registrations start for the semester, Students of every major tries to register for the courses in the same Comprehensive Website (eg-My central). This results in heavy traffic to the website and results in developing server problems and interrupting other activities provided by the website it also degrades the performance of the website and whenever an issue such as course cancellation or change of timings arises, the administrator of the website should carefully address and care should be taken such that no other activity is interrupted.

## 1.3 Proposed System:

A website which is completely dedicated for registration allows students to register for the courses swiftly. The proposed website contains all the details about the three important Majors of the college namely Computer Science, Computer Information Systems and Industrial Management. It provides the details about the Courses provided by them in Spring 2016 semester along with the information of all the lecturers and the timings of their proposed classes and the registration module. A database is created and maintained wherein the data regarding the Lecturers of different Courses, their contact details are kept. Students can simply browse along the website to see different Courses and register for the interested Course. The Students and the lecturers are intimated through Confirmation email whenever a student registers to their Course. The proposed system also reduces the work load on the administrator of the website as he should only care about the registrations and corresponding database.

**Advantages:**

1. Students can have access to their desired courses of corresponding Majors. The major advantage of the system is the manipulation of data is easy for the admin as it only serves few activities.
2. Students get Confirmation Emails after Successful Registrations and they can ask queries related to the courses. No paper work is required for registration.
3. In this proposed system once a student gets registered he cannot register the same course again but he can register two other courses.

**1.4 Scope:**

The goal of this project is to allow the users to easily navigate through the web application and get registered to their interested Courses as quickly as possible. It provides the users with an interactive interface which responds to user inputs.

# 2. REQUIREMENTS

## 2.1 INTRODUCTION:

The product described here is **COURSE REGISTRATION SYSTEM** web application. The aim of creating this application is to provide the users a place where information related to different courses offered by different departments and lecturers in the university and these are available to be viewed by all the students and get registered in their interested course. It also allows the user to send their queries regarding the courses.

### 2.1.1 Product Scope

The objective of this project is to allow the users to quickly navigate through the website, find out the number of different courses that are offered in the college for the spring 2016 semester and get registered to them.

### 2.1.2Product Functions

The major functionalities of this website will be as follows:

* Drop down list.
* Quick Registration.
* Confirmation Email.
* User query.

### 2.1.3Development Environment:

Software and hardware required to develop this project are mentioned below.

*3.1.3.1* ***Software Requirements:***

|  |  |
| --- | --- |
| Programming Platform : | HTML5,CSS,JavaScript,PHP5.4 |
| Editor : | Notepad++/Edit+ |
| Database : | MySQL 5.5 |
| Web Server : | Apache Http Server 2.4 |

2.1.3.2Hardware requirements:

|  |  |
| --- | --- |
| RAM : | 512MB or high |
| Processor : | INTEL-P4 or higher |
| Hard disk : | 40GB or more |

### 2.1.4 Operating Environment

## Environment in which this product operates is described here.

#### 2.1.4.1 Client Side Requirements

##### 2.1.4.1.1Software requirements:

|  |  |
| --- | --- |
| Operating System : | Windows XP or Advanced |
| Web Browser : | Chrome or Mozilla |

##### 2.1.4.1.2Hardware requirements:

|  |  |
| --- | --- |
| Hard disk : | NA |
| RAM : | 512MB or higher |
| Processor : | Pentium 4 or Advanced |

#### 2.1.4.2 Server Side Requirements

##### 2.1.4.2.1Software requirements

|  |  |
| --- | --- |
| Operating System : | Windows XP or advanced |
| Web Server : | Apache Http Server |
| Database : | MySQL 5.5 |

##### 2.1.4.2.2Hardware requirements:

|  |  |
| --- | --- |
| Hard disk : | 40GB or more |
| RAM : | 1 GB or higher |
| Processor : | INTEL-P4 or advanced |

## 2.2 SYSTEM FEATURES

The major functionalities of this project will be as follows:

* Drop down List
* Quick Registration
* Confirmation Email
* User Query

### 2.2.1Drop down List:

It helps the user to find all the subjects available for spring 2016 in different majors and by different lecturers organized in a list manner. This is a high priority feature as it allows the user to find what he is looking for on the home page itself.

#### 2.2.1.1Functional Requirements

The application allows the user to hover the mouse over the ***Major*** tab of the navigation bar. And whenever the mouse is hovered on it, it triggers a JavaScript event that shows a drop down list. From this list, the user need to again hover over the given set of options in the list and then again another list appears. Here the user needs to click on the interested course to get navigated to information page of that course and then to the registration page.

### 2.2.2Quick Registration:

The user can register to any given course under a particular major in the website. User can quickly navigate through the course list to find out his interesting course. On the registration page, the user can quickly register by submitting his details along with the course name lecturer name and time.

#### 2.2.2.1 Functional Requirements:

### This web application allows the user to submit their details in order to get registered to the desired courses. The user needs to enter correct information for a successful registration. The system checks for the falsity of the entered data. If the user enters wrong details, then the application triggers an error message showing what the user has done wrong.

In case if the current user has already registered for a particular course and trying to register to it again then the application redirects him to a page where it tells the user that he is already registered to the course and also encourages the user to register to other courses by providing a link to registration page.

### 2.2.3Confirmation Email:

As soon as the user enters the correct information and submits the details, a confirmation email is sent to the students and lecturers email address. This is a high priority feature as the student needs to show the confirmation email to the university if needed for authentication purpose. And the lecturers have the list of the students under their course with the confirmation mail.

**2.2.3.1 Functional Requirements:**

The application allows the user to enter their details in a form, where the user needs to enter the correct email address. This is the email address where a confirmation email is sent on successful registration. So whenever a user clicks submit button in the form, a php code for sending email is triggered and also the details get entered into the database.

**2.2.4 User Query:**

This is a special feature for the user where he can post his queries regarding any information related to the courses and the admin responds to the user on their email address. There is a query section at the end of each web page of the application.

**2.2.4.1 Functional Requirements:**

The user (student) can use this feature for submitting their queries to the admin. They need to attach the email address along with their query before submitting it. After hitting the submit button, the query is sent to the admin’s email address along with the users email address in the message and the admin will respond to them on the email address which the user has provided.

## 2.3 NONFUNCTIONAL REQUIREMENTS

### 2.3.1. Performance Requirements

This will be a web based product and so it has to be run from a web browser. The product will take time to load depending upon the internet connection speed which also depends on the media from which the application is run. The performance of the application will depend upon hardware components of the client/user.

### 2.3.2. Safety Requirements

The application’s database is maintained by Administrator who is responsible to secure the database and also to respond to the user queries. The access of unauthorized users may result in loss or damage of data. So the access to the database is restricted.

### 2.3.3. Security Requirements

### The information given in by the user is directed to the database after submission and it is not visible to any other users. Only administrator can view the user information.

* Administrator is provided with secure login, password thus database will be secure.
* The system’s back-end servers will not be visible or accessible to the users.

