# University of Central Missouri



**Advanced System Project - CRN 31484**

**Hemanthreddy Mulagundla**

**700641425**

**CIS 5690 – Advanced Systems Project**

**Dr. Sam Ramanujan**

**Dept. of Computer Information’s Systems**

**TABLE OF CONTENTS**

1. **Project Description**

1.1. Functionalities in the website

1. **List of actors and their goals**
   1. Functionalities in the website

2.1.1. Student

2.1.2. Faculty

2.1.3. Administrator

* 1. Goals

1. **Use Cases**
   1. Use Case Descriptions
   2. Use case Diagram
2. **Class Diagram**
3. **Sequence diagrams** 
   1. Sequence diagram for User Page
   2. Sequence diagram for Faculty Page
4. **Entity Relationship Diagram**
5. **Screen Shots**
6. **Technologies**
7. **Project Description.**

**Institute Management System (Institute Website)**

* Created a web application for institute purpose. And this site is useful for both students and faculty.
* This website is designed to help the students in registering the courses, update their information and students can view their registered courses. Students can select and unselect the courses.
* Faculty would also be able to manage their account and create and update the course details. And a faculty can update their personal details.
* Admin would be able to delete and edit both students and faculty. And he can be able to delete and edit course information.

**1.1 Functionalities in the website are as below:**

* Website consists of various pages like Login, User Registration, User Page, Admin Page and Faculty Page altogether would help the students and faculty in updating the details. It would help the students to enroll the courses.
* In this website, Students privileges include

1. Can able to create an account.
2. Update his/her Profile.
3. Can enroll the courses.
4. Can able to delete from the enroll courses.
5. Have access to check all the course details, faculty, timings of course, number of seats in that course and fee structure for that course.

* Admin privileges include

1. Can manage Both faculty and students accounts.
2. Can delete and update students and faculty.
3. Can update and delete courses.

* Faculty Account Privileges include

1. Can create an account.
2. Can able to update their personal information.
3. Can able to Create a course.
4. Can able to update the course.

# **List of Actors and their Goals or Primary Purpose**

**Actors:**

1. Student
2. Faculty
3. Administrator

**Goals:**

1. **Student:**
   1. Enroll a course
   2. Check available course details
   3. Unselect the course
   4. Update his/her information
2. **Faculty:**
   1. Create a course
   2. Update a course
   3. Update his/her personal information
3. **Administrator:**
   1. Manage Students
   2. Manage Faculty
   3. Edit and delete students and faculty information.
   4. Edit and delete courses.

# **Use Cases**

**3.1 Use Case for more than 2 Subject and will get 1% discount on total fee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | 3.1 | | | |
| **Use Case Name:** | If a student creates more than 2 subjects, student will get 1%. | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Student | | |
| **Description:** | | While enrolling for a course, if a student enrolls more than 2 courses student will get 1% discount on his total fee. | | |
| **Trigger:** | | It will show all the registered courses in the selected courses tab. | | |
| **Preconditions:** | | If a student wants to get 1% discount he should enroll more than 2 subjects then only he can eligible to get 1% discount. | | |
| **Postconditions:** | | Once he done with courses the sum fee will be appear in a label control and he will go for payment option. | | |
| **Normal Flow:** | | a. Student select the available course.  b. If he selects more than 2 the sum will appear in a label control.  c. Then the student will go for payment option. | | |
| **Exceptions:** | | If he doesn’t enroll more than 2 courses the fee will be available in a label control | | |
| **Includes:** | | It will include the already existing course fee in a label control. | | |
| **Frequency of Use:** | | 10 per day. | | |
| **Special Requirements:** | | a. Student must enroll more than 2 courses.  b. Student must be an active member. | | |
| **Notes and Issues:** | | a. If the student doesn’t register for the more than 2 courses then this use case will not play any role.  b. This will work only when student enroll more than 2 courses. | | |

3.2 Use Case for different payment methods.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | 3.2 | | | |
| **Use Case Name:** | Use case for different payment methods and extra charge for each payment method | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Students | | |
| **Description:** | | This use case will give information about different payment methods for every payment type student will charge some extra amount | | |
| **Trigger:** | | It will show all the available payment methods student can go for any of the payment option listed in the dropdown list. | | |
| **Preconditions:** | | The student must enroll the course and enrolled sum will be available in a label control then only student can go for payment option. | | |
| **Postconditions:** | | Once he selects the payment method. the additional extra fee will be added to the student existing fee. | | |
| **Normal Flow:** | | a. Student will register for the course  b. Course will be available in the selected course.  c. Course sum will be displayed in a label control  d. Then student will go for payment method  e. depends on payment method he will charge some extra amount | | |
| **Exceptions:** | | If he doesn’t go for payment option | | |
| **Frequency of Use:** | | 10 per day | | |
| **Special Requirements:** | | a. Student must be an active member  b. Student must enroll the course | | |
| **Notes and Issues:** | | a. If the student does not register or the course he can’t go for payment option.  b. The student must be an active member and enrolled for the course then only he can able to do payment option. | | |

**3.3 Use Case for faculty can create limited number of courses.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | 3.3 | | | |
| **Use Case Name:** | A Faculty can create limited number of courses | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Faculty | | |
| **Description:** | | A faculty can create limited number of courses in a semester. He can't create more than 4 courses per term. | | |
| **Trigger:** | | While creating a course he need to provide all the required information like course name, description, Fee, No of seats per course, duration of the course, timings of the course. | | |
| **Preconditions:** | | a. Faculty must be active  b. Faculty must have less than 4 courses | | |
| **Postconditions:** | | Created course must be available to the all active students | | |
| **Normal Flow:** | | a. faculty will create the course  b. Created course will be available to the students.  c. Student can enroll for the course.  d. Enrolled course will be added to the selected courses | | |
| **Exceptions:** | | If the faculty try to create more than 4 courses the error message " The maximum courses limit reached. You should not be able to add any more courses; Thanks! An error message will be displayed in a label control. | | |
| **Includes:** | | Once the course is created all the available courses displayed in a grid view | | |
| **Frequency of Use:** | | 10 per day | | |
| **Special Requirements:** | | a. Faculty must be an active member.  b. He must not have less than 4 courses | | |
| **Notes and Issues:** | | If a faculty trying to create five (5) course. He can’t create 5 course and he will get an error message. | | |

