Integrated University Department Information System

Submitted By
Vaishnavi Bhambure (1156082)

December 15th, 2017

Presented to
Department of Computer Science
New York Institute of Technology

Software Engineering
CSCI_665-2017 FALL
Dr. Ennoure

Table of Contents

ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to our advisor Prof. Dr. Ennoure for the continuous support of our study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of designing and implementing this project. We could not have imagined having a better advisor and mentor for our Course.

We are also thankful to all the faculty and staff members of the Department of Computer Science of New York Institute of Technology for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

Vaishnavi Bhambure

ABSTRACT

The Integrated University Department Information System is gaining more importance as the number of its users are increasing rapidly. As the number is rising there is a need of effective management of library, one such effective system is our Integrated University Information System it's designed using PHP, HTML and CSS as front end and communicating with PHP MyAdmin SQL as back-end using XAMPP as a local server.

The transactions like log in, register, add, search, is provided. The Integrated University Department Information System stores the details like name, address, student ID number, of users who come to University. The details of students like student name, student number, Fees, year of Admission, the total number of students, Faculty that are present in the university etc. are also stored.

Modules

Basically, there are two main modules, they are

- Students
- Teacher
- Administrator

Student Module

The Student module can login, add class, can pay the fees. Student can view grades, manage transcripts. Student can communicate via mail.

Teacher Module

The Teacher have the functionalists like adding class, giving grades to student, can check the student profile, send mail or notification, can update the class information.

Admin Module

The Admin module is able to create/modify/delete/view class. Also able to create/modify/delete/view any user account. Admin from university system must be able to handle all expenses. Administrators have access to entire IUDIS and its database.

1. Introduction:

There are three access levels in this Integrated University Department Information System, which are ‘Administrator’, ‘Teacher’ and ‘Student’ . The admin can be is managing all the students and faculty in the university .

Integrated University Department Information System offers many flexible and convenient features, allowing admin, student and teacher to maximize time and efficiency. IUDIS gives the all detailed information about student and teacher. It will track how many class are available in university and whether that particular class is filled or not . It will provide the total student count in a University. It keeps the record of the students and teachers. software is customizing for any IUDIS.

Features of Integrated University Department Information System:

- Only basic knowledge of computers is required for operation of Integrated University Department Information System. As it has user-friendly application interface.
- Integrated University Department Information System is customizable and User Configurable.
- An inbuilt Settings module make IUDIS flexibility to cater to diverse organizational needs.
- IUDIS brings information to the user’s desktop through integration across all modules.
  - Student record is maintained
  - Teacher record is maintained
  - Automatic fees calculation
  - Keeps record of number of students enrolled
  - Configurable as per user’s requirements

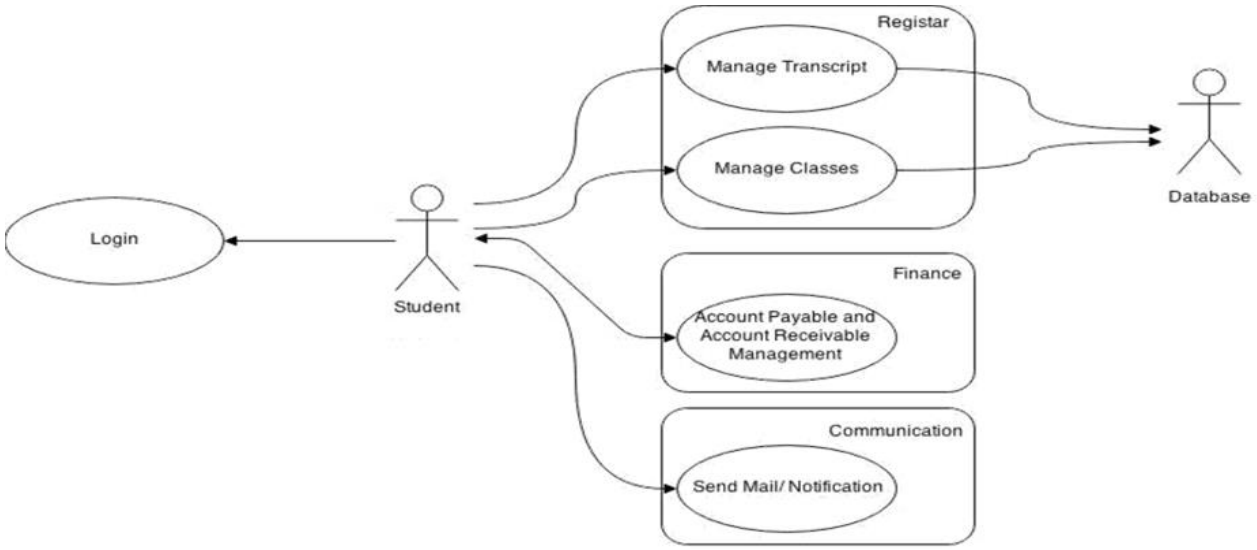
Why you need it:

- Improved customer service through greater access to accurate information.
- Increased productivity and it eliminates duplication of effort.
- More economical and safer means of storing and keeping track of information.
- Easier access to Information like management reports and stock etc, as well as more accurate and faster results from statistical analyses.
- Reduces errors and eliminating the long and repetitive manual processing.
- Greater accountability and transparency in operations.
- Improved efficiency and effectiveness in administration and management as it has unprecedented access to real-time information.
- More reliable security for sensitive and confidential information.

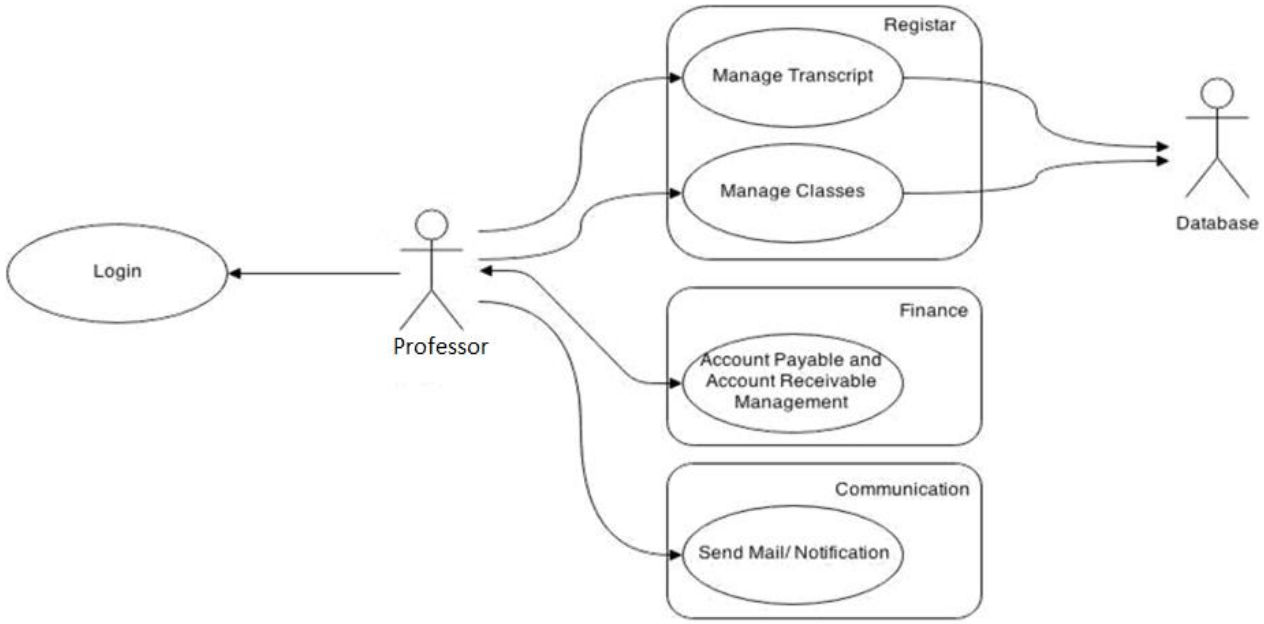
Appropriate knowledge-based action and intervention