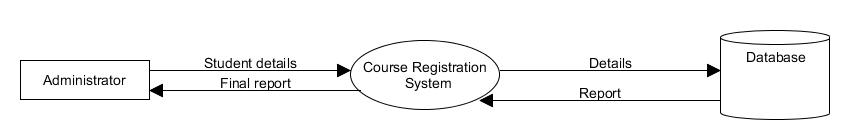
### 

**3 ANALYSIS**

The main aim of analysis phase to understand the requirements of the client by analyzing and refining them for more deeper understanding for the development of the software. The output of the analysis work flow should be more precise so that the design & implementation phases are smoothly done. For the current project the structured systems analysis is used. This analysis is a nine step process in order to analyze the client requirements.

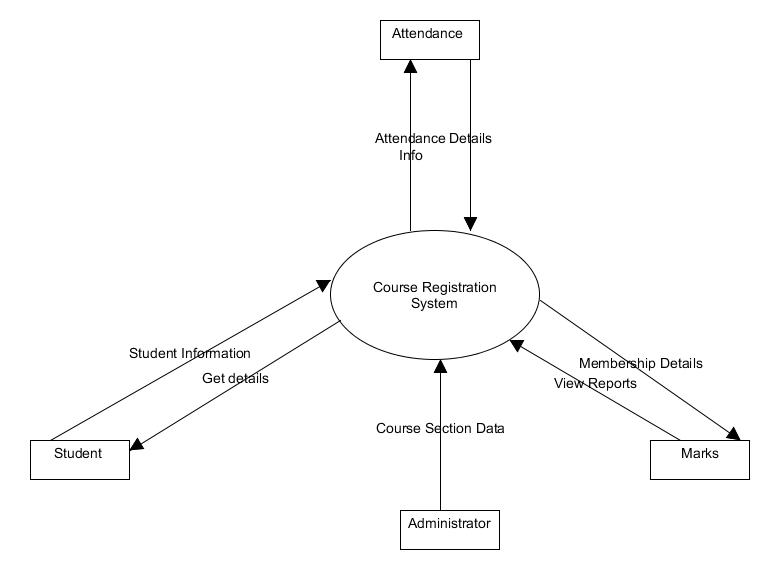
**Step 1: Drawing the Data Flow Diagrams:** The following are the two levels of data flow diagrams used to analyze the requirements more precisely.

**Data Flow Diagram Level 0**



**Fig: 3.1 Data Flow Diagram Level 0 of Course Registration System**

**Data Flow Diagram Level 1**



**Fig: 3.2 Data Flow Diagram Level 1 of Course Registration System**

**Step 2:**

**Sections to computerize:** For this project the entire operations of the system are computerized.

**Step 3:**

**Determine the details of the Data Flows:** The details to be determined are as follows,

Student details

Course details

**Data Dictionary:**

|  |  |  |
| --- | --- | --- |
| Data Element | Description | Narrative |
| Student details | Name  Roll No  Email-id | unique details of the student |
| Course details | Subject Name  Course Timings  Professor Name | receives email to the student and respective professor for each selected course |

**Step 4: Logic of the process Definition:**

The details of the users is validated whenever a request is placed. The course details are entered, stored, retrieved whenever a user request(s) to enroll a subject(s).

**Step 5: Define the Data stores:** The data stores Student details and Course details are defined in this step.

**Step 6: Define the Physical Resources:** The data stores are stored on the hard disk and no cloud storages are defined.

**Step 7: Determine the Input/Output Specifications:** The input will be the entering the student details, choosing the courses, submitting the details in the google forms and selecting the professors. The output will be the received emails and getting registered for the courses.

**Step 8: Perform Sizing:** The storage requirements are based on the hard disk in the system.

**Step 9:**

Pentium based systems with a minimum of P4

512MB RAM (minimum)

1. **SOFTWARE PROJECT PLAN**

**Project Summary :**

The main purpose of the project is to allow the student to register for the courses without any difficulties. The university enables the student to enroll the courses without any server issues by providing a separate website for each department.

**4.2: Assumptions & Constraints:**

The application should be user friendly, the time and budget are to be met & the architecture of the system should be easily changeable for future modifications if needed.

**4.3: Project Deliverables:**

The entire application of this project will be delivered within a duration of 7 weeks.

**4.4 Project Organization.**

**4.4.1 External Structure:** The entire project is equally distributed & discussed every week.

**4.4.2 Internal Structure:** The team consists of four members.

**4.4.3 Roles and Responsibilities:**

**Sandeep Reddy Aredla [700635578]:**

- Worked on Design phase

- Documented Requirements

**Manoj Kumar Potlapalli [700635028]**

**-**Worked on Implementation

**-**Documented Requirements

**Ravali Gujja [700641004]**

-Documented Requirements

-Documented Introduction

**Himabindu Chittimalla[700641264]**

-Worked on Analysis, Testing

-Documented Requirements

**4.5 Work Plan:**

**Work Activities and Schedule Allocation:**

**Week 1:** The requirements are determined and completed successfully.

**Weeks 2, 3:** The analysis of the project is completed in a week and discussed with the team. The following week the project plan was developed & was approved by the team.

**Weeks 4, 5:** By the end of the fifth week the design phase is completed.

**Weeks 6, 7:** The last two weeks plan includes the implementation phase and ended with the testing along with the complete documentation.

**4.6 Resource Allocation:** Each person in the group worked equally in the assigned modules in order to meet the deadline of the project.

**4.7 Control Plan:** Any major changes that effect the schedule of the project are discussed with the team members.

**4.8 Risk Management Plan:** As the project is for the college the users have the computer knowledge so the risk factor is less as they have the idea of selecting the courses.

**4.9 Supporting Process Plan:**

**4.9.1 Configuration Management Plan:** Not applicable here.

**4.9.2 Testing Plan:** The entire process follows the testing work flow.

**4.9.3 Documentation Plan:** Documentation will be delivered according to the specifications.

**4.9.4 Quality Assurance Plan and Reviews and Audits Plan:** The entire code will be tested by the testing team.

**4.10 Additional Plans:**

**Maintenance:** 6 months maintenance will be provided.

1. **DESIGN**

**5.1, 5.2 Use case Diagram:** A use case diagram is a graphic depiction of the interactions among the elements of a system. The Course Registration System has two actors - **user and the admin.** The following use case represents their interaction with the system and the functionality that they have access to the Admin.