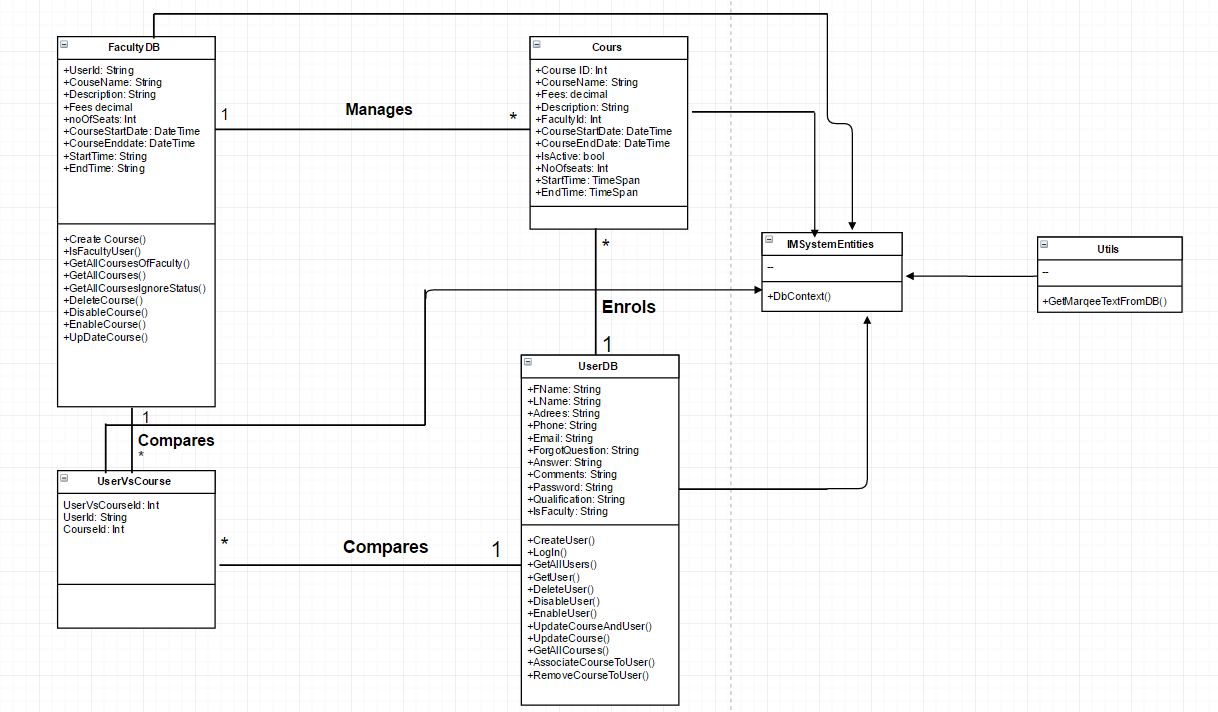
**3.4 Use Case for Sum of the fee.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | 3.4 | | | |
| **Use Case Name:** | Sum of fee displayed in a label control | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Enrolled student | | |
| **Description:** | | It will show the sum of all courses fee in a label control. | | |
| **Trigger:** | | It will show all the courses available in a grid view called selected courses | | |
| **Preconditions:** | | * 1. Student must enroll the course   2. The student must be active member | | |
| **Postconditions:** | | Student will be enrolled for the course. | | |
| **Normal Flow:** | | 1. Student will select the course 2. Selected course will be available in textboxes. 3. Student enrolled for the course. 4. Selected courses fee will be shown in a label control | | |
| **Exceptions:** | | The total sum is “0 “if there is no selected course. | | |
| **Frequency of Use:** | | 10 per day | | |
| **Special Requirements:** | | 1. The student must be an enrolled student. 2. The course must be an active and available course. | | |
| **Notes and Issues:** | | If a student select more than two courses will get some an extra discount. | | |