2. Use Case Daigram:

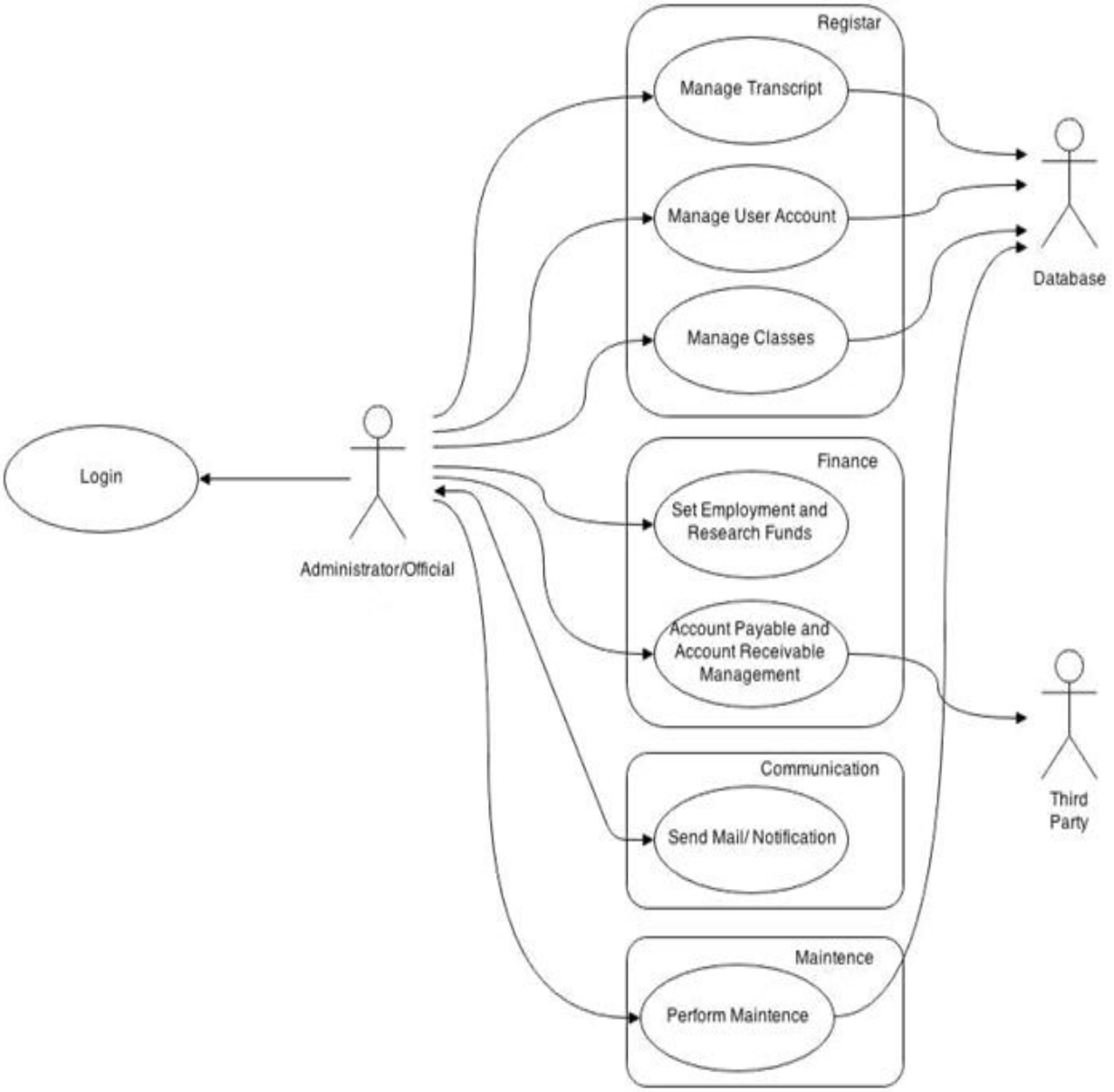
- **Student**

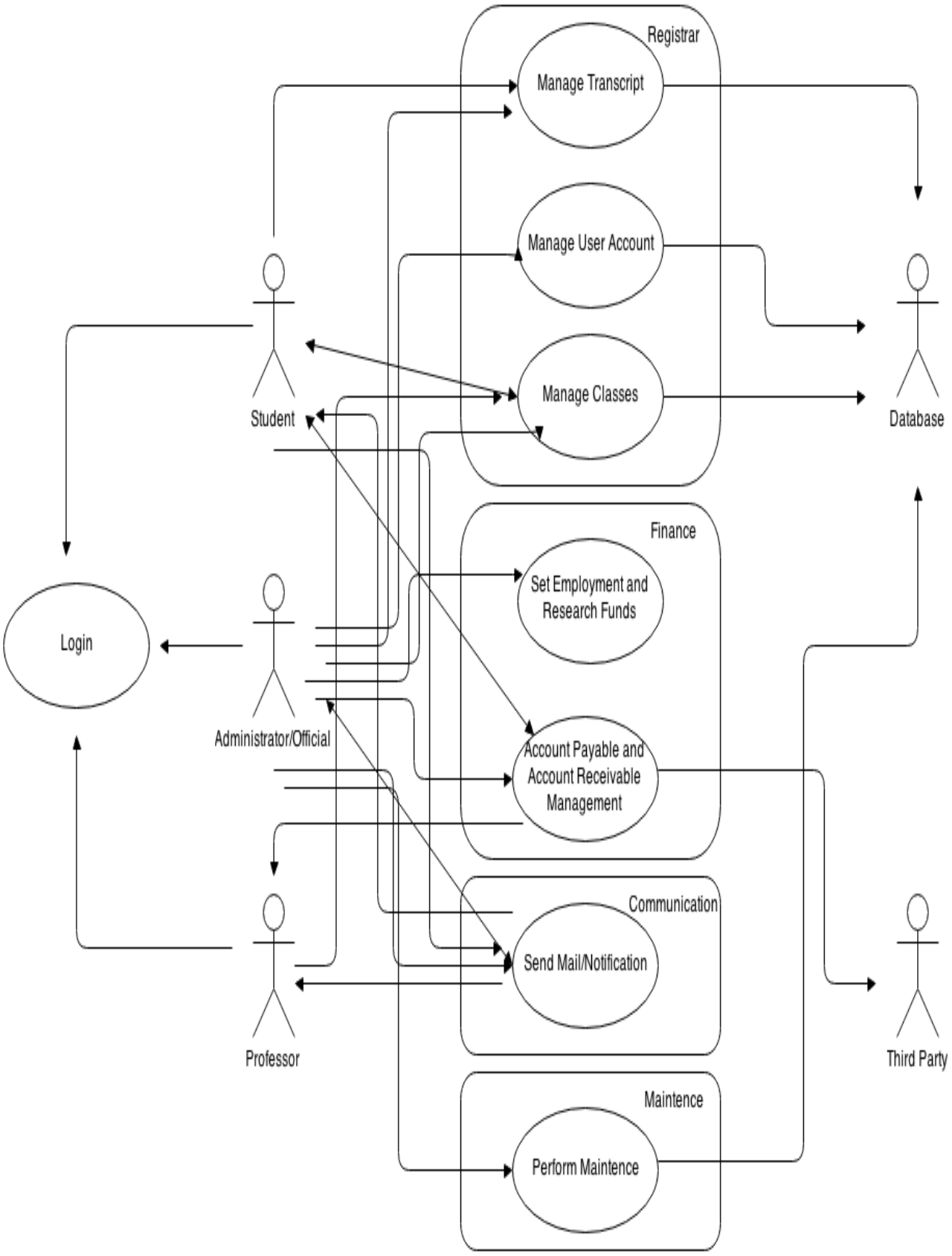


- **Professor**

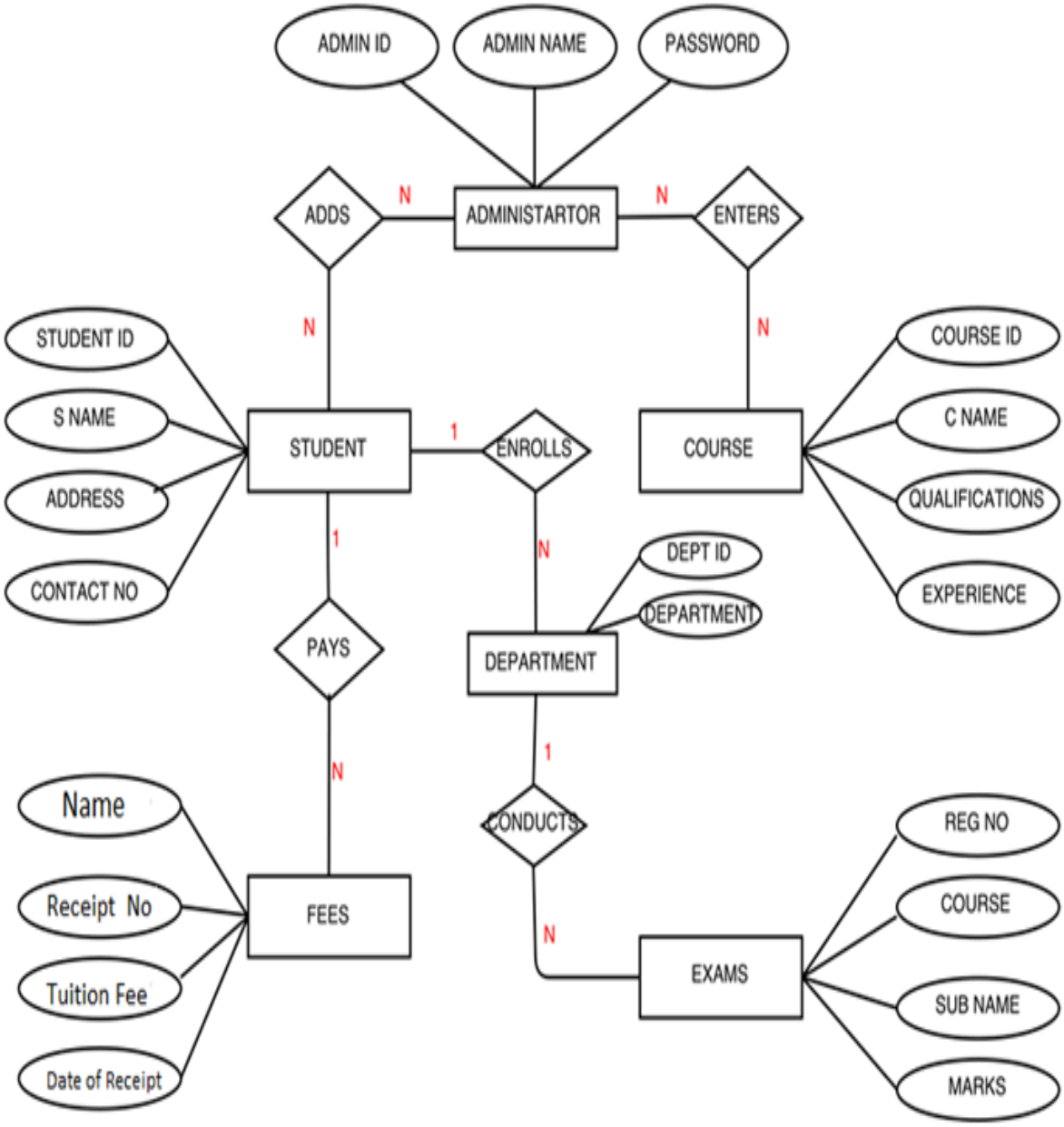


- **Admin**


Registar
Manage Transcript
Manage User Account
Manage Classes
Database
Login
Administrator/Official
Finance
Set Employment and Research Funds
Account Payable and Account Receivable Management
Third Party
Communication
Send Mail/ Notification
Maintence
Perform Maintence


Registrar
Manage Transcript
Manage User Account
Manage Classes
Student
Database
Finance
Set Employment and Research Funds
Login
Administrator/Official
Account Payable and Account Receivable Management
Communication
Send Mail/Notification
Professor
Third Party
Maintence
Perform Maintence

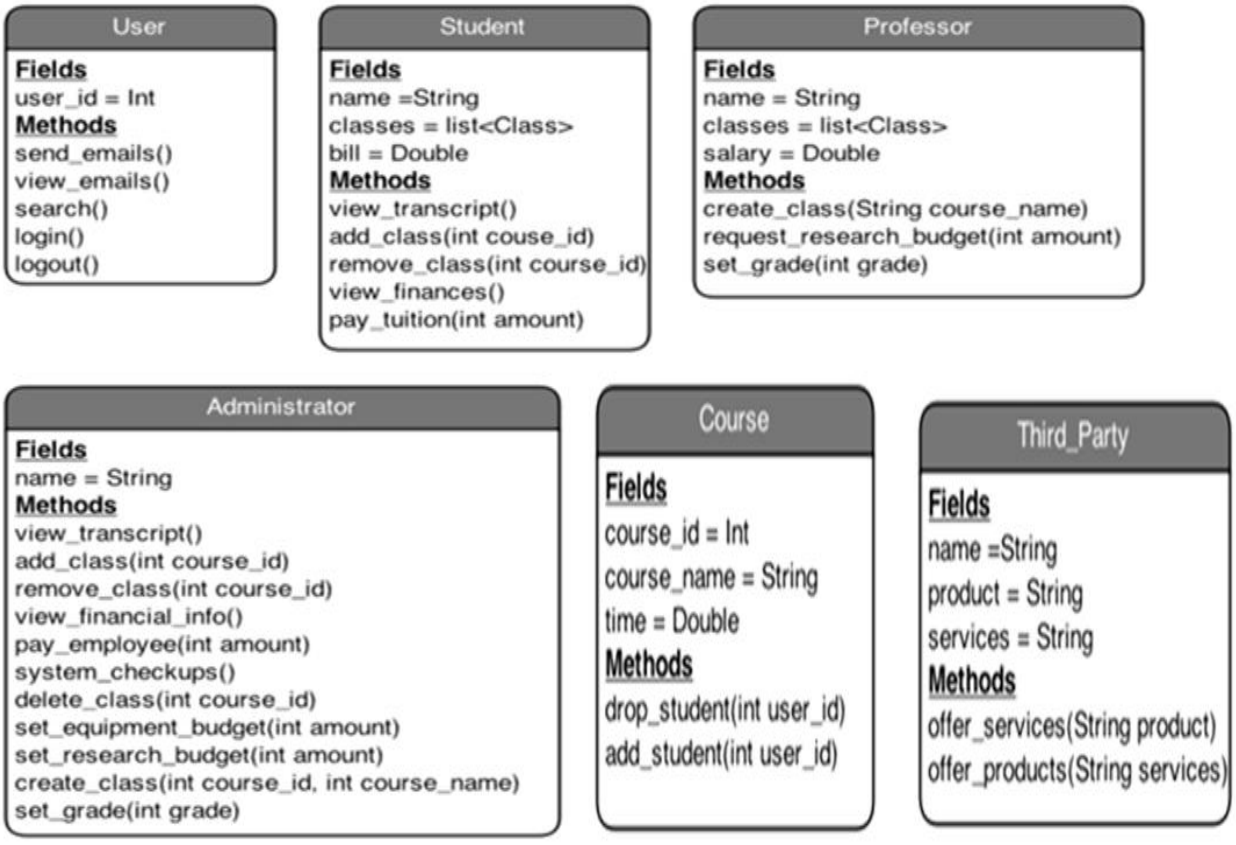
3. ER data model design:



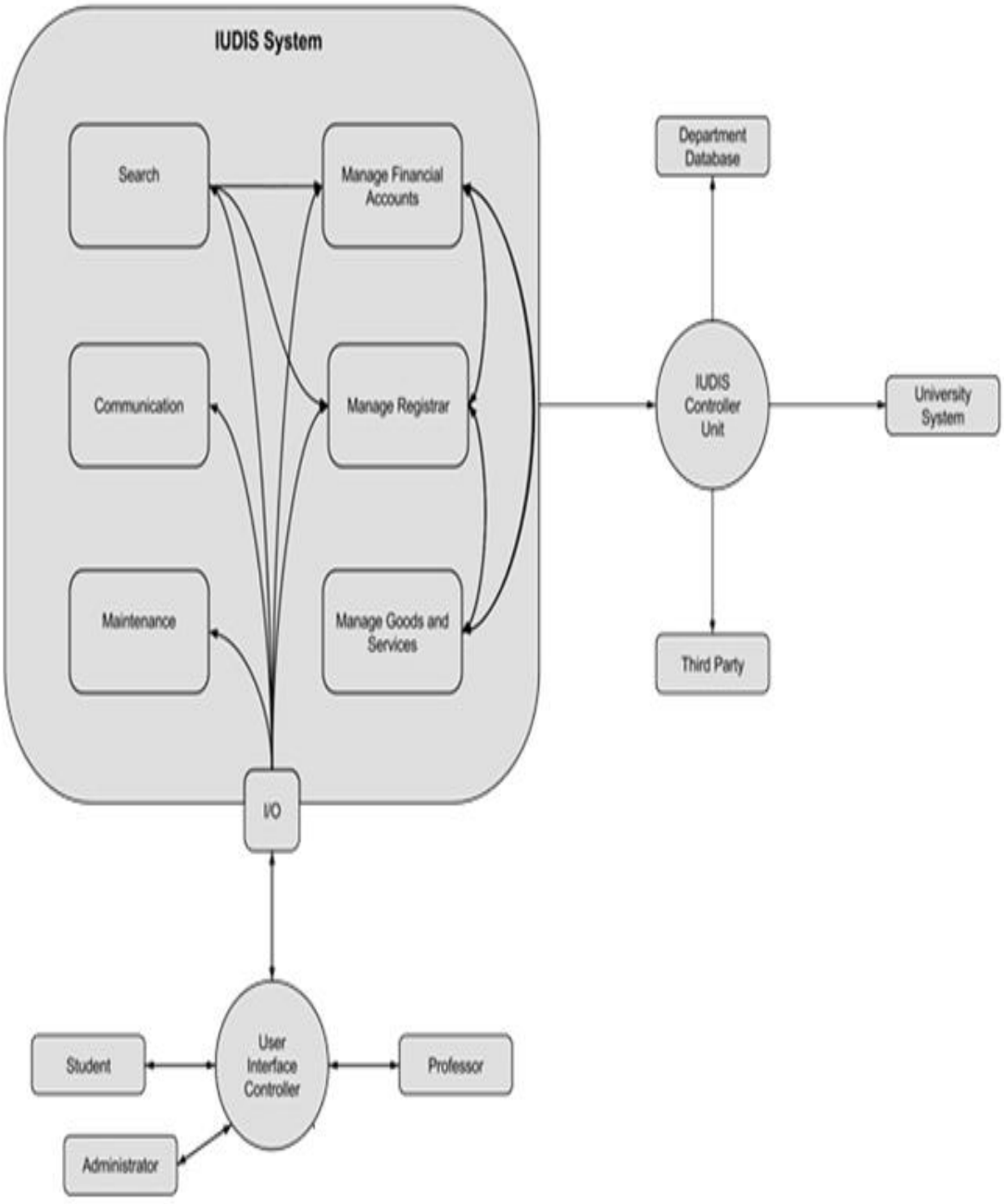
- List of Entities and their attributes:

ENTITIES	**ATTRIBUTES**
Student	Student_ID, Name, Phone, Address, Password
Teacher	Teacher_ID, Name, Phone, Address, Password
Admin	Admin_ID, Name, Password
Department	Department_ID, Department_Name
Course	Course_ID, Course_Name, Qualification, Experience
Fees	Name, Receip_No, Tuition_Fee, Date_of_Receipt
Admin	Admin_ID, User name, Password

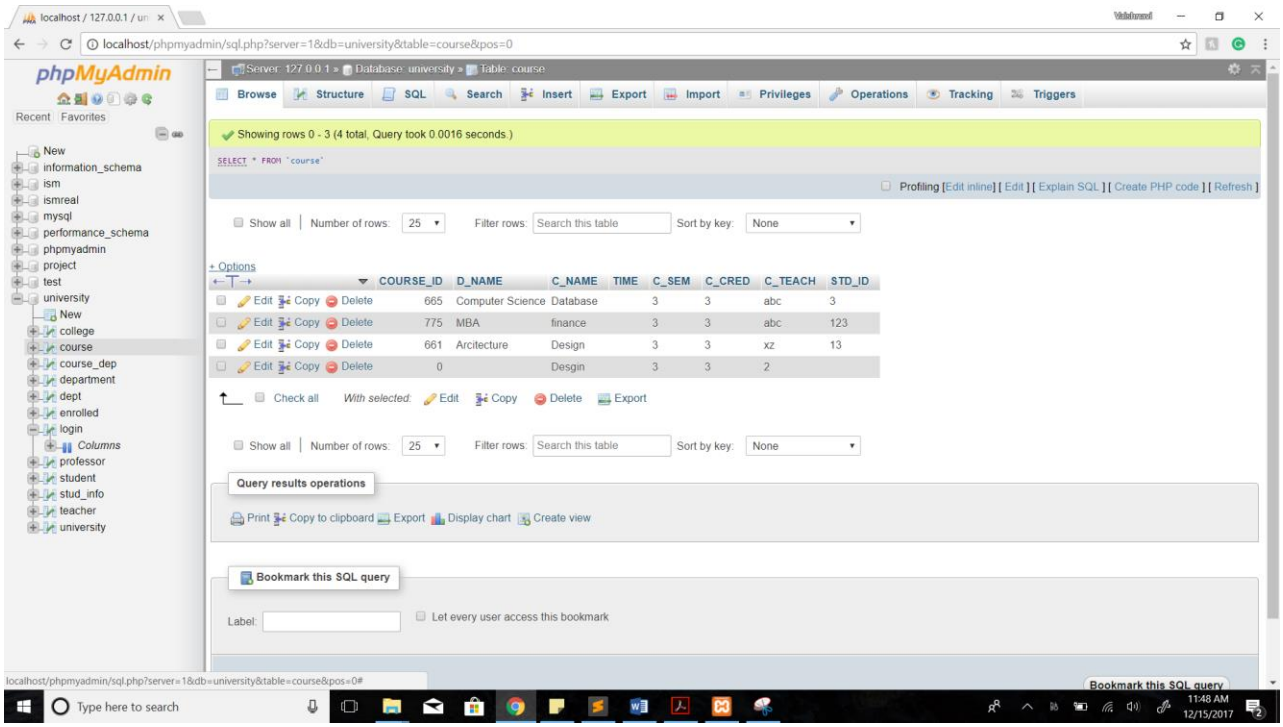
4. Class Diagram:

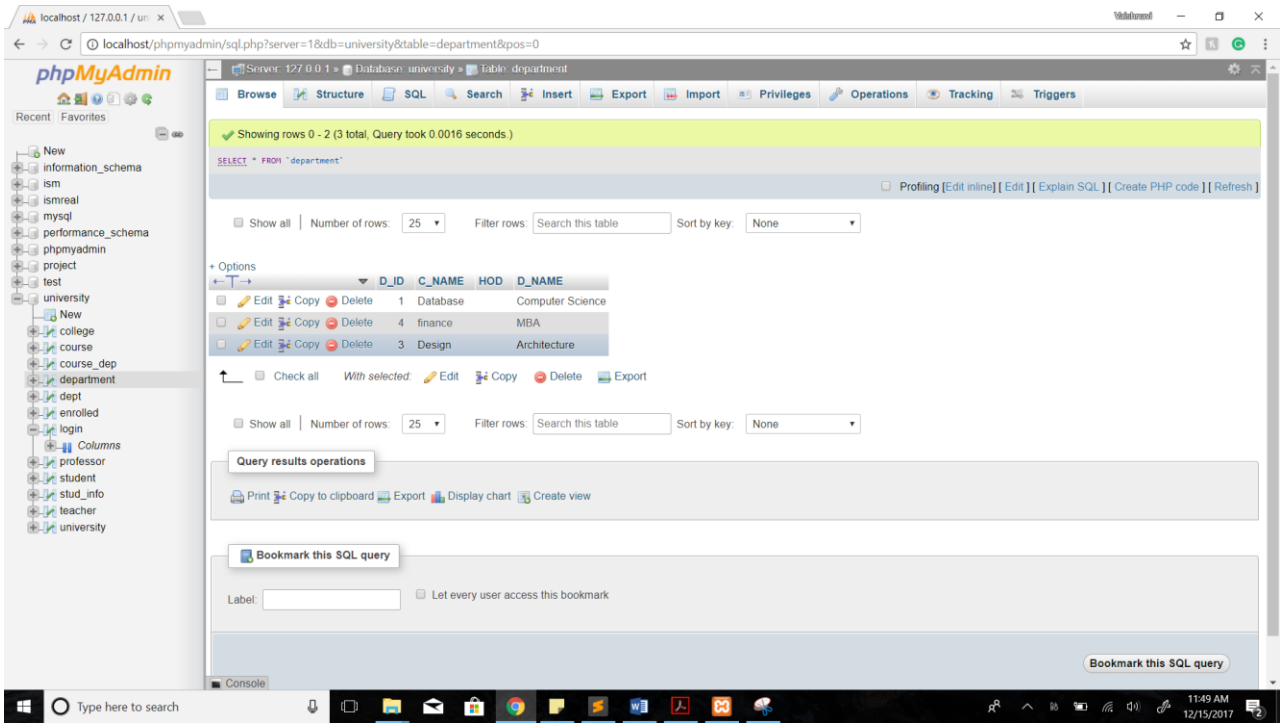


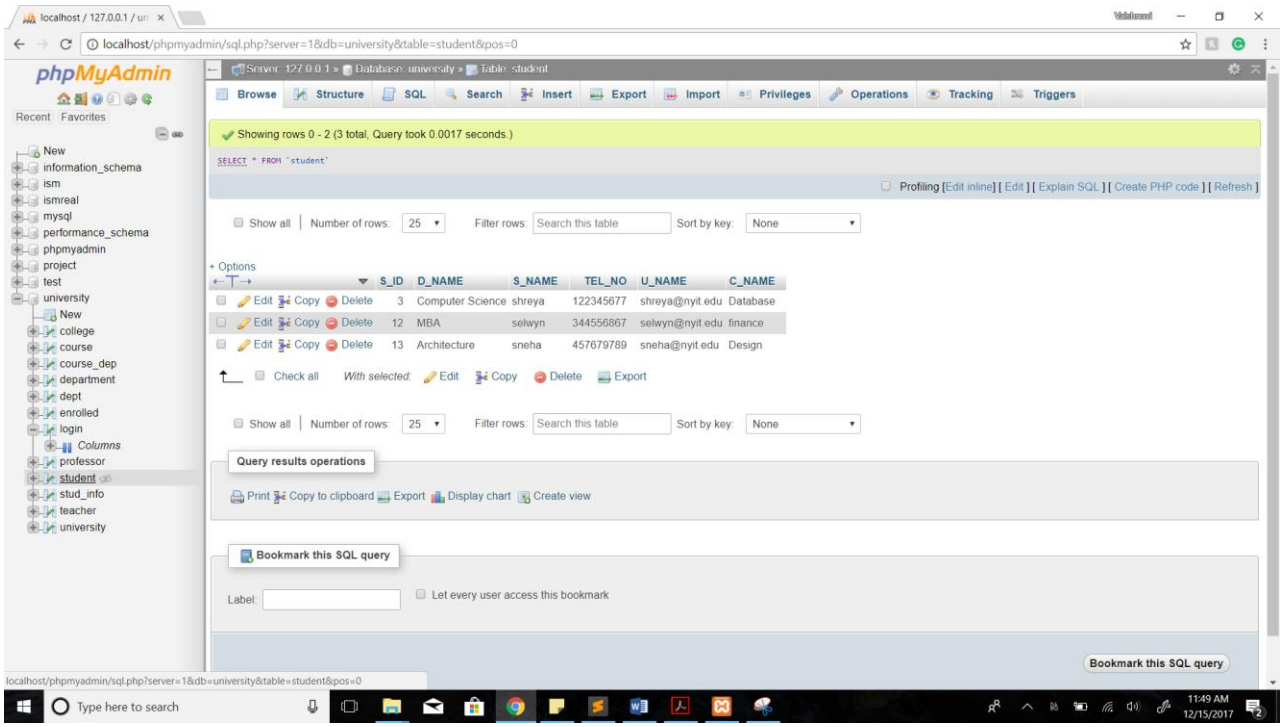
5. Architecture Diagram:



6. Logical Design of the Database







Entity	**Primary/Candidate Key**	**Foreign key**
Student	Student_ID	
Teacher	Teacher_ID	
Admin	Admin_ID	
Department	Department_ID	
Course	Course_ID	
Fees	Receipt_No	
Admin	Admin_ID	
Course_Dept	{Course_ID, Dept_ID}	{Course_ID, Dept_ID}
Enroll_Course	{Student_ID, Course_ID}	{Student_ID, Course_ID}

7. Relational Database Design

- First Normal Form (1NF)

The relations are already in first normal form as all the attributes are atomic. Thus, there is no need of flattening or decomposing the tables.

- Second Normal Form (2NF)

The relations are in second normal form if are all non primary attributes are fully dependent on the primary key.

ENTITIES	ATTRIBUTES
Student	Student_ID, Name, Phone, Address, Password
Teacher	Teacher_ID, Name, Phone, Address, Password
Admin	Admin_ID, Name, Password
Department	Department_ID, Department_Name
Course	Course_ID, Course_Name, Qualification, Experience
Fees	Name, Receip_No, Tuition_Fee, Date_of_Receipt
Admin	Admin_ID, User name, Password
Course_Dept	Course_ID, Dept_ID
Enroll_Course	{Student_ID, Course_ID}

- BCNF

All the relations are already in BCNF.

- Final Normalized Relations:

ENTITIES	ATTRIBUTES
Student	Student_ID, Name, Phone, Address, Password
Teacher	Teacher_ID, Name, Phone, Address, Password
Admin	Admin_ID, Name, Password
Department	Department_ID, Department_Name
Course	Course_ID, Course_Name, Qualification, Experience
Fees	Name, Receip_No, Tuition_Fee, Date_of_Receipt
Admin	Admin_ID, User name, Password
Course_Dept	Course_ID, Dept_ID
Enroll_Course	{Student_ID, Course_ID}

8. Implementation of Database and SQL Query

- A description of the creation of the database schema and instance.

Database:

- Primary Keys and Foreign Keys are assigned to each table.