**5.1: The admin can perform the following functions:**

- login

- update source files

- run queries

- manipulate data

- logout

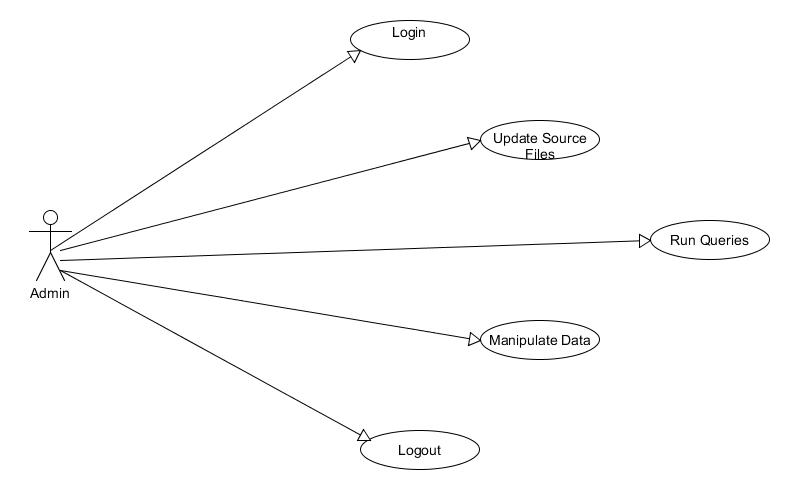
**5.2: The user has the following functions:**

- courses

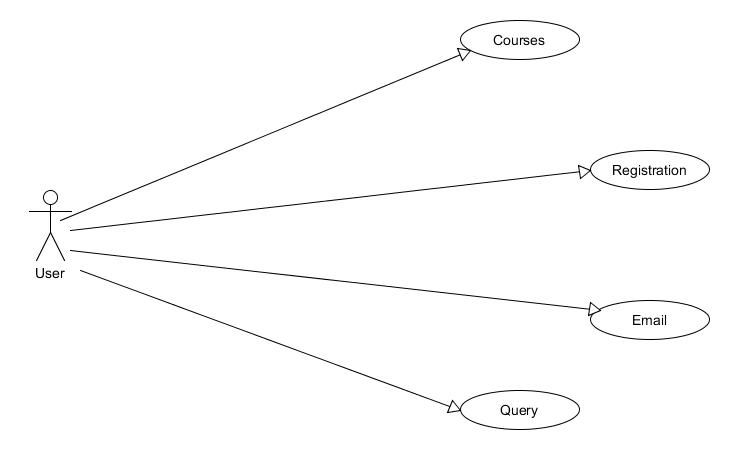
- registration

- query

-email

****

**Fig 5.1: Use case diagram for Admin**

****

**Fig 5.2: Use case diagram for User**

**5.3 Class Diagram:**

The class diagram denotes to the major classes we will execute and their association with one another. We have attempted to keep low coupling between the modules so changes made in one module won't influence the others. Additionally, every module is in charge of free errand like registering the courses.

**Browser:** It is a boundary class. It contains the attribute namely web address and includes operations like open() and close().

**User**: It is a entity class. It manages the entity classes like register, email and query classes.

**Admin:** It is also an entity class which includes attributes like User Name and Password and manages the entity classes like ResponseEmail and RunQuery.

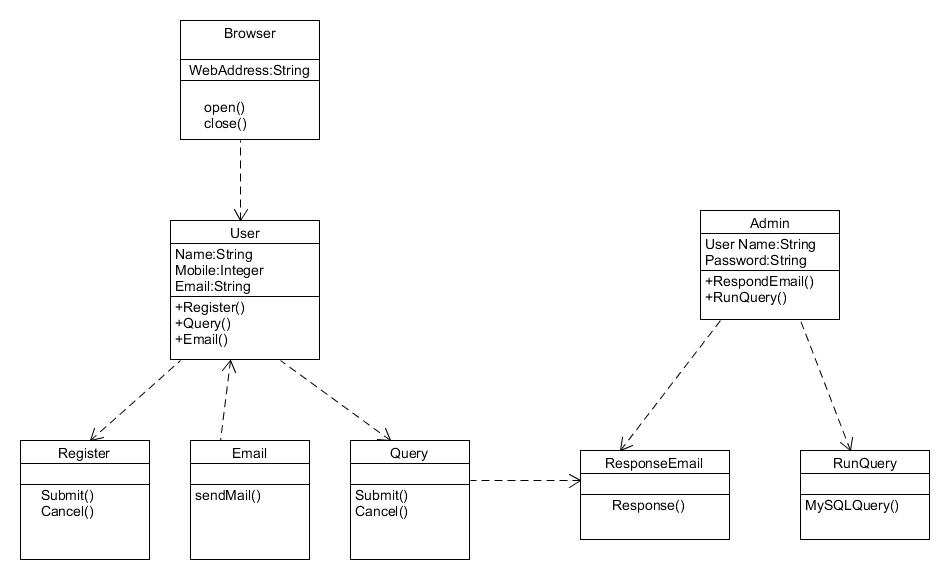
**Register:** It helps the user to register the courses and includes the operations like submit() and cancel().

**Email:** It generates an email to the student and from the professor when he/she registers for the course.

**Query:** It provides the user to submit the queries to clarify his/her doubts.

**ResponseEmail:** A email will be generated from the admin when the query is posted and from the professor when his course is selected.

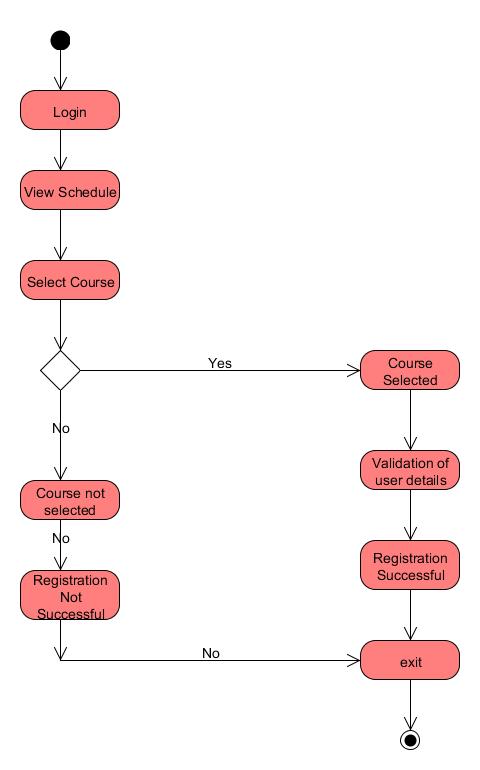
**RunQuery:** An automatic response will be generated when the query is submitted and includes the operations like MySQLQuery() which retrieves the data from the database.