**3.5 USE CASE DIAGRAM**

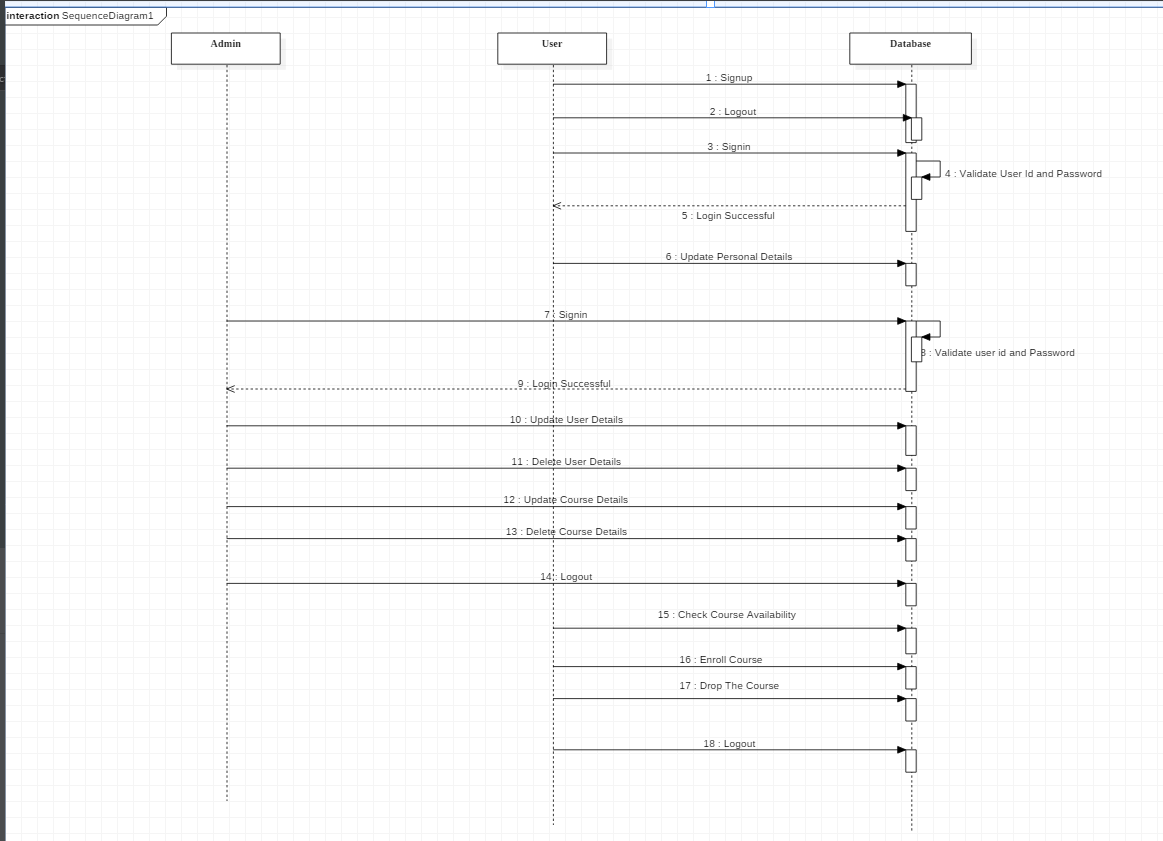


1. **Class Diagram**

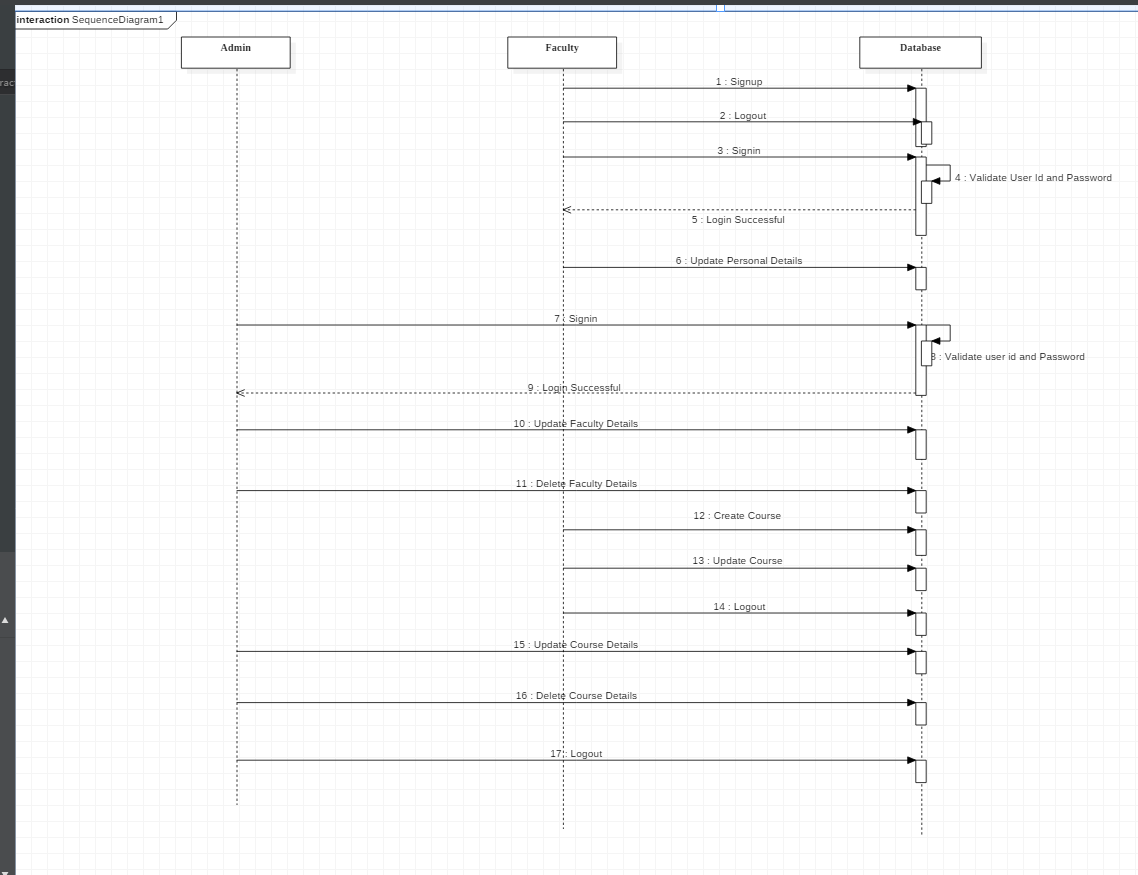


1. **SEQUENCE DIAGRAMS**