Database Queries

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `university management`
--
CREATE DATABASE `university` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `university`;

-- --------------------------------------------------------
Table structure for table `admin`

CREATE TABLE `admin` (
  `adminId` int(11) NOT NULL AUTO_INCREMENT,
  `userName` varchar(45) NOT NULL,
  `password` varchar(45) NOT NULL,
  PRIMARY KEY (`adminId`),
  UNIQUE KEY `userName_UNIQUE` (`userName`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;


--
-- Table structure for table `college`
--

CREATE TABLE IF NOT EXISTS `college` (
  `C_ID` int(5) NOT NULL,
  `U_NAME` varchar(20) NOT NULL,
  `C_NAME` varchar(20) NOT NULL,
  `DEAN` varchar(20) NOT NULL,
  PRIMARY KEY (`C_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


-- --------------------------------------------------------

--
-- Table structure for table `course`
--
```

```
CREATE TABLE IF NOT EXISTS `course` (
  `COURSE_ID` int(5) NOT NULL,
  `D_NAME` varchar(20) NOT NULL,
  `COURSE_NAME` varchar(20) NOT NULL,
  `TIME` varchar(10) NOT NULL,
  PRIMARY KEY (`COURSE_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


-- --------------------------------------------------------

--
-- Table structure for table `department`
--


CREATE TABLE IF NOT EXISTS `department` (
  `D_ID` int(5) NOT NULL,
  `C_NAME` varchar(20) NOT NULL,
  `HOD` varchar(50) NOT NULL,
  `D_NAME` varchar(20) NOT NULL,
  PRIMARY KEY (`D_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Dumping data for table `department`
--

-- --------------------------------------------------------

--
-- Table structure for table `login`
--

CREATE TABLE IF NOT EXISTS `login` (
  `username` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Dumping data for table `login`
--

INSERT INTO `login` (`username`, `password`) VALUES
('admin', 'admin');
```

```
-- --------------------------------------------------------

--
-- Table structure for table `professor`
--

CREATE TABLE IF NOT EXISTS `professor` (
  `P_ID` int(5) NOT NULL,
  `D_NAME` varchar(20) NOT NULL,
  `P_NAME` varchar(20) NOT NULL,
  `AGE` int(3) NOT NULL,
  PRIMARY KEY (`P_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Dumping data for table `professor`
--


-- --------------------------------------------------------

--
-- Table structure for table `student`
--

CREATE TABLE IF NOT EXISTS `student` (
  `S_ID` int(5) NOT NULL,
  `D_NAME` varchar(20) NOT NULL,
  `S_NAME` varchar(20) NOT NULL,
  `TEL_NO` int(13) NOT NULL,
  `U_NAME` varchar(20) NOT NULL,
  `C_NAME` varchar(20) NOT NULL,
  PRIMARY KEY (`S_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Dumping data for table `student`
--


-- --------------------------------------------------------


CREATE TABLE IF NOT EXISTS stud_info (
  s_id int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
  s_name varchar(60) NOT NULL,
  gender char(10) NOT NULL,
  dob date NOT NULL,
  address varchar(100) NOT NULL,
  phone varchar(50) NOT NULL,
  e_id varchar(70) NOT NULL,
  s_cred int(10)NOT NULL,
  PRIMARY KEY (s_id)
);
```

```
CREATE TABLE IF NOT EXISTS dept (
 dept_id int(10) unsigned NOT NULL AUTO_INCREMENT,
  dept_name varchar(50) NOT NULL,
  PRIMARY KEY (dept_id)
);
```

```
CREATE TABLE IF NOT EXISTS teacher (
  teacher_id int(10) unsigned NOT NULL AUTO_INCREMENT,
  t_name varchar(30) NOT NULL,
  gender char(10) NOT NULL,
  dob date NOT NULL,
  address varchar(100) NOT NULL,
  phone varchar(50) NOT NULL,
  e_id varchar(70) NOT NULL,
  t_dep int(10) unsigned NOT NULL,
  salary float NOT NULL,
  PRIMARY KEY (teacher_id),
  FOREIGN KEY (t_dep)
  REFERENCES dept (dept_id)