****

**Fig 5.3: Class diagram**

**5.4 Activity Diagram:**

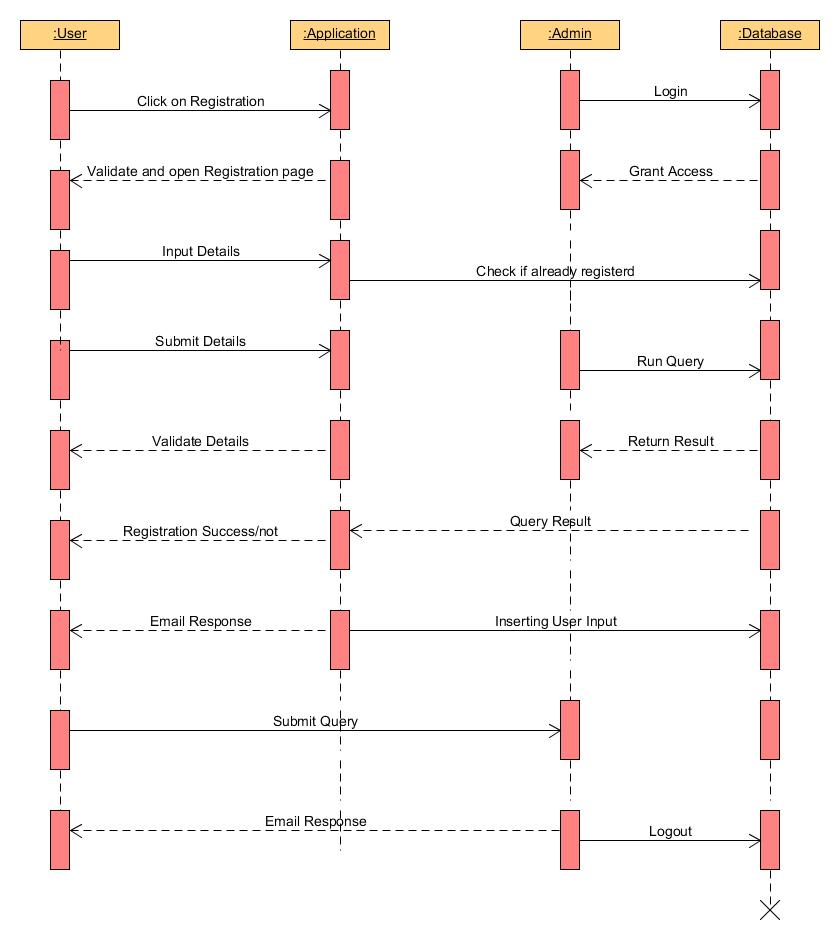
An activity diagram is used to display the sequence of activities in the system. We have distinguished different work processes in the framework and represented them from the begin point to the completion point enumerating all the choice ways that exist in the movement of occasions contained in the activity.

****

**Fig 5.4: Activity diagram**

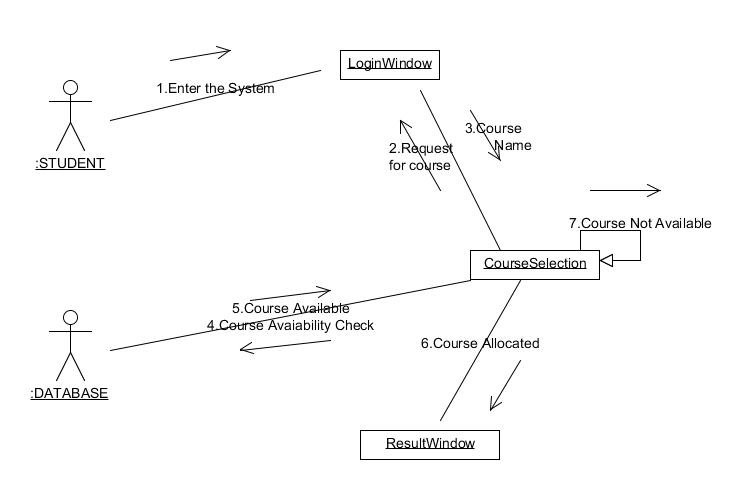
**5.5 Sequence Diagram:**

The sequence diagram shows how processes operate with one another and in a particular order. In the following diagram, we have explained the object interactions arranged in time sequence. It denotes the objects and classes involved in each of the scenarios in the system and the sequence of messages exchanged between the objects in this system.

****

**Fig 5.5: Sequence diagram**

**5.6 Collaboration Diagram (Course Registration work-flow):** The course registration work-flow is the major component in our system and is represented in the following collaboration diagram.

****

**Fig 5.6: Collaboration diagram**

1. **IMPLEMENTATION**

The project is developed using HTML CSS Java Script and PHP language .

**Sample Source code of Registration.php file implemented in our code:**

<!DOCTYPE html>

<html>

<head>

<style>

.success

{

color:black;

font-size:1em;

border:1px solid green;

padding:6px;

text-align:center;

font-weight:bold;

}

.warning

{

color:black;

font-size:1em;

border:1px solid red;

padding:6px;

text-align:center;

font-weight:bold;

}

</style>

</head>

</html>

<?php

//get variables from the form

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

$mobile = $\_POST['phone'];

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

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

$lecturer = $\_POST['lecturers'];

$class\_time = $\_POST['class\_time'];

require\_once('/php/connection.php'); //connect to database

if($connection)

{

//connected to the database. Perform required operations.

//get number of rows with same email and name, if more than 3 then say only 3 times they can register.

$get\_data = mysqli\_query($connection,"SELECT \* FROM registrations where name='$name' AND email='$email'");

$num\_of\_registrations = mysqli\_num\_rows($get\_data);

if($num\_of\_registrations>=3)

{

echo '<div class="warning">Quota exceeded. You are allowed to register to only 3 courses per semister.</div>';

}

else

{

//insert the data into the database and send the emails to the lecturer and student.

$insert\_data = mysqli\_query($connection,"INSERT INTO registrations VALUES('','$name','$mobile','$email','$course','$lecturer','$class\_time')");

if($insert\_data)

{

//echo '<div class="success">Data inserted.</div>';

//get email id of lecturer from lecturer table and send email to lecturer and user.

$get\_lecturer\_email = "SELECT lecturer\_email FROM lecturers where lecturer\_name = '$lecturer'";

$result = mysqli\_query($connection, $get\_lecturer\_email);

if (mysqli\_num\_rows($result) > 0)

{

// output data of each row

while($row = mysqli\_fetch\_assoc($result))

{

$send\_email\_to = $row['lecturer\_email'];

$message = $name." with ROLL Number ".$mobile." has registered your class, @ ".$class\_time;

$subject ="Registration successful";

$header = "FROM: segrp6@noreply.com";

$send = mail($send\_email\_to,$subject,$message,$header);

if($send)

{

//email sent to professor.

//echo "Thanks ".$name." for registering to my class, <br/> Regards, ".$lecturer.".";

$to\_address=$email;

$to\_subject="REGISTRATION SUCESSFULL";

$message = "Hey ".$name.". You have successfully enrolled to ".$course. " for Spring 2016 semester. The course work starts on january 11th of 2016.";

$header="FROM:segrp6@noreply.com";

$sent\_to\_student = mail($to\_address,$to\_subject,$message,$header);

echo '<div class="success">Thanks '.$name.' for registering to my class.<br /> Regards, '.$lecturer.'.</div>

break;

}

else

{

//Failed to send the email.

echo '<div class="warning">Email was not sent to '.$send\_email\_to.'</div>';

break;

}

}

}

else

{

echo '<div class="warning">Lecturer email not found in the Database</div>';

}

}

else

{

//problem with query. QUERY DID NOT RUN

echo '<div class="warning">OOPS! Something went wrong. Could not register. Please try again, After sometime.</div>';

}

}

}

else

{

echo '<div class="warning">Failed to connect to db.</div>';

}

?>

**7. TESTING**

**7.1 System Testing:**

System Testing is a kind of black box testing. It is performed as the final test to verify that the system meets the specifications and its purpose The main focus of this testing is to evaluate End-user requirements.