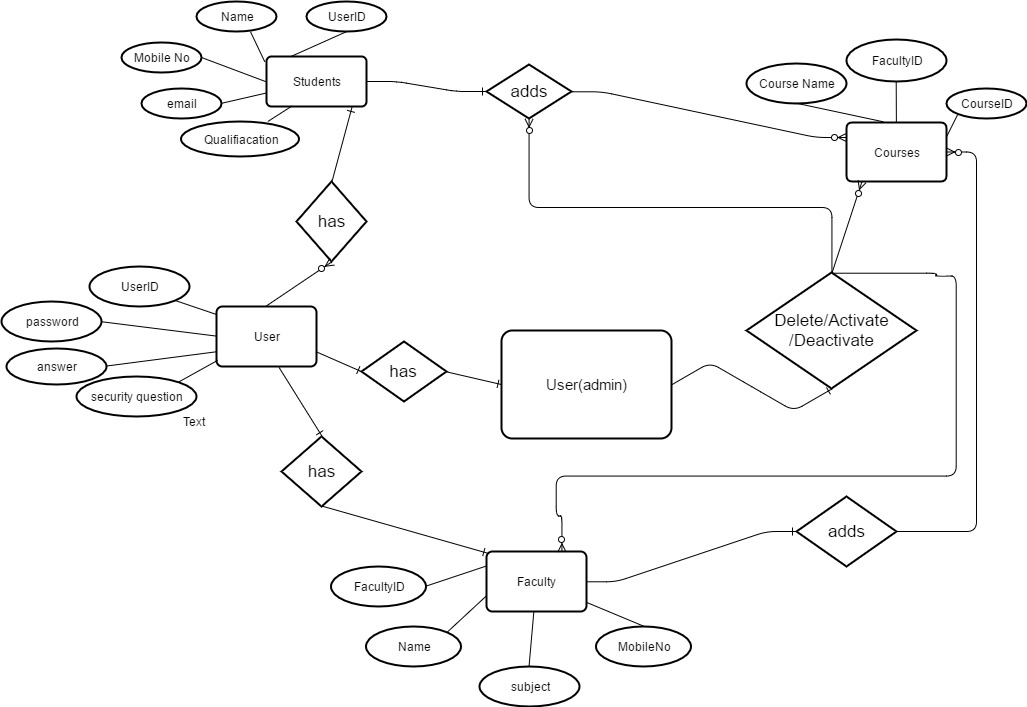
**5.1 Sequence Diagram for User Page:**

****

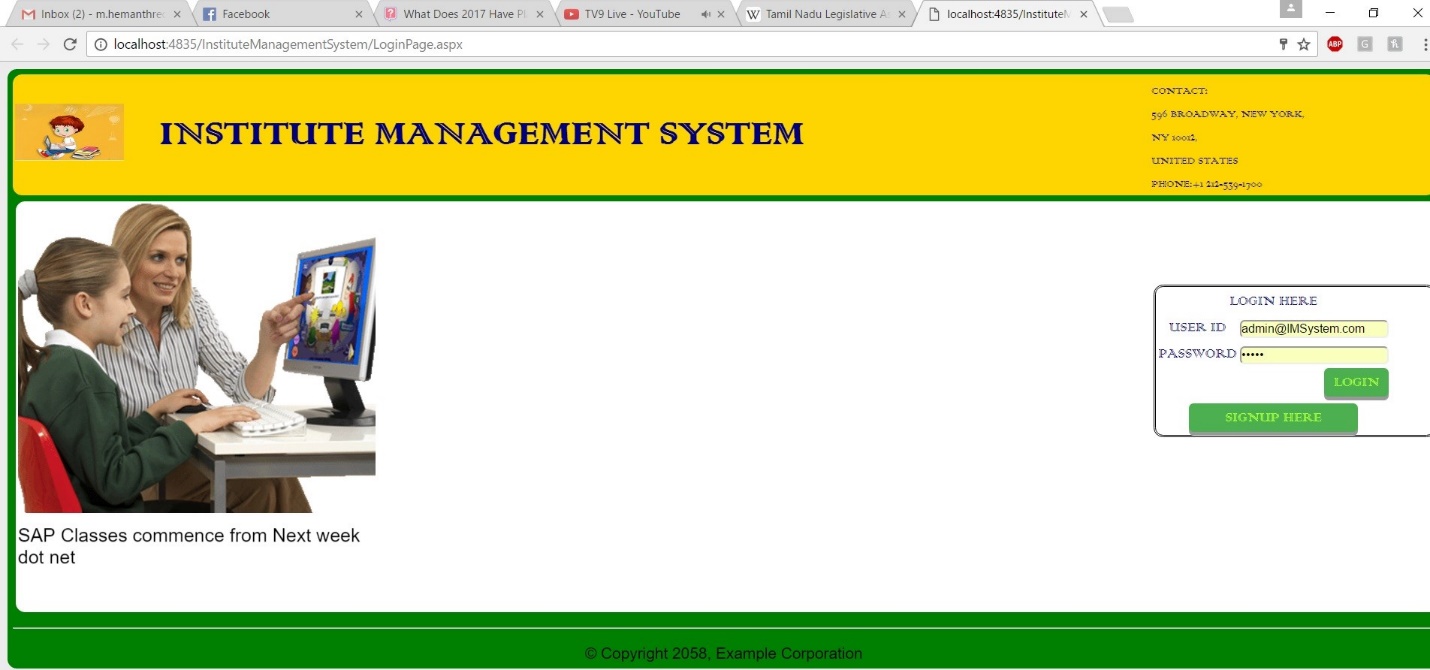
**5.2 Sequence Diagram for Faculty Page:**