);
```

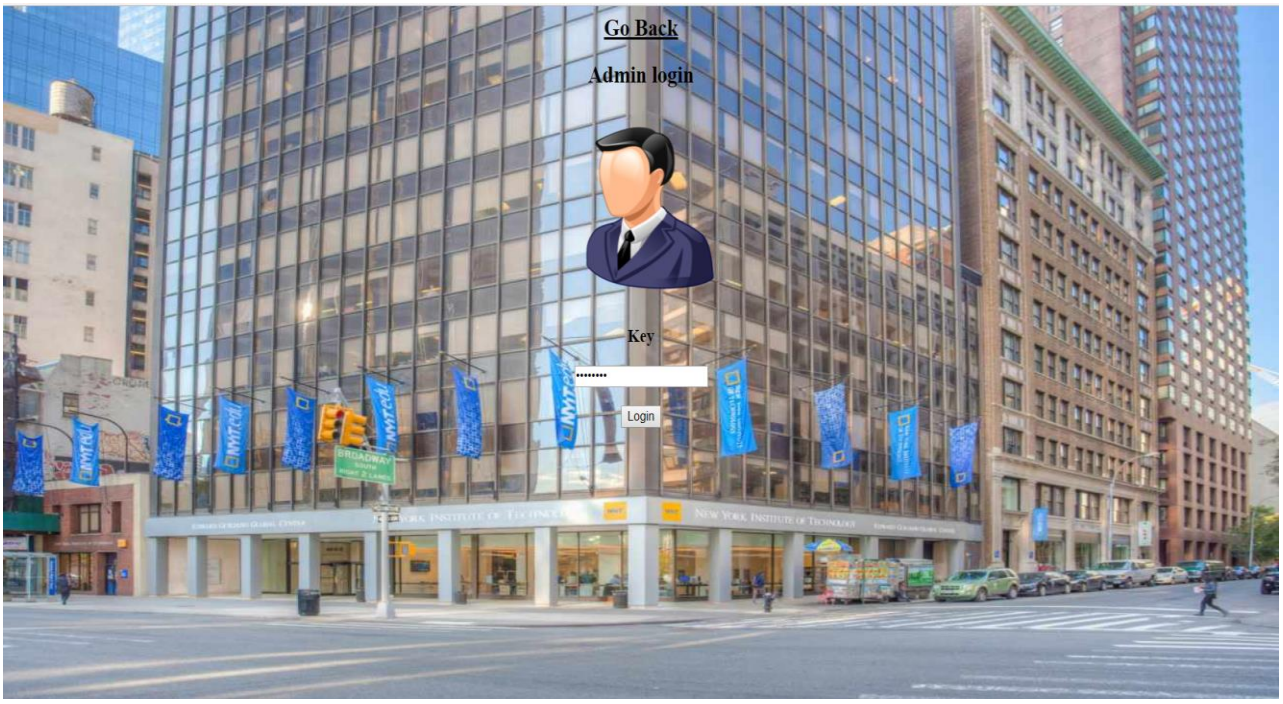
```
CREATE TABLE IF NOT EXISTS course (
  course_id int(10) unsigned NOT NULL AUTO_INCREMENT,
  c_name varchar(50) NOT NULL,
  c_cred int NOT NULL,
  c_sem int NOT NULL,
  c_teach int(10) unsigned NOT NULL,
  PRIMARY KEY (course_id),
  FOREIGN KEY (c_teach)
  REFERENCES teacher (teacher_id)
);
```

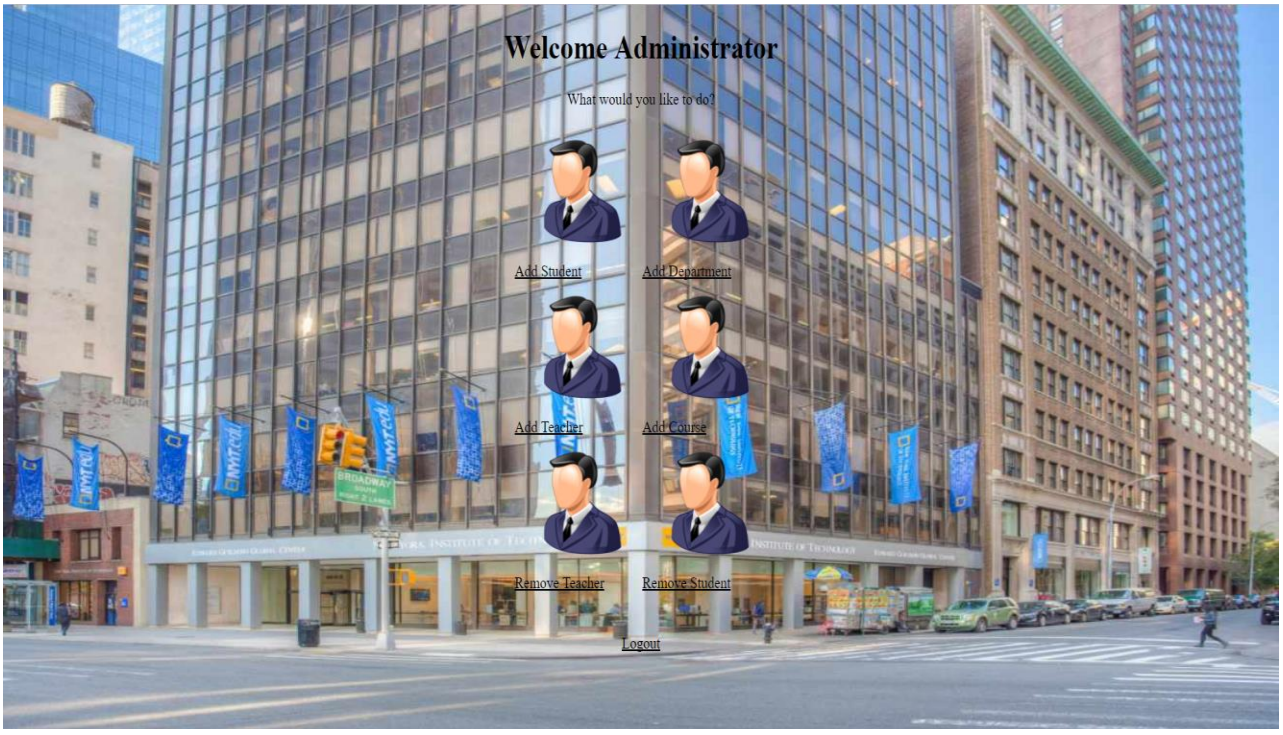
CREATE TABLE IF NOT EXISTS enrolled (
std_id int(10) unsigned NOT NULL,
c_id int(10) unsigned NOT NULL,
FOREIGN KEY (std_id)
  REFERENCES stud_info (s_id),
FOREIGN KEY (c_id)
  REFERENCES course (course_id)
);

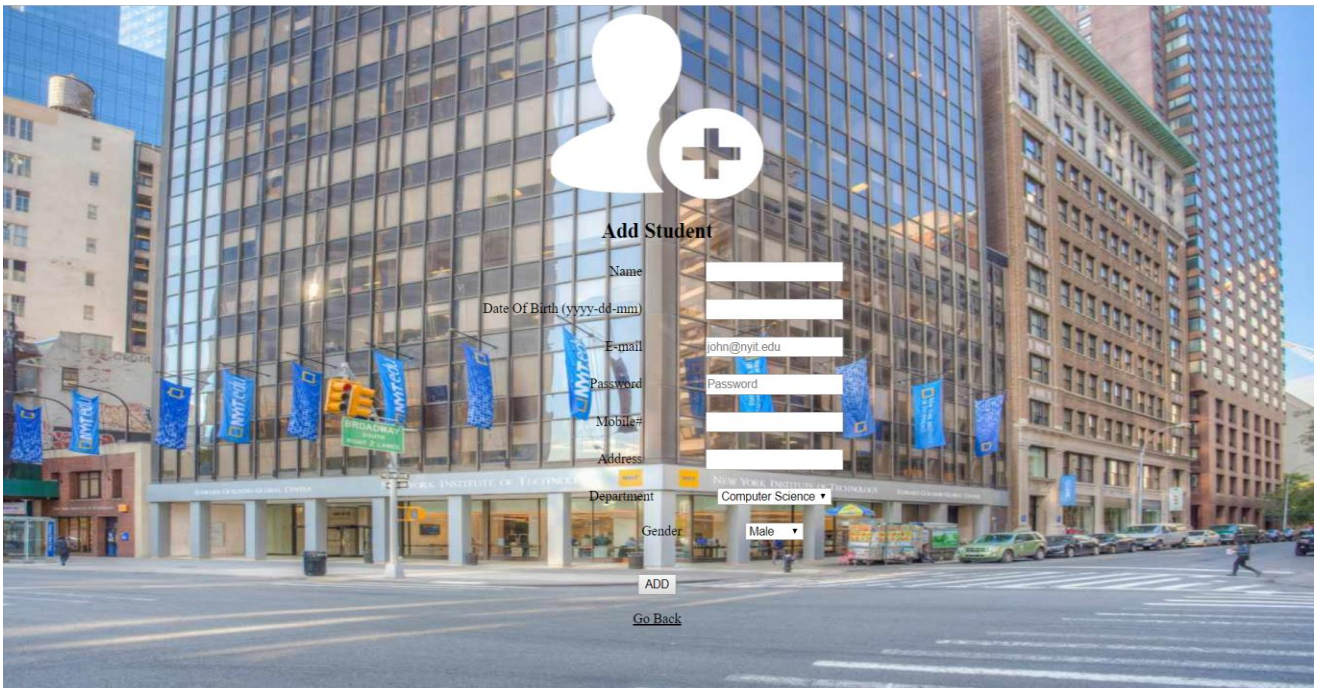
CREATE TABLE IF NOT EXISTS course_dep (
c_id int(10) unsigned NOT NULL,
dep_id int(10) unsigned NOT NULL,
FOREIGN KEY (c_id)
  REFERENCES course (course_id),
FOREIGN KEY (dep_id)
  REFERENCES dept (dept_id)
);

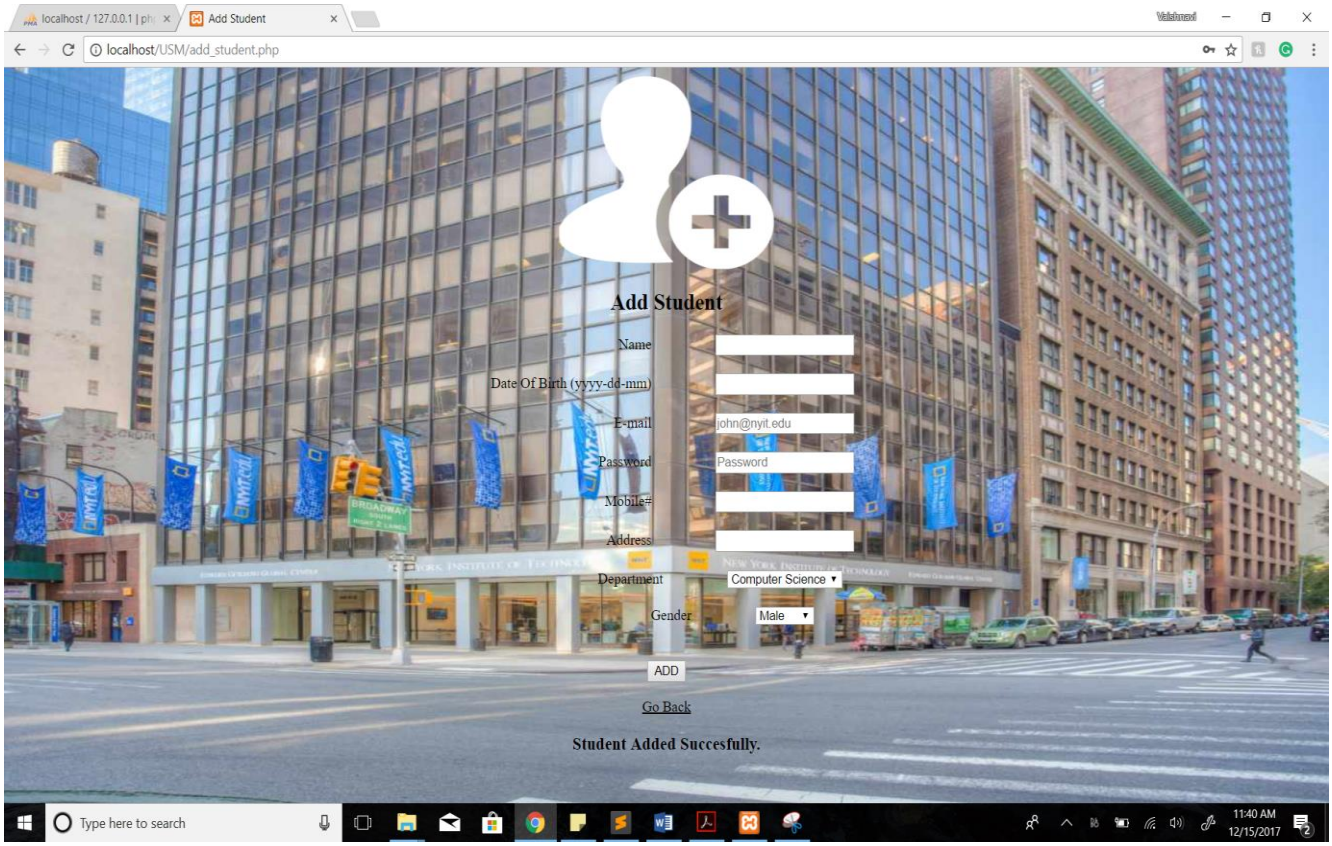
9. Application Design AND Snapshot of Application

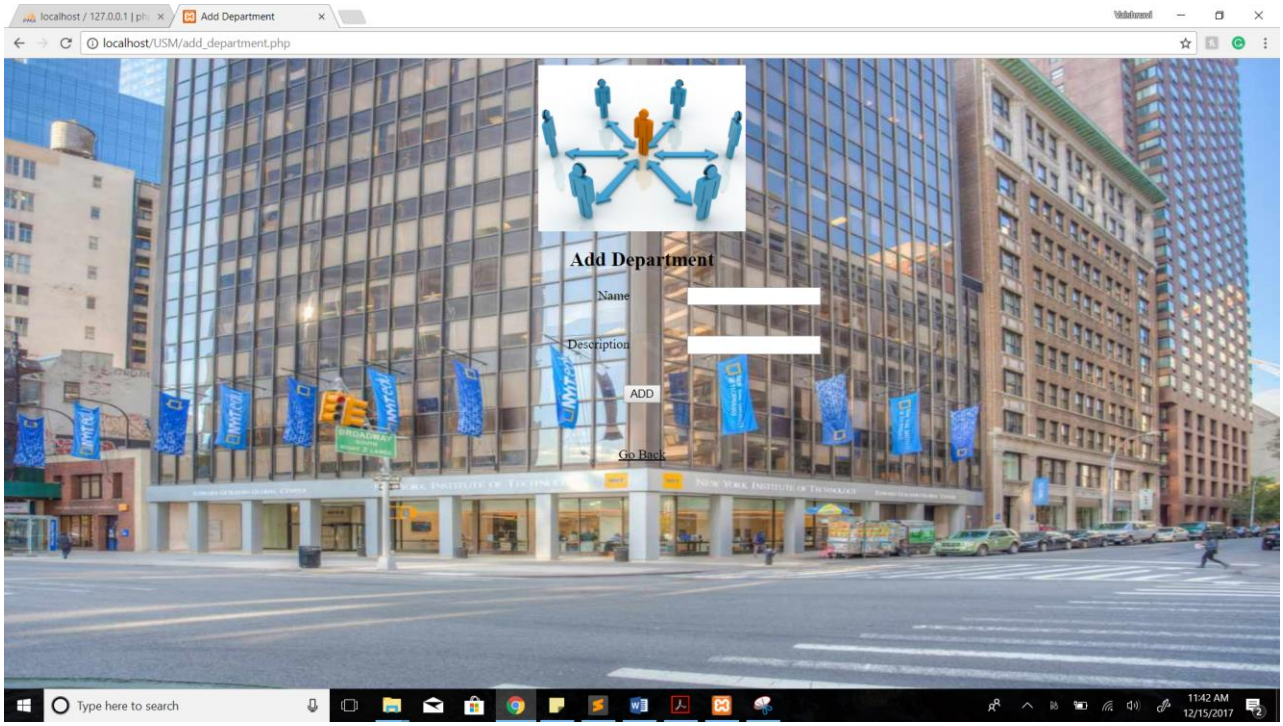
• A description of the application programs and all tasks.
Application Program and all tasks:

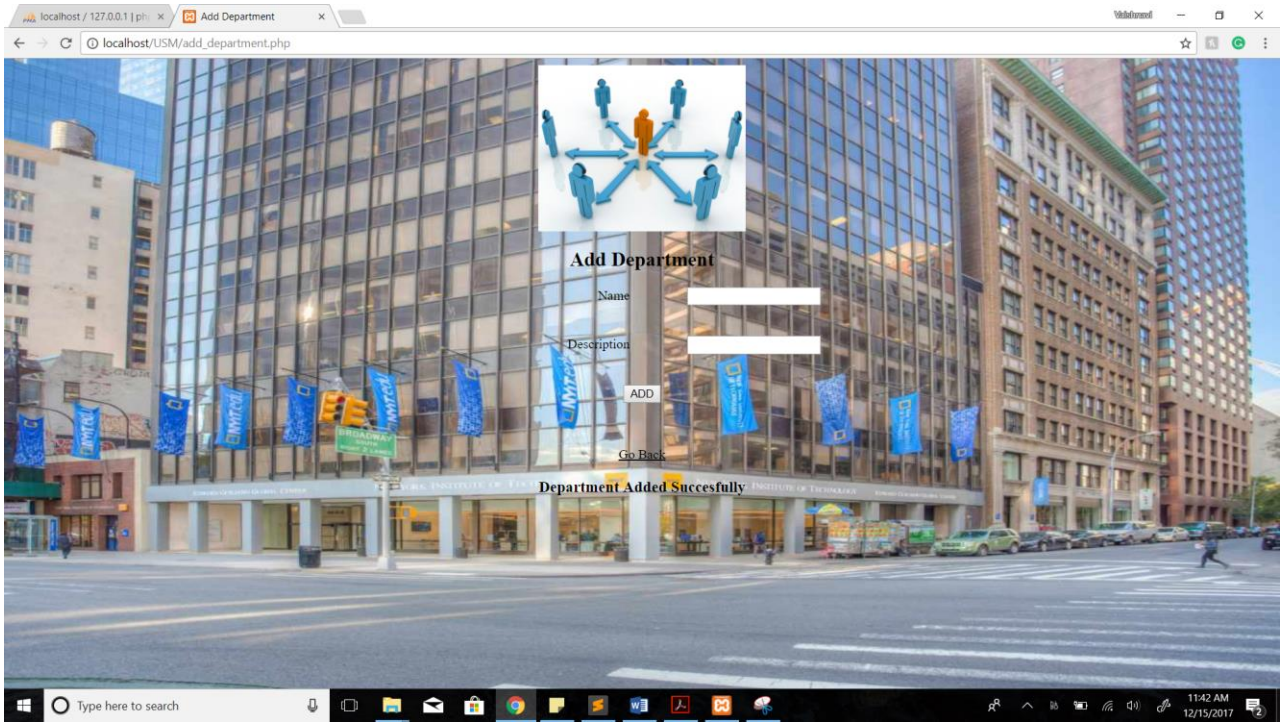