**7.2 Test Environment:**

Operating System: Windows 7

RAM : 4GB

Web Browser : Chrome/Mozilla Firefox

Technical Software: XAMPP

## 7.3 Test Cases:

## *Test Case: 1*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Id | Case Type | Description | Expected Value | Actual value | Result |
| 1 | Checking the internet connection | We need to go to the network connection | Internet connected | Internet connected Successfully | Internet connected Successfully |
| 2 | Unable to start the internet service | Internet is not getting connected due to some network problems | Error in connecting to internet | Internet will not get connected | Internet is not connected. Checking the network connection. |

***Test Case: 2***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Id | Case Type | Description | Expected value | Actual value | Result |
| 1 | Giving proper credentials by user while registering | all the details of the user should be entered correctly | All the details of the user are correct | User should check all the details | User will get registered after checking the details |
| 2 | Not giving correct credentials by user while registering | Username ,number and all user details are not entered correctly | All the details are correct | User didnot check all the details | User will not be able to register because of wrong credentials |

***Test Case: 3***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case id | Case Type | Description | Expected Value | Actual value | Result |
| 1 | Registering before or after deadline | Clicked button before or after deadline | Before deadline | Allow Registration | Open Registration Page |
| 2 | Registering before or after deadline | Clicked button before or after deadline | After deadline | Don’t Allow Registration | Show Time out page |

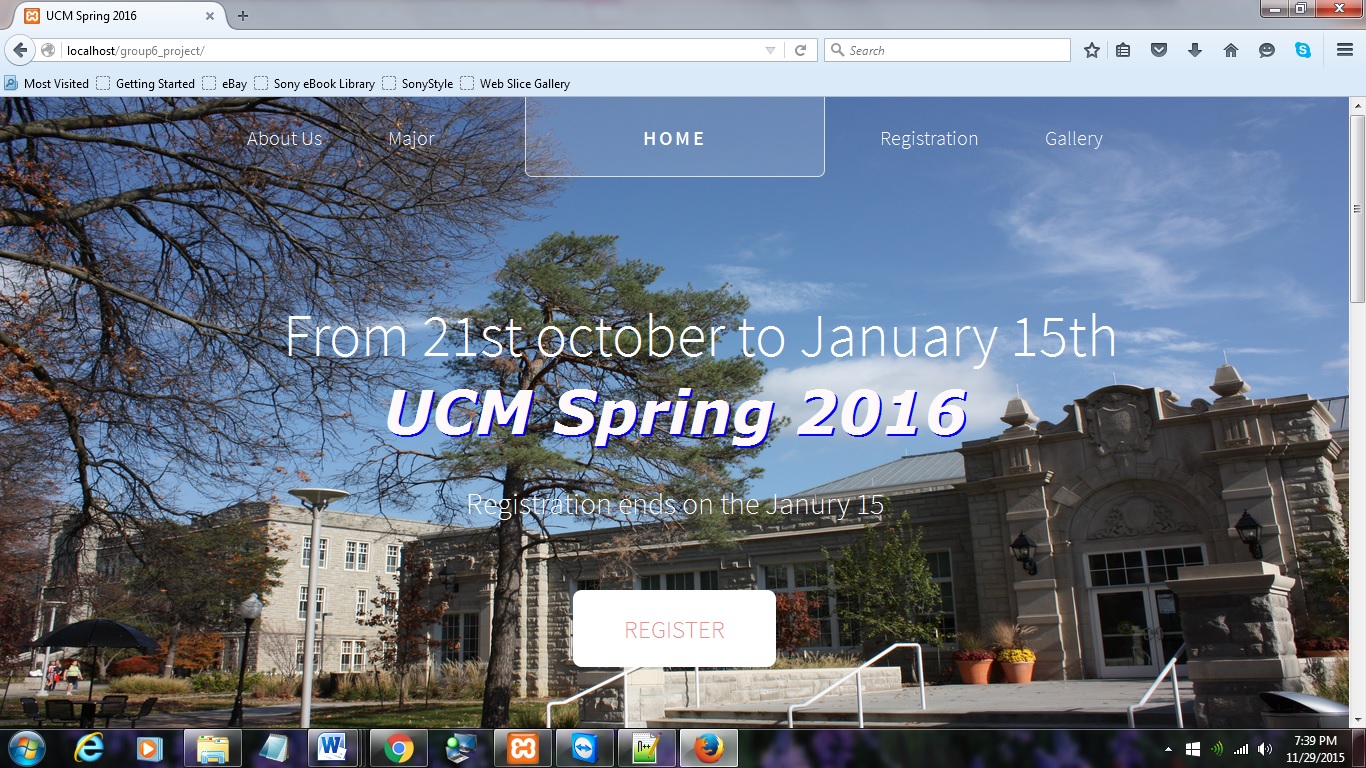
***Test Case: 4***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case id | Case Type | Description | Expected Value | Actual value | Result |
| 1 | Correct Email or Not | The user can send Query to admin | Send Query to admin | An email is sent to admin | Email sent successfully |
| 2 | Correct Email or Not | The user can send Query to admin | Show a warning | A warning is displayed | The user is warned for wrong email. |