****

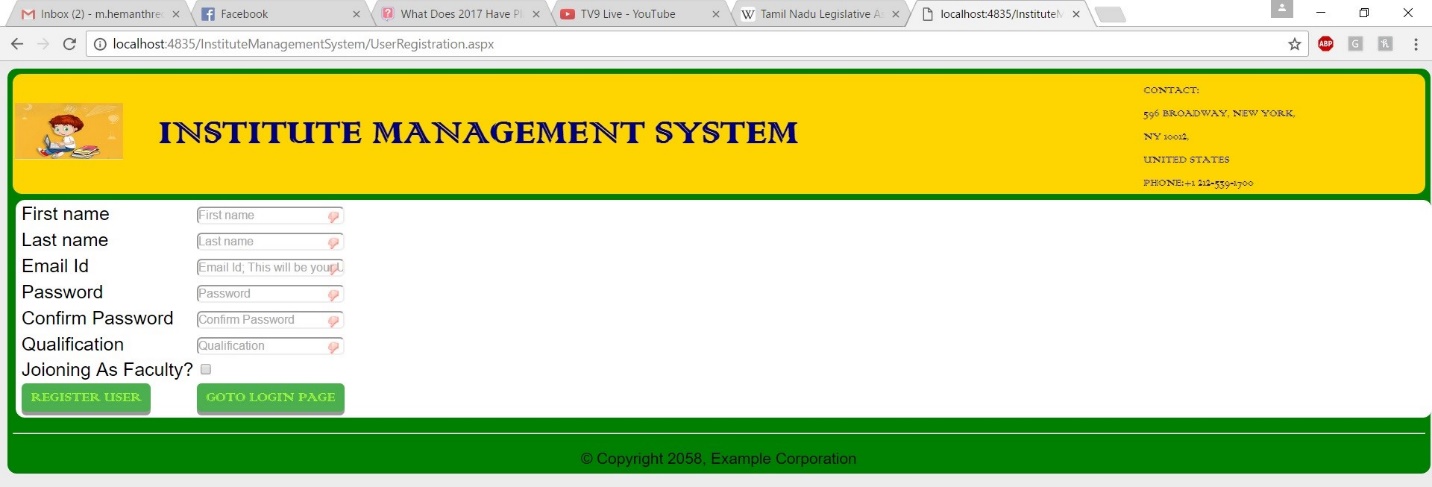
1. **ER Diagram**

****

1. **Screen shots:**
   1. **LOGIN PAGE**

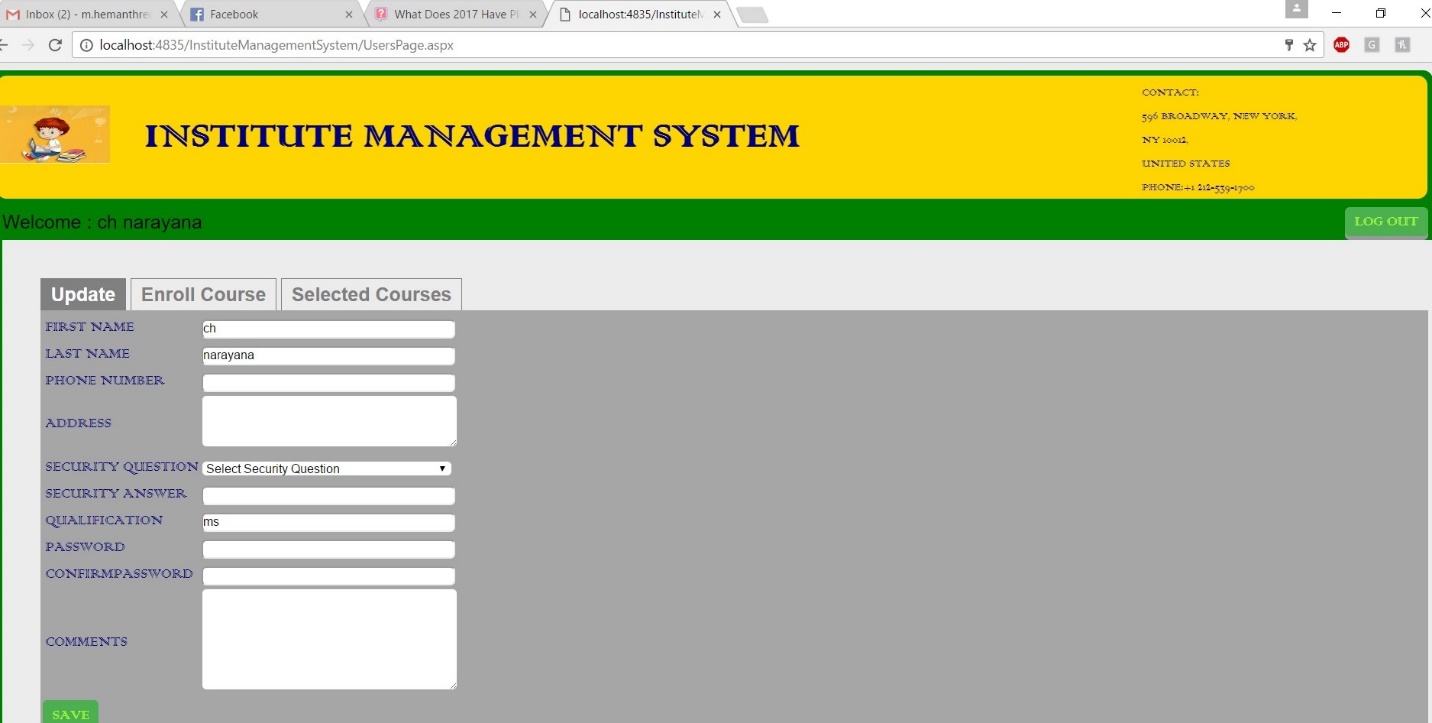
****

**7.2.1 USER REGISTRATION**

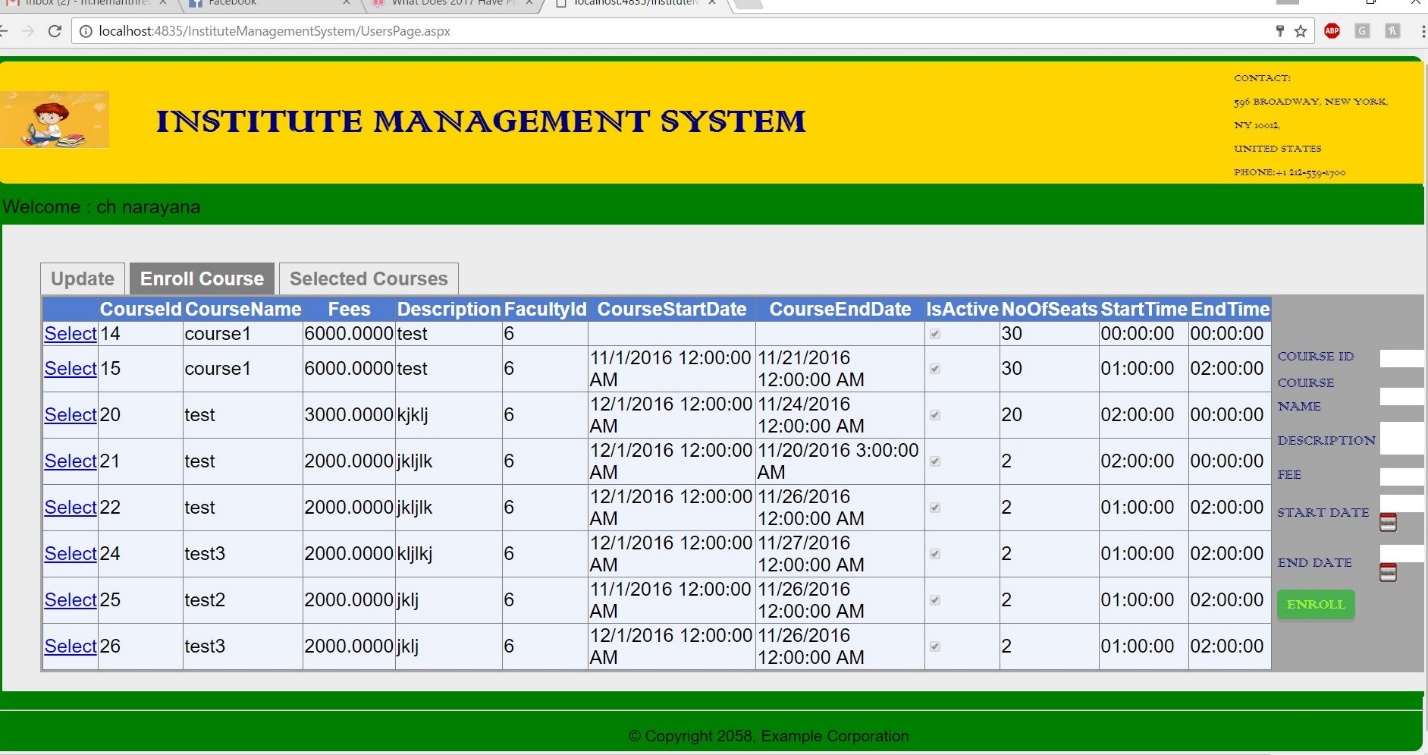