Go Back
Admin login
Key
Login

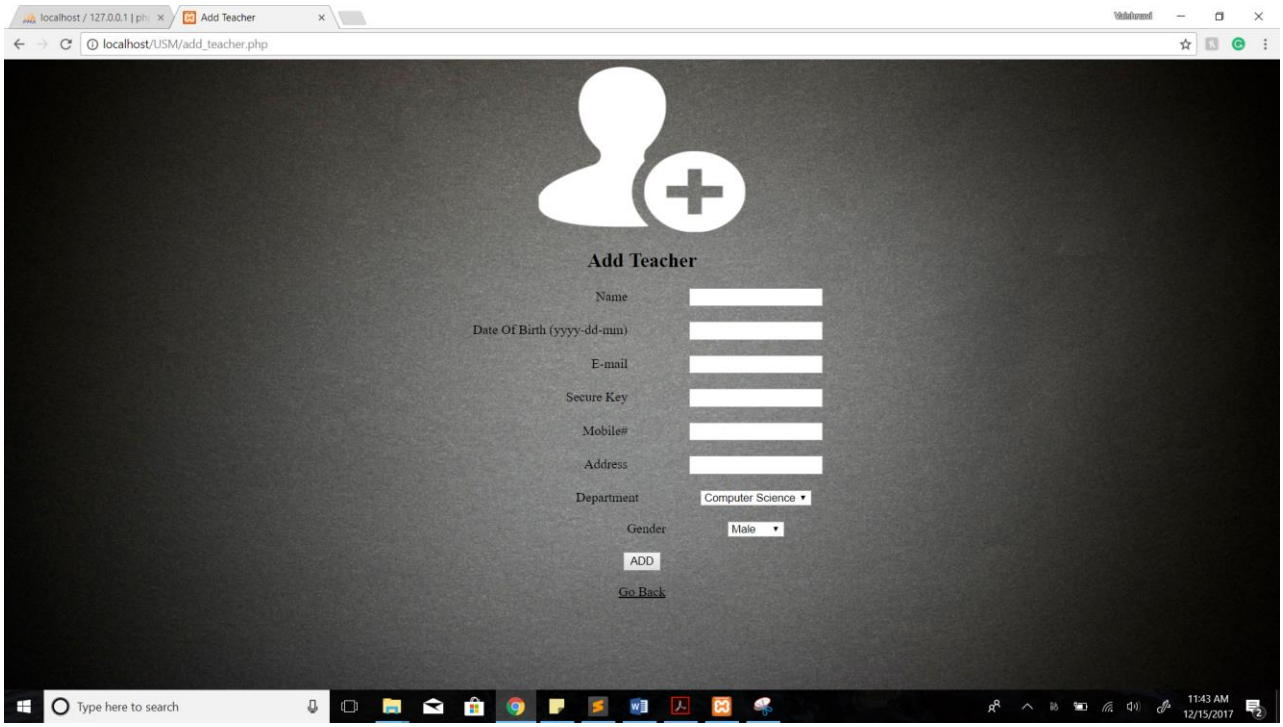

Welcome Administrator
What would you like to do?
Add Student
Add Department
Add Teacher
Add Course
Remove Teacher
Remove Student
Logout

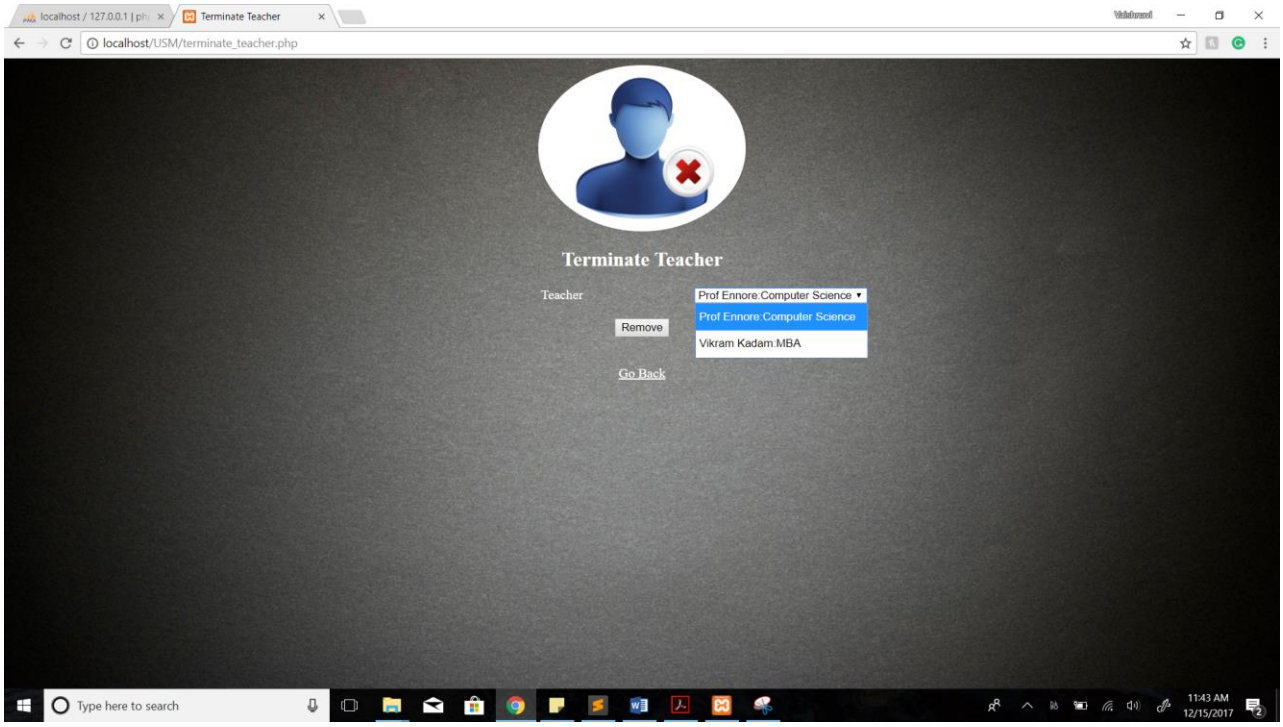

Add Student
Name
Date Of Birth (yyyy-dd-mm)
E-mail
john@nyit.edu
Password
Password
Mobile#
Address
Department
Computer Science
Gender
Male
ADD
Go Back

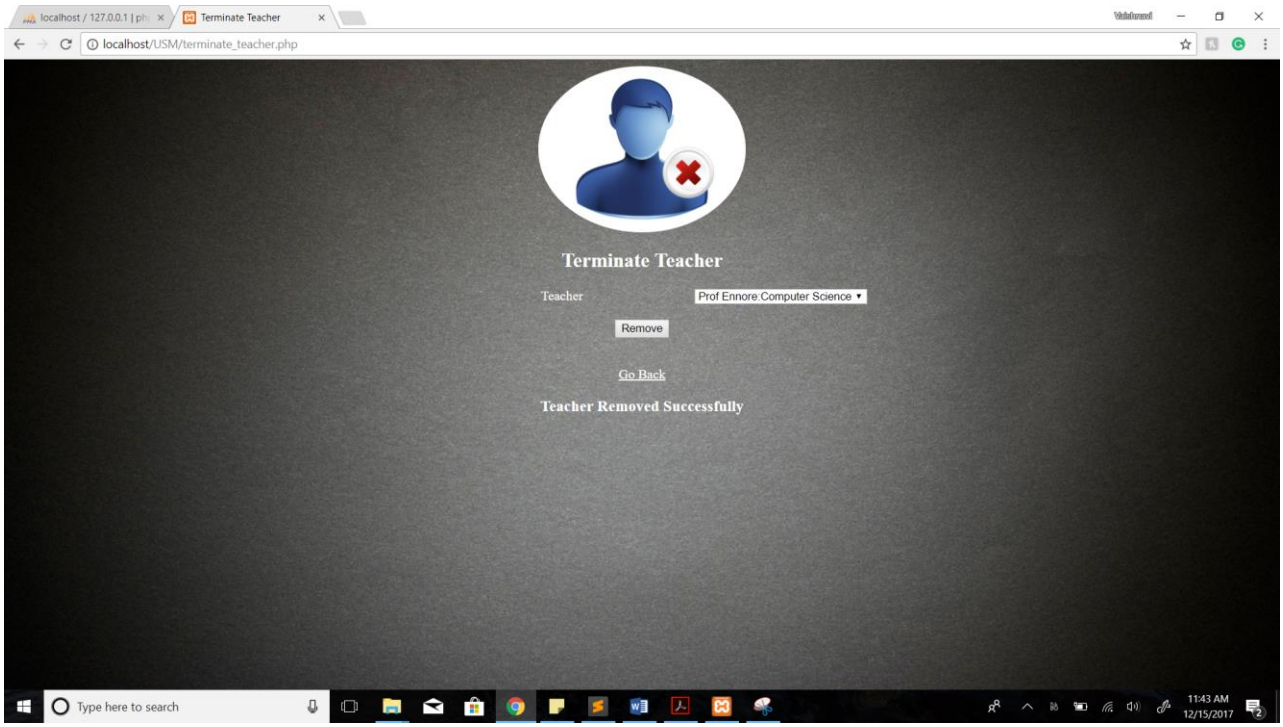

localhost / 127.0.0.1 | ph
Add Student
localhost/USM/add_student.php
Add Student
Name
Date Of Birth (yyyy-dd-mm)
E-mail
john@nyit.edu
Password
Password
Mobile#
Address
Department
Computer Science
Gender
Male
ADD
Go Back
Student Added Succesfully.
Type here to search
11:40 AM
12/15/2017

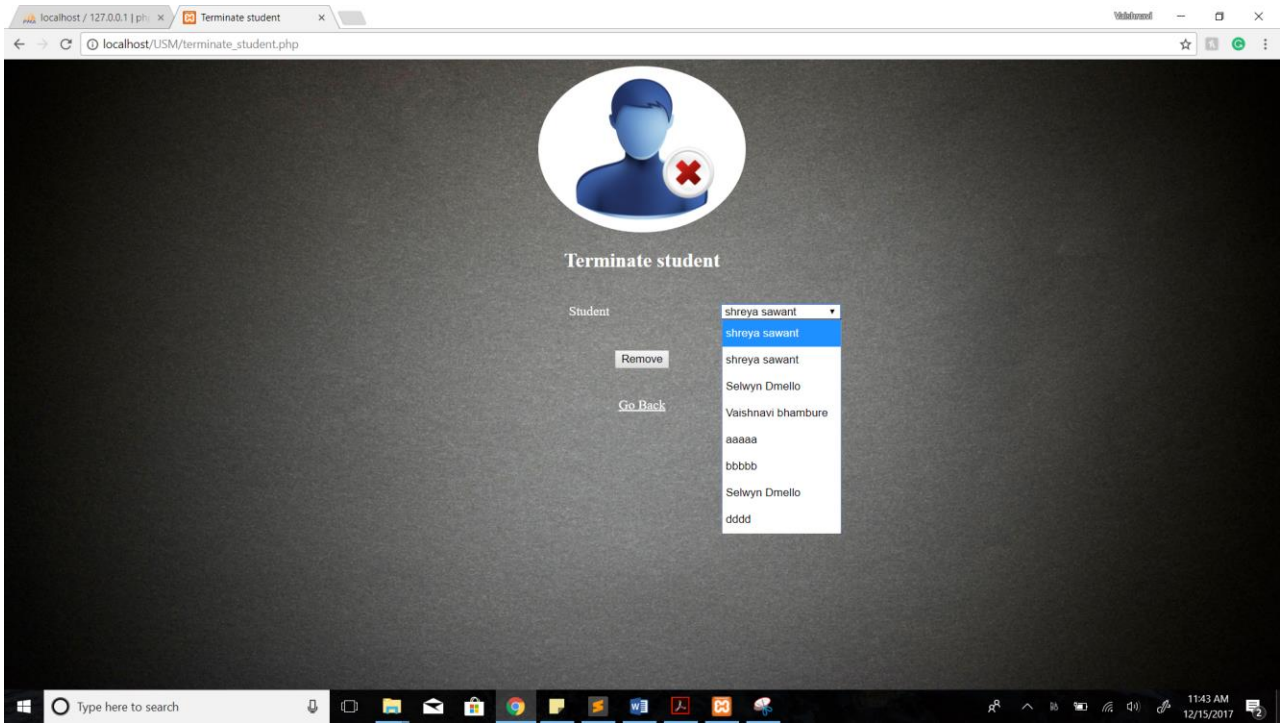

localhost / 127.0.0.1 | ph
Add Department
localhost/USM/add_department.php
Add Department
Name
Description
ADD
Go Back
Type here to search
11:42 AM
12/15/2017

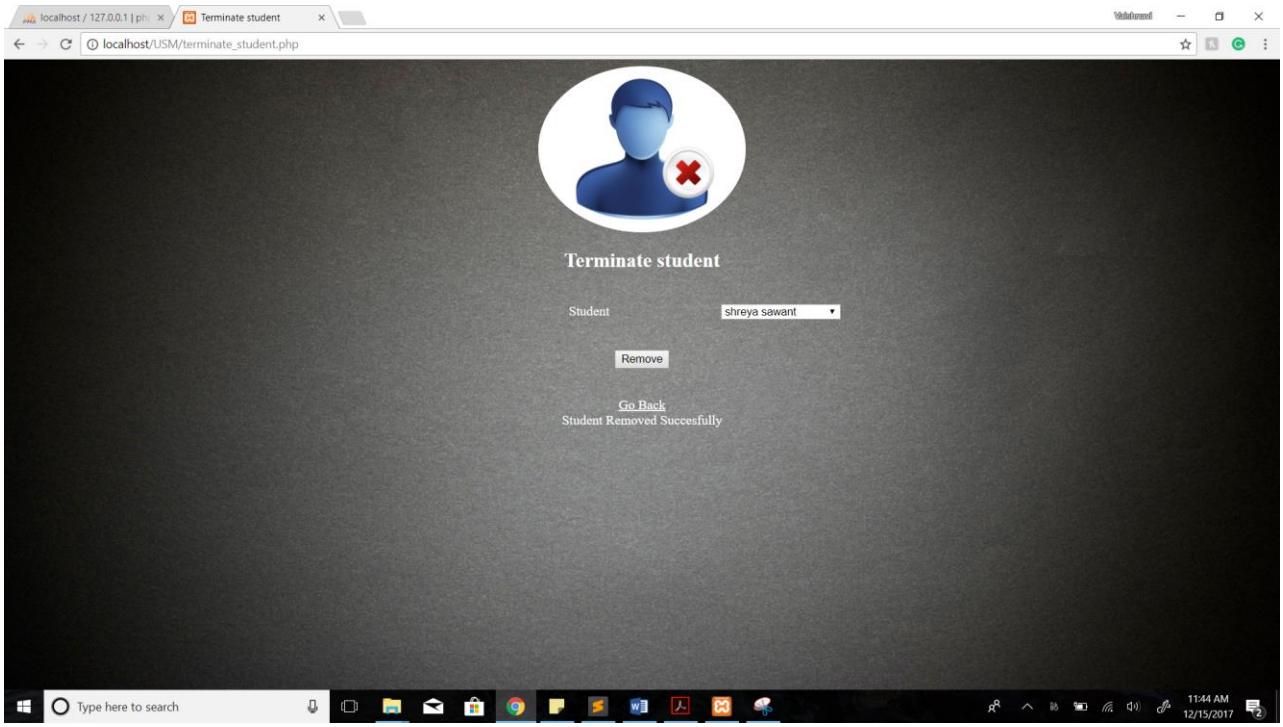

localhost / 127.0.0.1 | ph
Add Department
localhost/USM/add_department.php
Add Department
Name
Description
ADD
Go Back
Department Added Succesfully