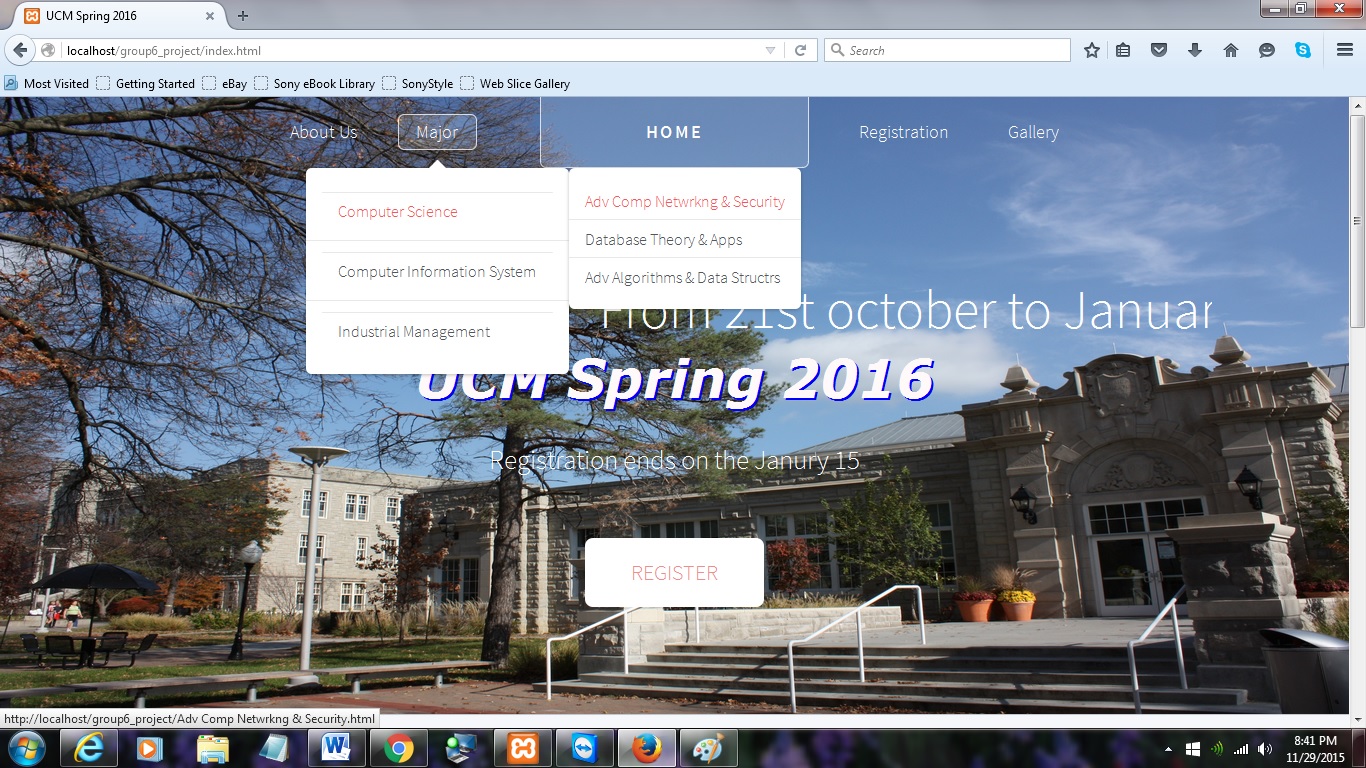
**Test Results:**

The tests were conducted on the web application on the different major web Browsers such as Mozilla Firefox and Google Chrome to find out if the application is working according to the specifications made. The bugs were found and removed during various stages of development.