****

* 1. **USERSPAGE**

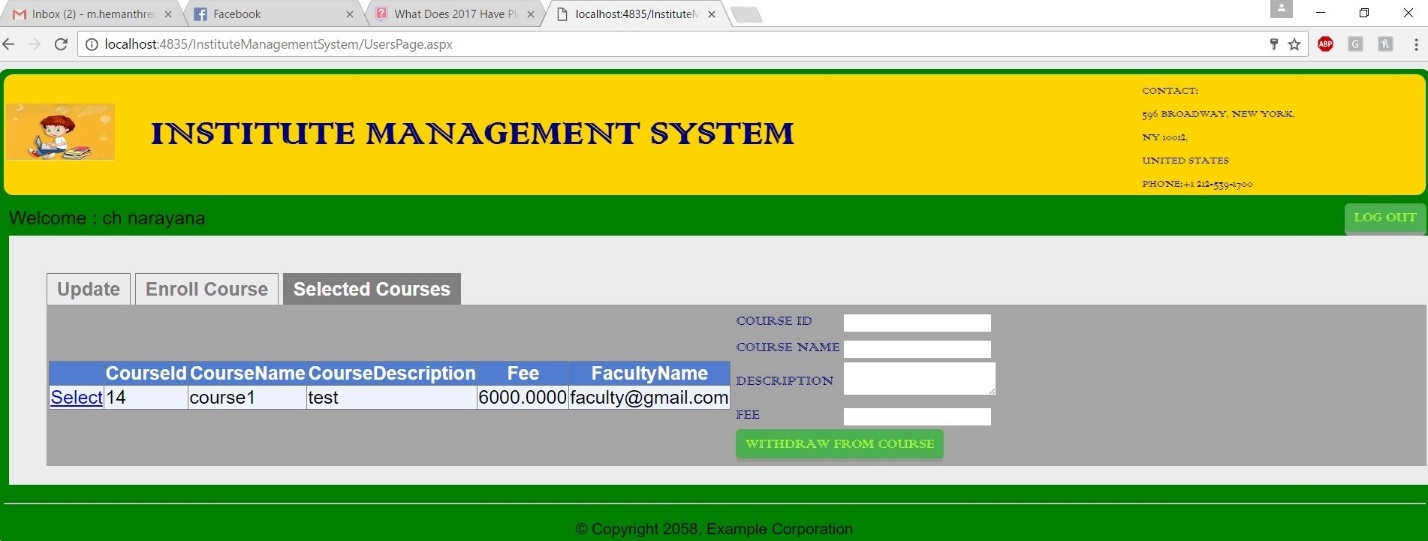
**7.3.1 STUDENT UPDATE**

****

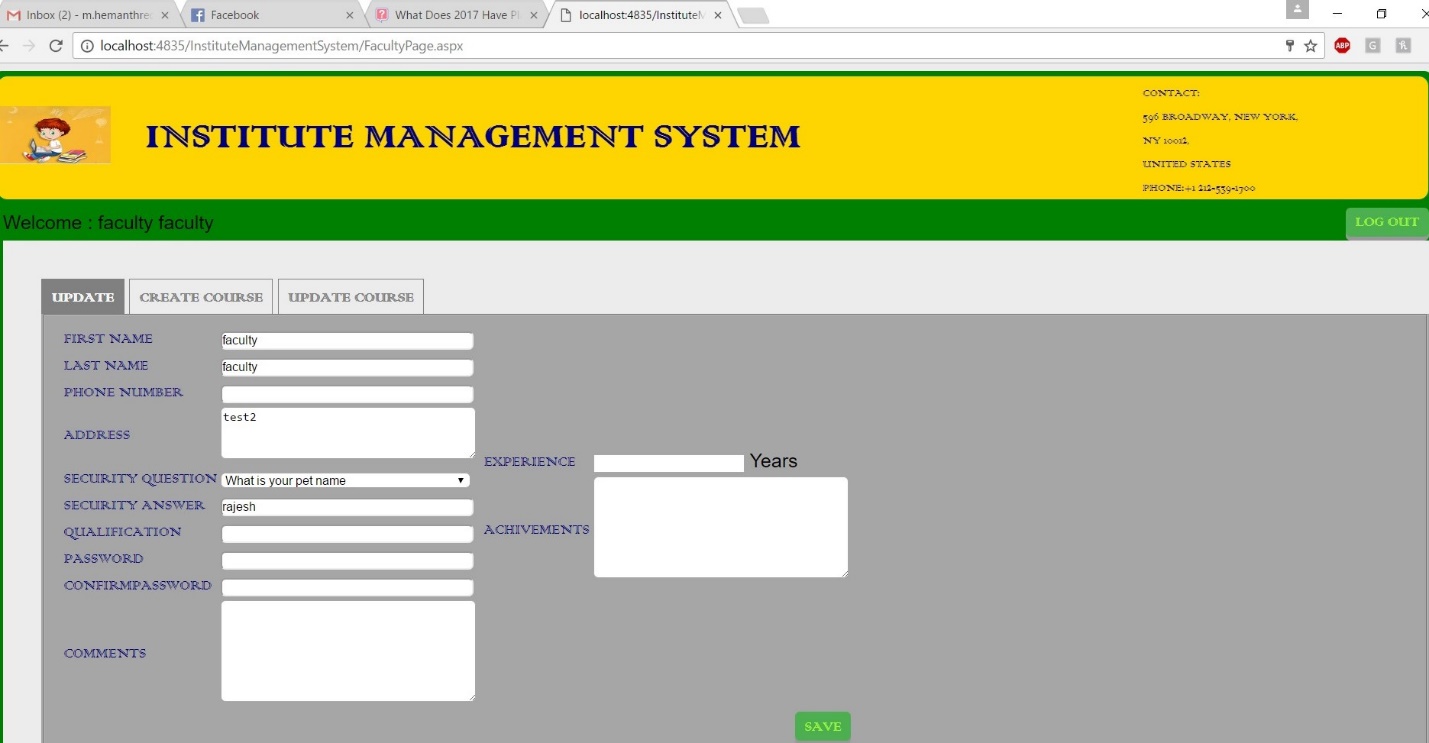
* + 1. **STUDENT ENROLLCOURSE**

****

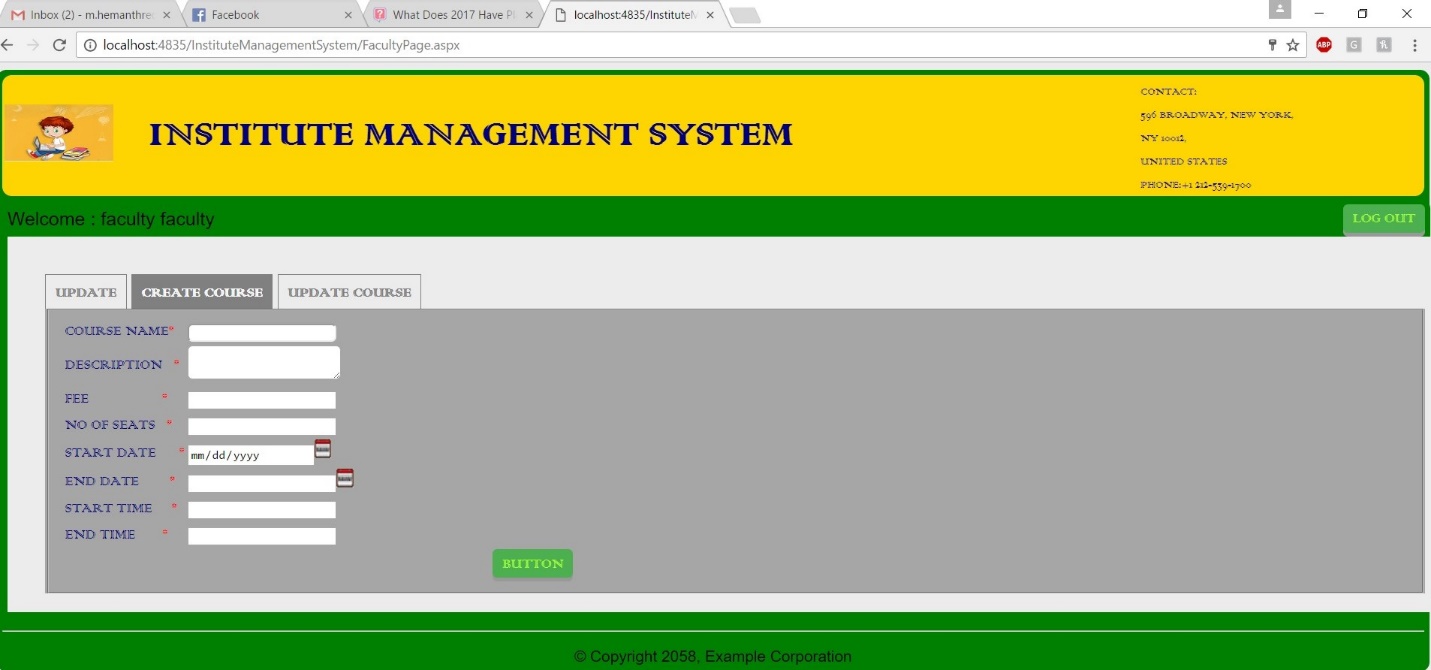
* + 1. **STUDENT SELECTED WITHDRAW**