Type here to search
11:42 AM
12/15/2017

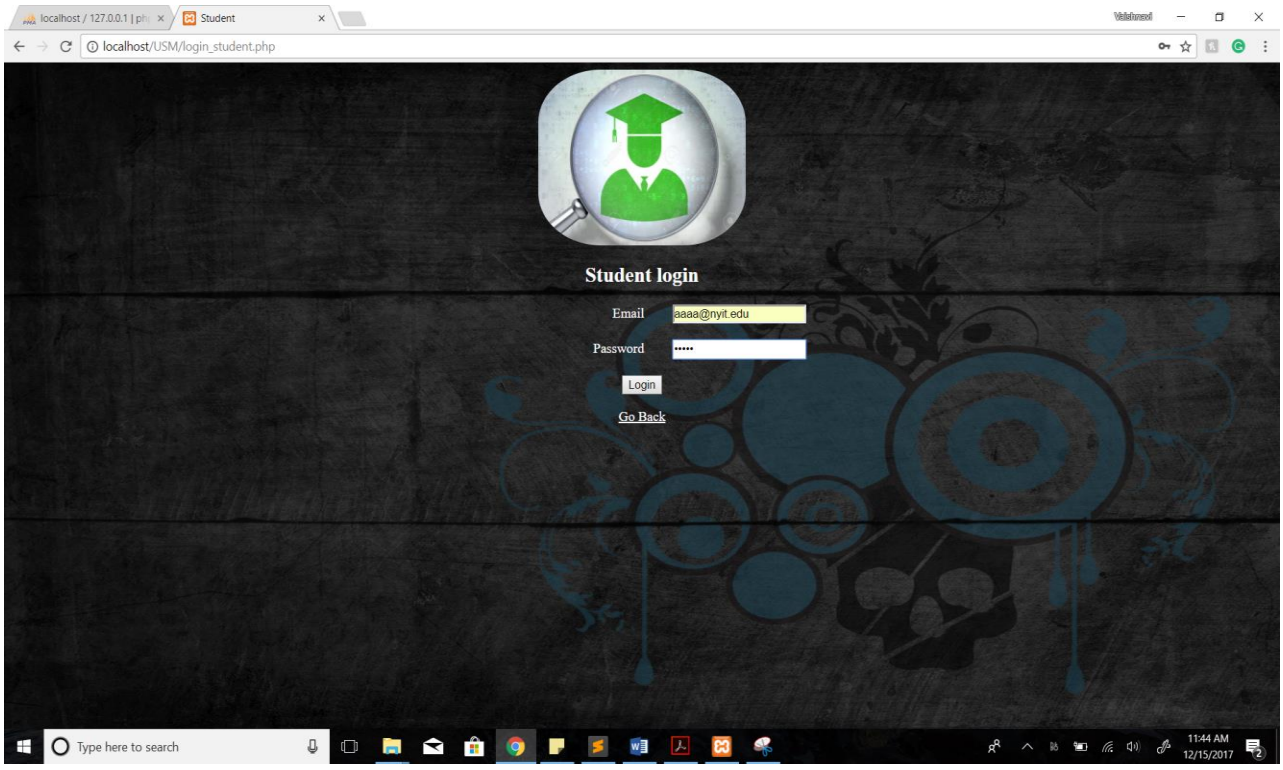

localhost / 127.0.0.1 | ph
Add Teacher
localhost/USM/add_teacher.php
Add Teacher
Name
Date Of Birth (yyyy-dd-mm)
E-mail
Secure Key
Mobile#
Address
Department
Computer Science
Gender
Male
ADD
Go Back
Type here to search
11:43 AM
12/15/2017

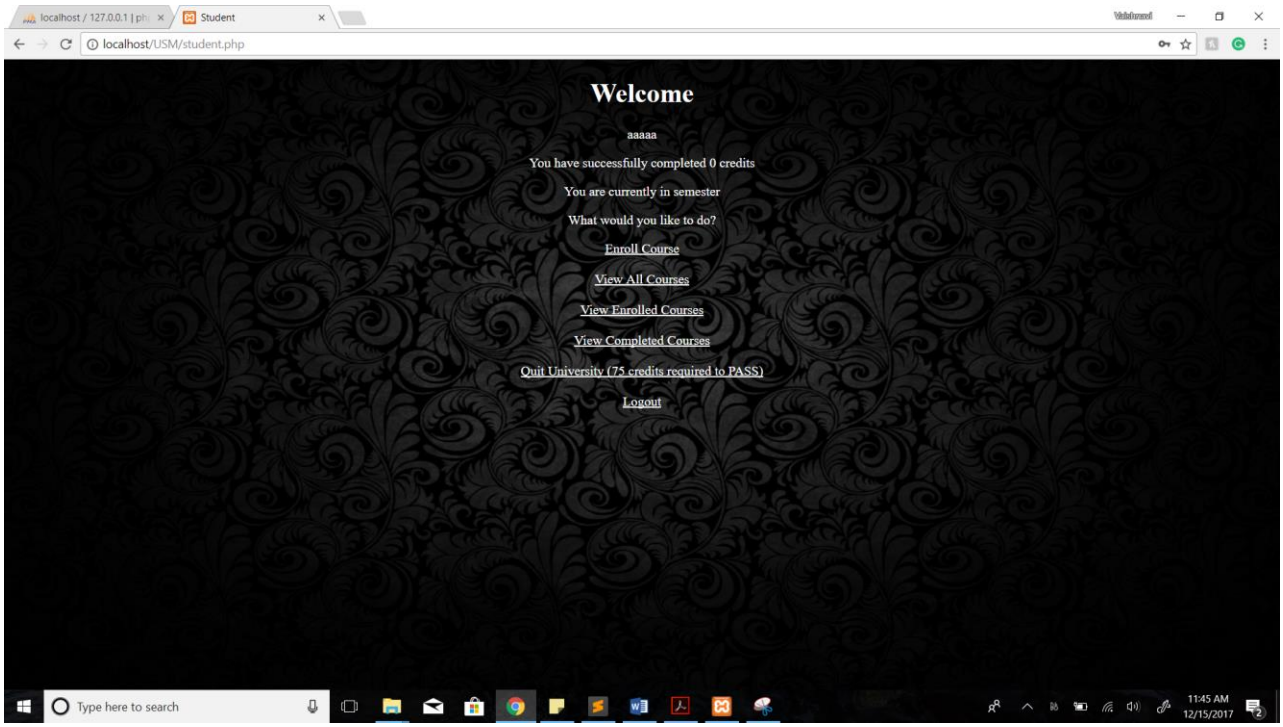

localhost / 127.0.0.1 | ph
Terminate Teacher
localhost/USM/terminate_teacher.php
Terminate Teacher
Teacher
Prof Ennore:Computer Science
Prof Ennore:Computer Science
Vikram Kadam:MBA
Remove
Go Back
Type here to search
11:43 AM
12/15/2017

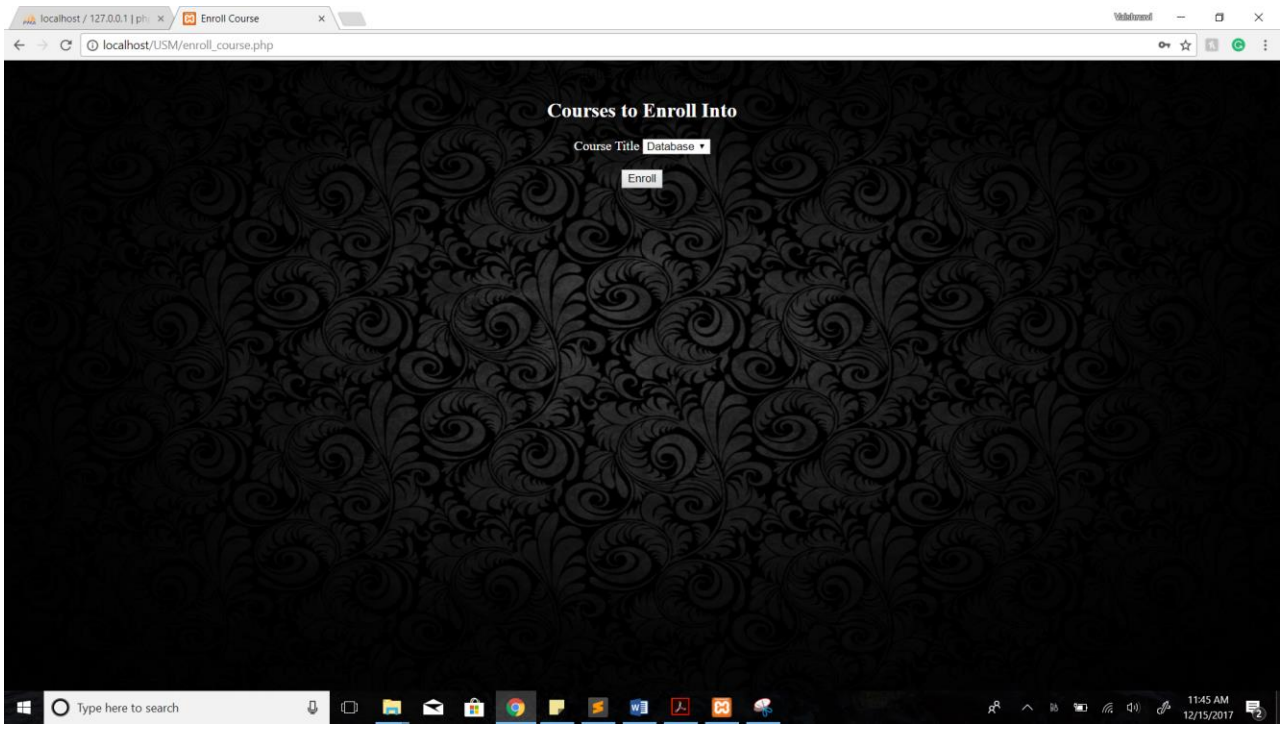

localhost / 127.0.0.1 | ph
Terminate Teacher
localhost/USM/terminate_teacher.php
Terminate Teacher
Teacher
Prof Ennore:Computer Science
Remove
Go Back
Teacher Removed Successfully
Type here to search
11:43 AM
12/15/2017

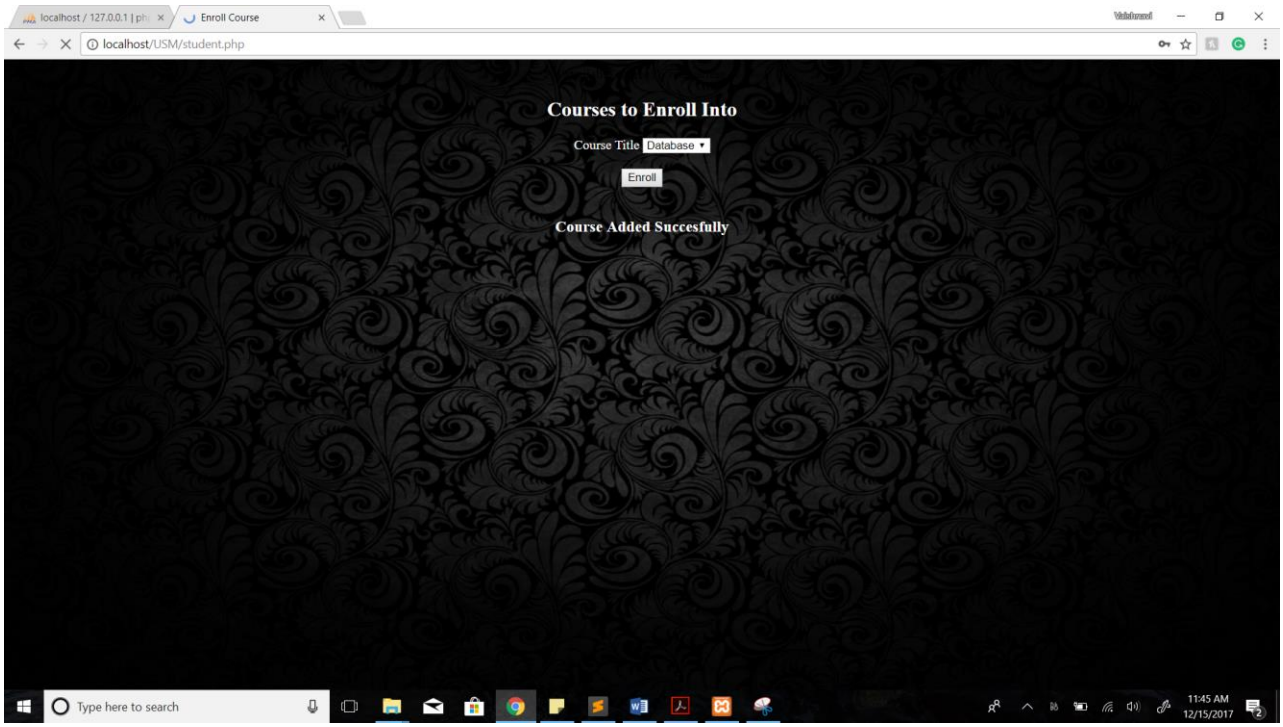

localhost / 127.0.0.1 | ph
Terminate student
localhost/USM/terminate_student.php
Terminate student
Student
shreya sawant
shreya sawant
shreya sawant
Selwyn Dmello
Vaishnavi bhambure
aaaaa
bbbbb