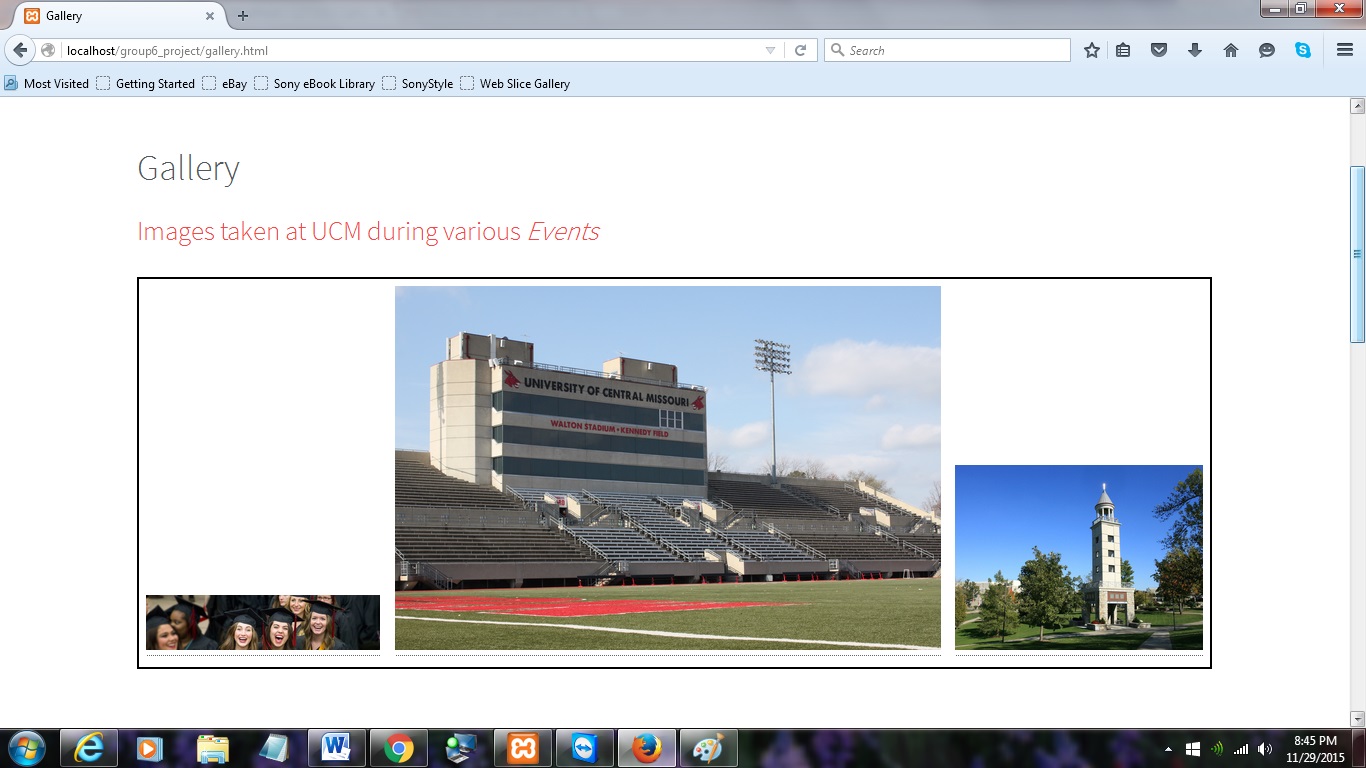
**8. SCREEN SHOTS**



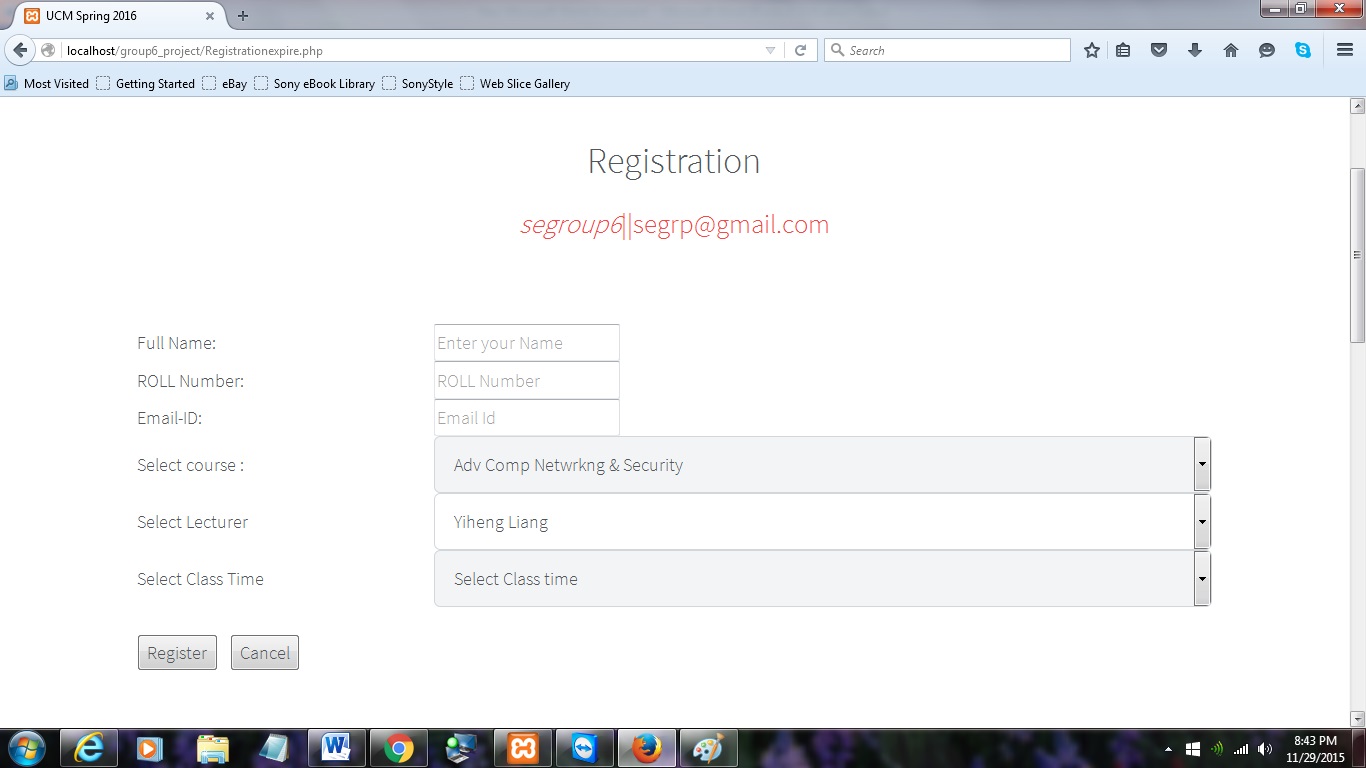
**Fig (7.1) Home page**



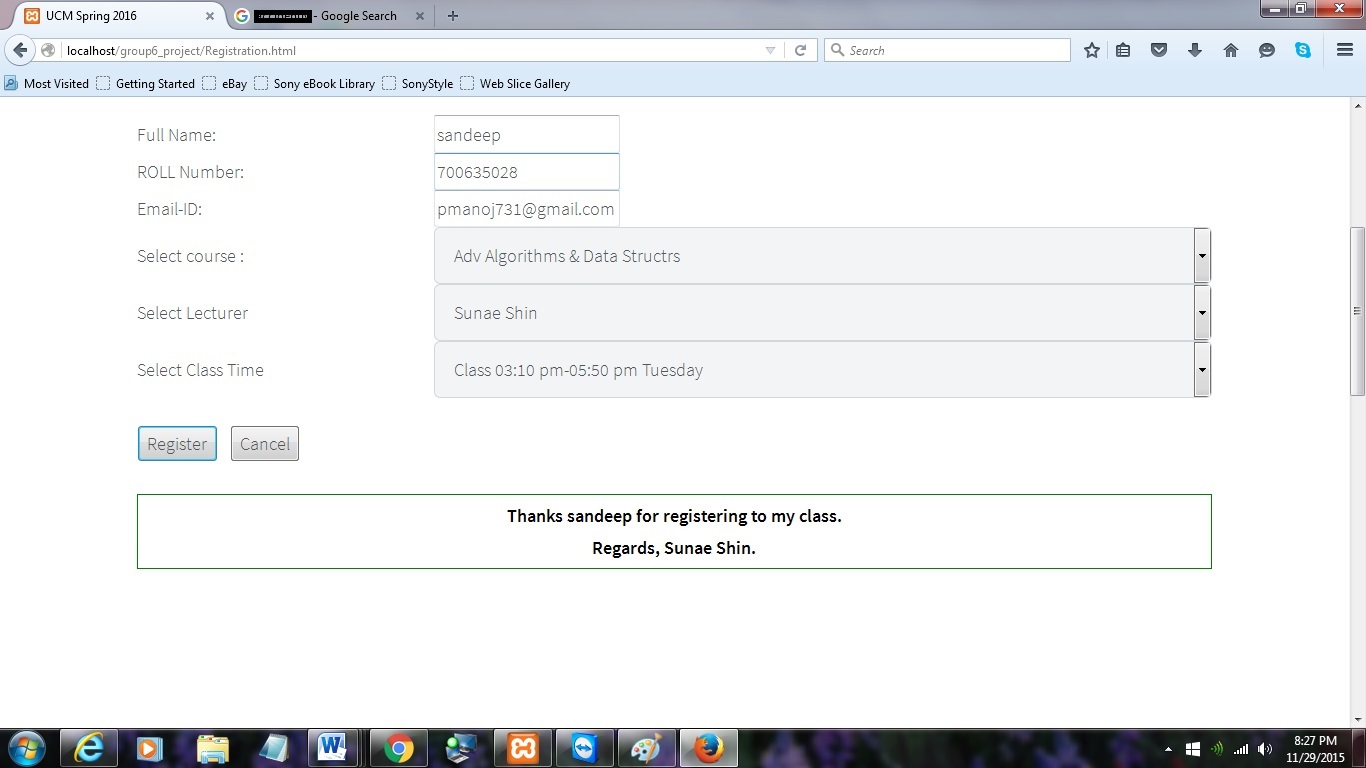
**Fig (7.2) Navigation Bar**



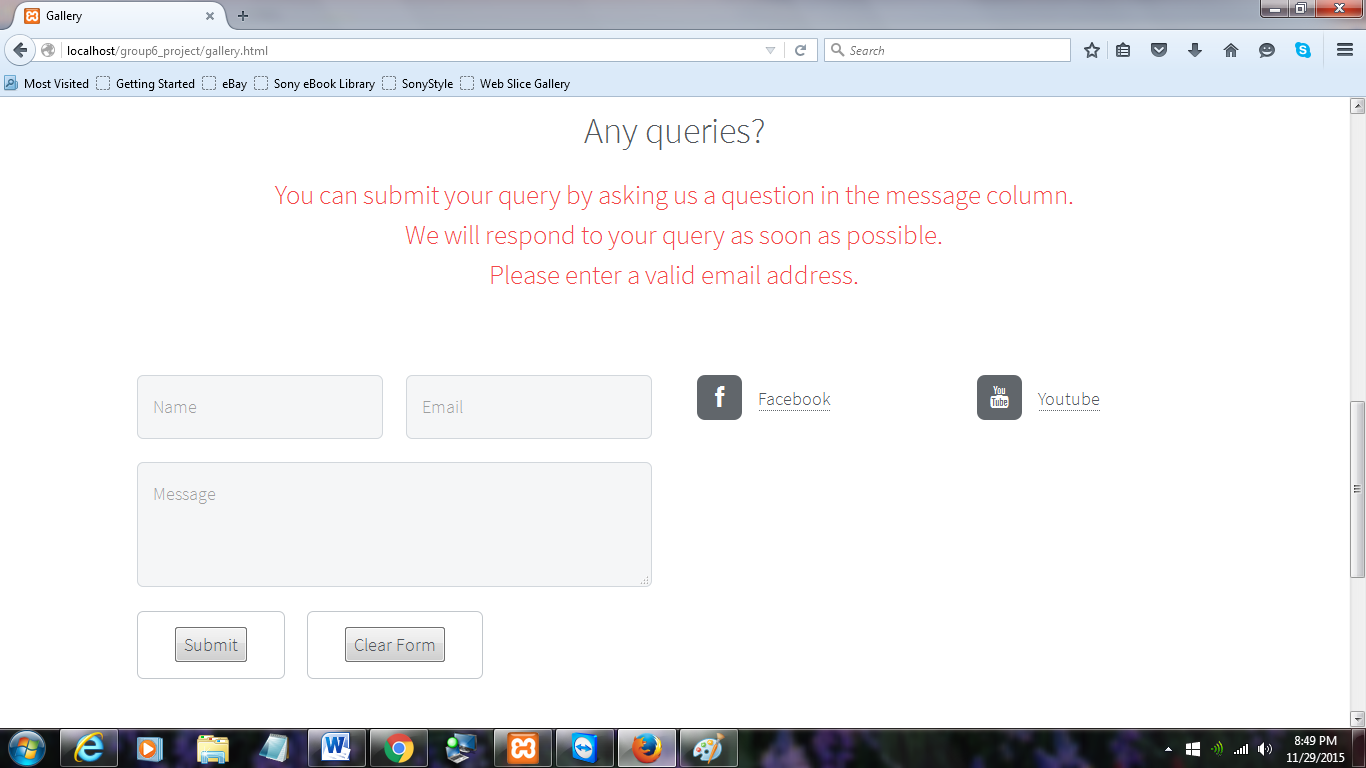
**Fig (7.3) Gallery Page**



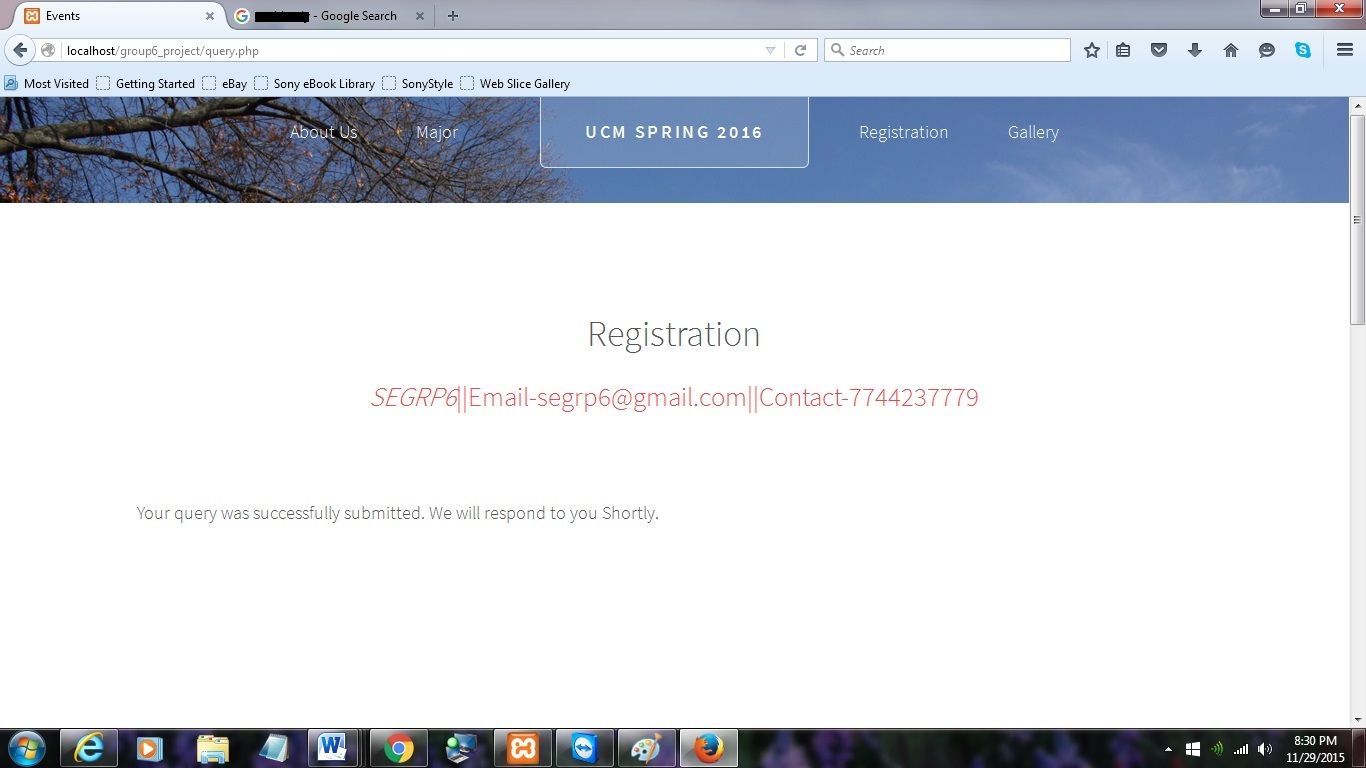
**Fig (7.4) Registration Page**



**Fig (7.5) After Sucessful Registration.**



**Fig (7.6) The Query option available to the users is shown above.**



**Fig (7.7) After Query has sent successfully.**

**9 CONTRIBUTIONS**

**Sandeep Reddy Aredla [700635578]:**

- Worked on Design phase

- Documented Requirements

-Had discussions on each workflow in the group

**Manoj Kumar Potlapalli [700635028]**

**-**Worked on Implementation

**-**Documented Requirements

- Had discussions on each workflow in the group

**Ravali Gujja [700641004]**

-Documented Requirements

-Documented Introduction

- Had discussions on each workflow in the group

**Himabindu Chittimalla[700641264]**

-Worked on Analysis, Testing

-Documented Requirements

- Had discussions on each workflow in the group

**10 References**

1. [www.html5up.net](http://www.html5up.net)
2. [www.youtube.com/phpacademy](http://www.youtube.com/phpacademy)
3. [www.apachefriends.com](http://www.apachefriends.com)
4. [www.w3cschools.com/css](http://www.w3cschools.com/css)

**10.1 Tools and Techonologies Used**

The tools used for creating this web application are

1. Notepad++
2. Apache HTTP server
3. SMTP Server
4. MySQL
5. PhpMyAdmin

The Technologies used are

1. HTML5
2. CSS
3. JavaScript
4. Jquery
5. Php