****

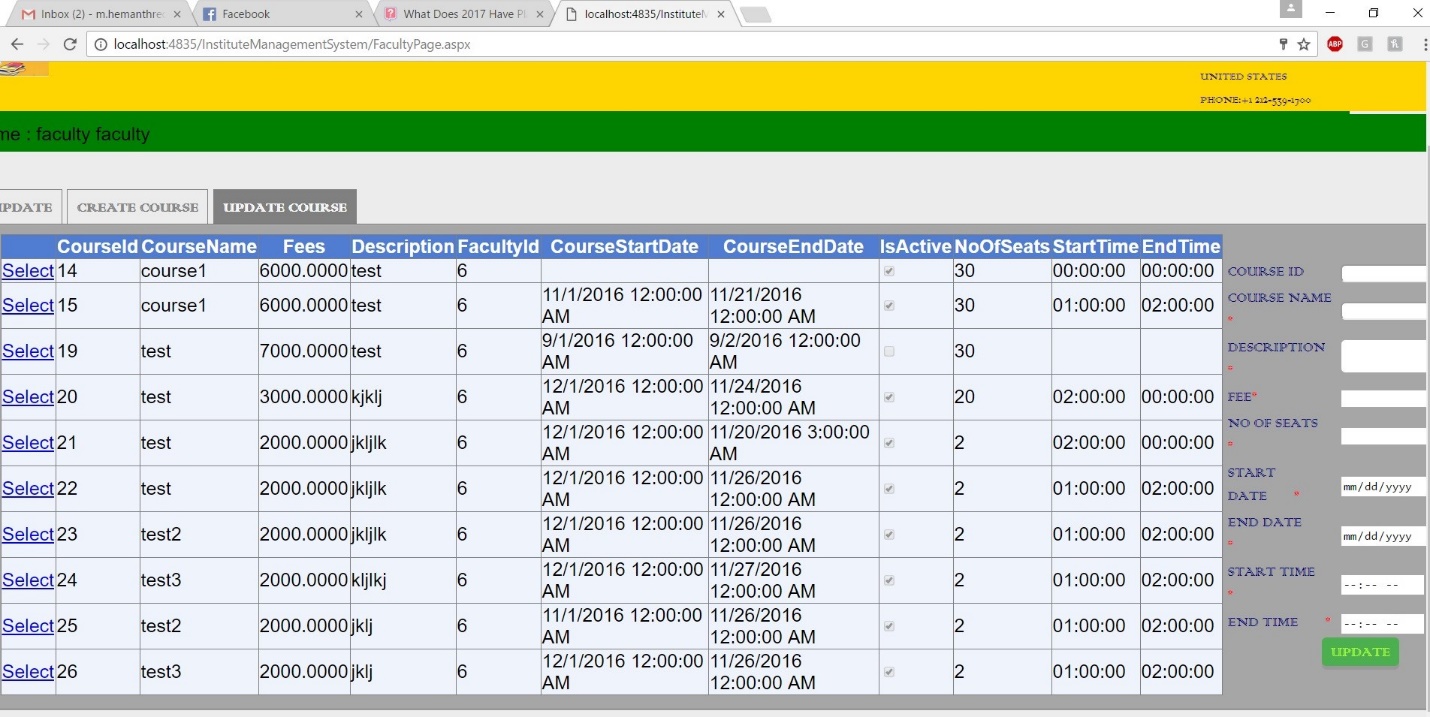
* 1. **FACULTYPAGE**

**7.4.1 FACULTY UPDATE** 

* + 1. **CREATE COURSE**

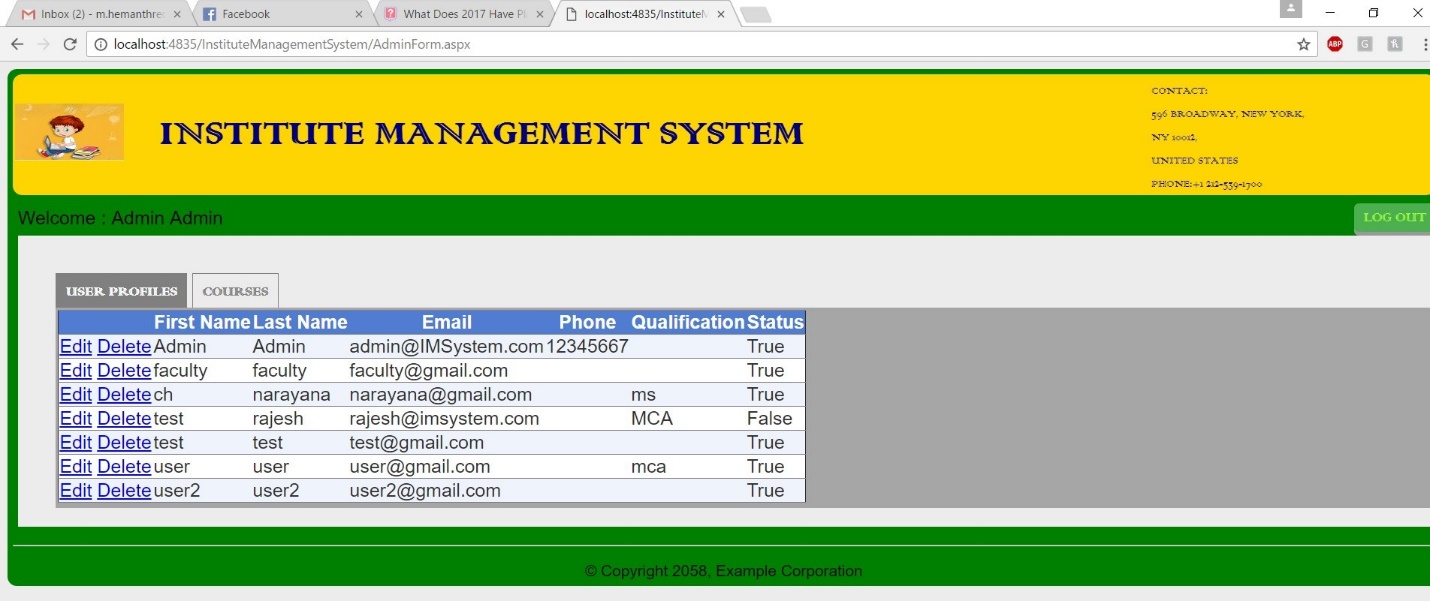
****

* + 1. **UPDATE COURSE**

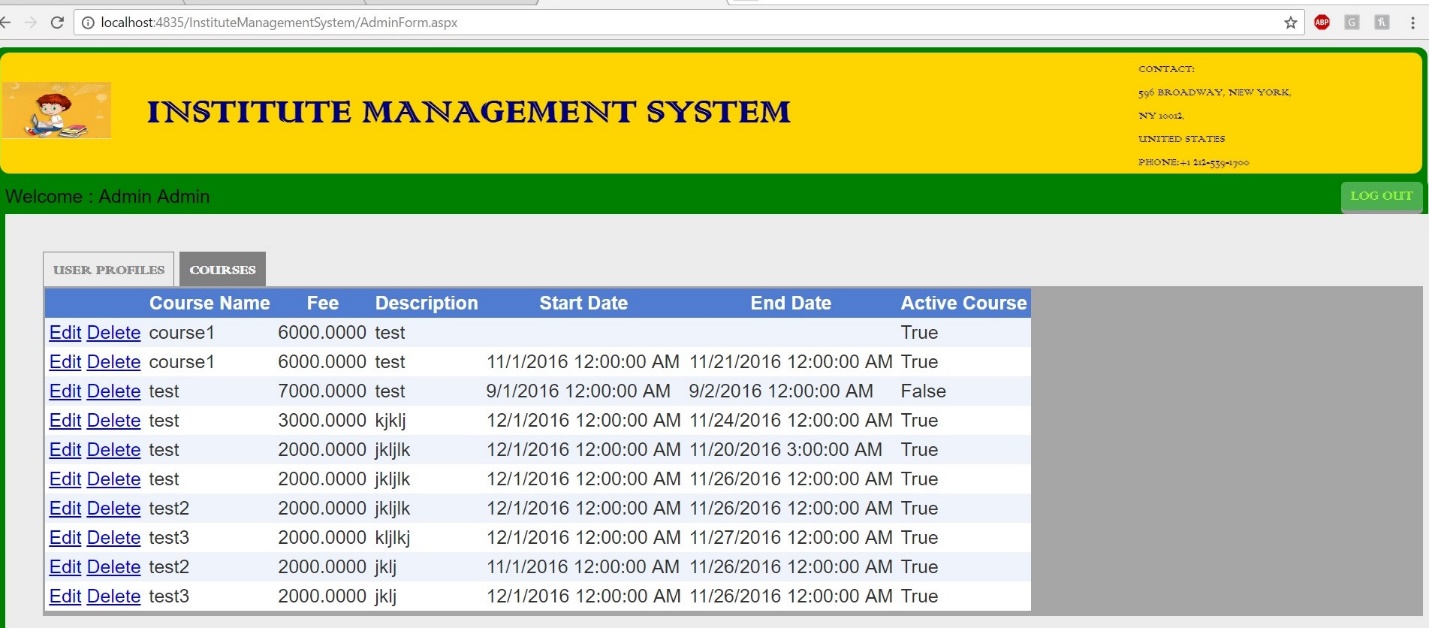
****

* 1. **ADMINPAGE**

**7.5.1 USER PROFILE**

****

* + 1. **ADMIN COURSES**

****

1. **Technologies used to design application:**
   1. ASP .NET
   2. CSS
   3. C#
   4. jQuery
   5. Entity framework
   6. SQL Server (SSMS)