Selwyn Dmello
dddd
Remove
Go Back
Type here to search
11:43 AM
12/15/2017

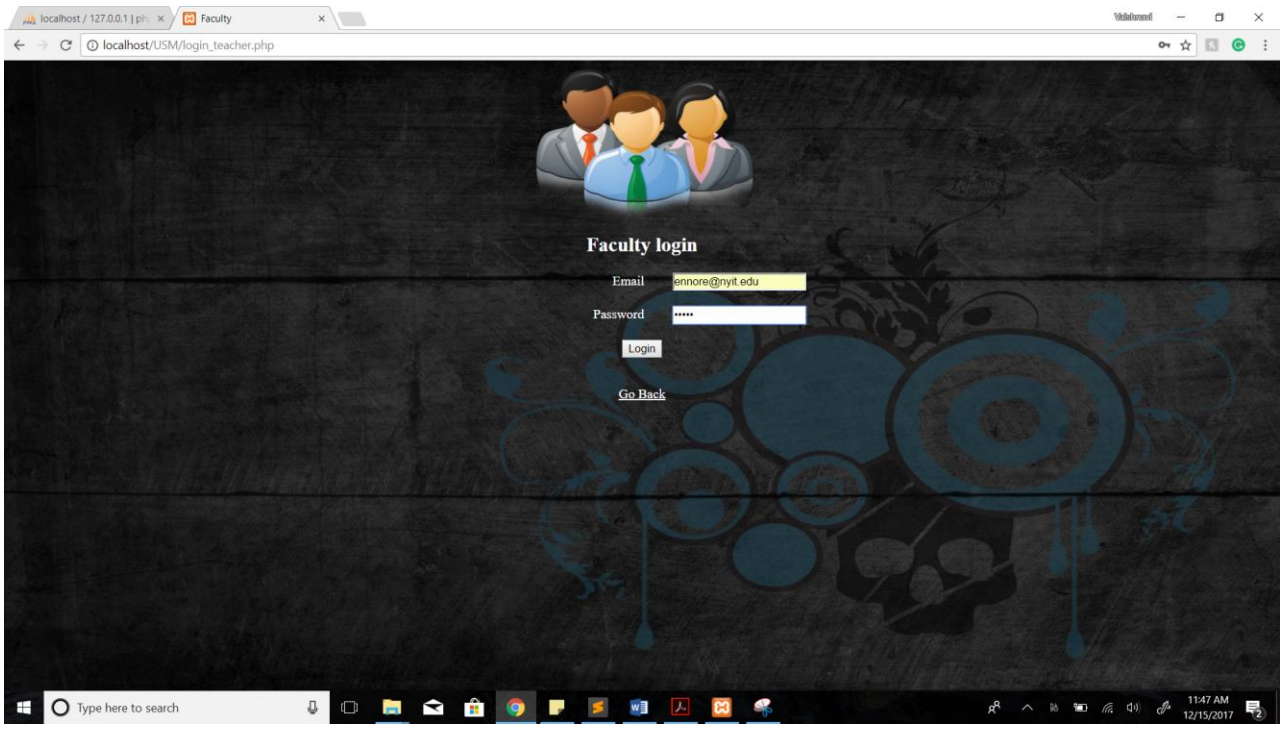

localhost / 127.0.0.1 | ph
Terminate student
localhost/USM/terminate_student.php
Terminate student
Student
shreya sawant
Remove
Go Back
Student Removed Succesfully
Type here to search
11:44 AM
12/15/2017

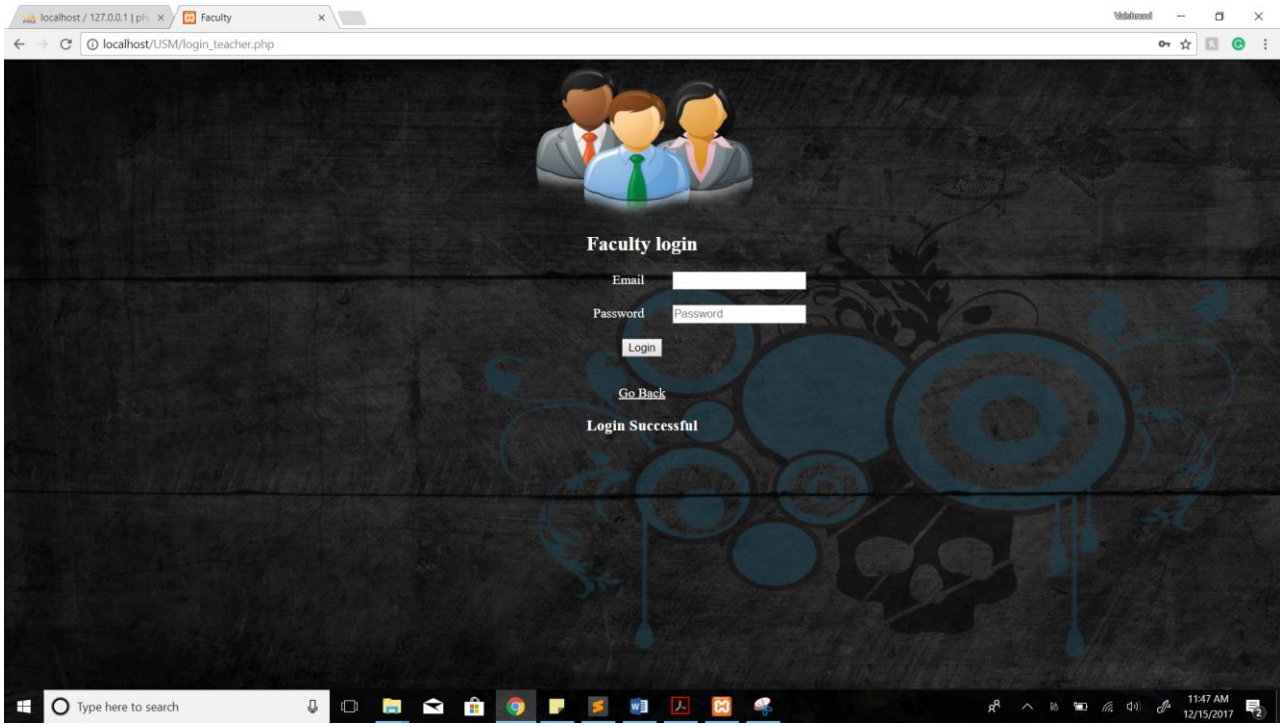

localhost / 127.0.0.1 | ph
Student
localhost/USM/login_student.php
Student login
Email
aaaa@nyit.edu
Password
Login
Go Back
Type here to search
11:44 AM
12/15/2017

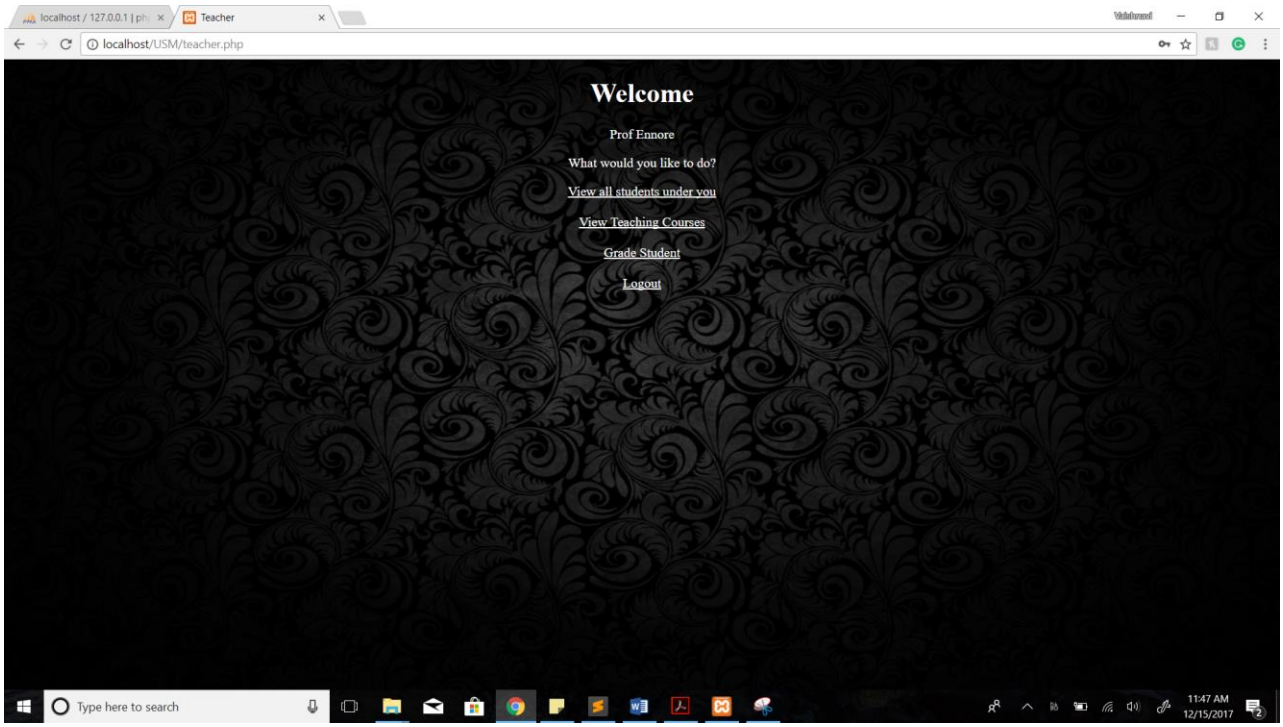

localhost / 127.0.0.1 | ph
Student
localhost/USM/student.php
Welcome
aaaaa
You have successfully completed 0 credits
You are currently in semester
What would you like to do?
Enroll Course
View All Courses
View Enrolled Courses
View Completed Courses
Quit University (75 credits required to PASS)
Logout
Type here to search
11:45 AM
12/15/2017

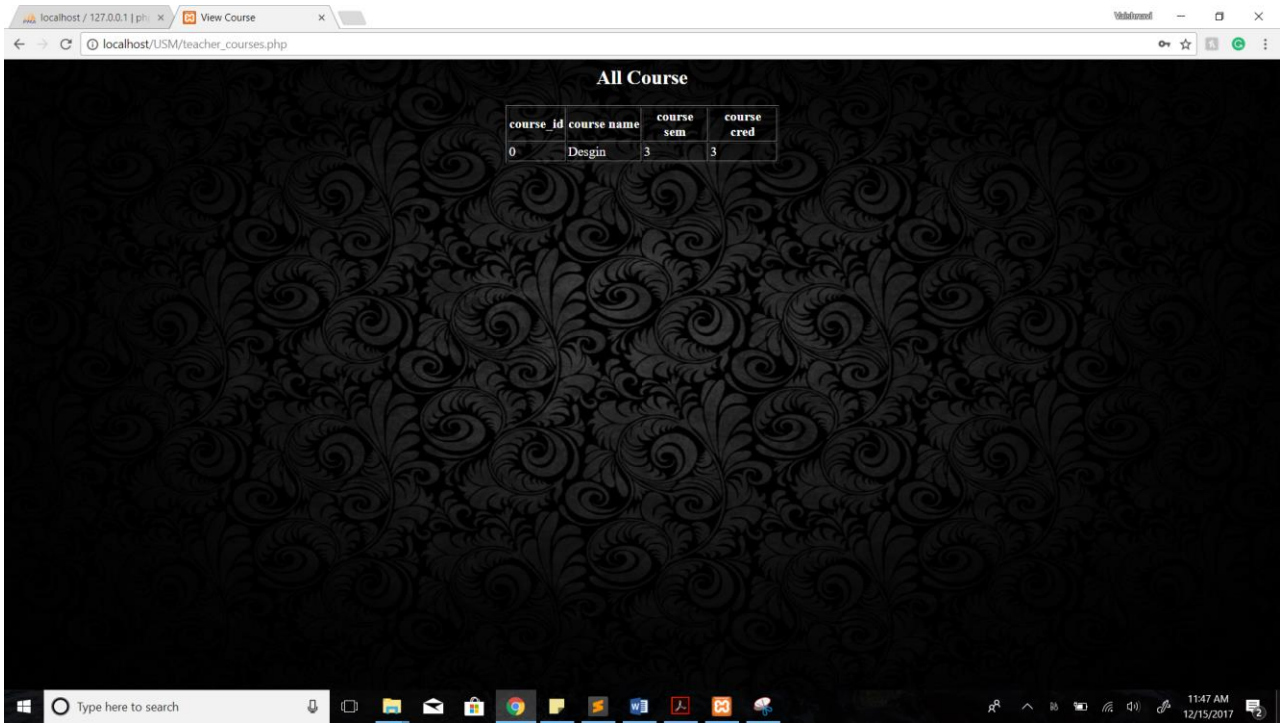

localhost / 127.0.0.1 | ph
Enroll Course
localhost/USM/enroll_course.php
Courses to Enroll Into
Course Title Database
Enroll
Type here to search
11:45 AM
12/15/2017

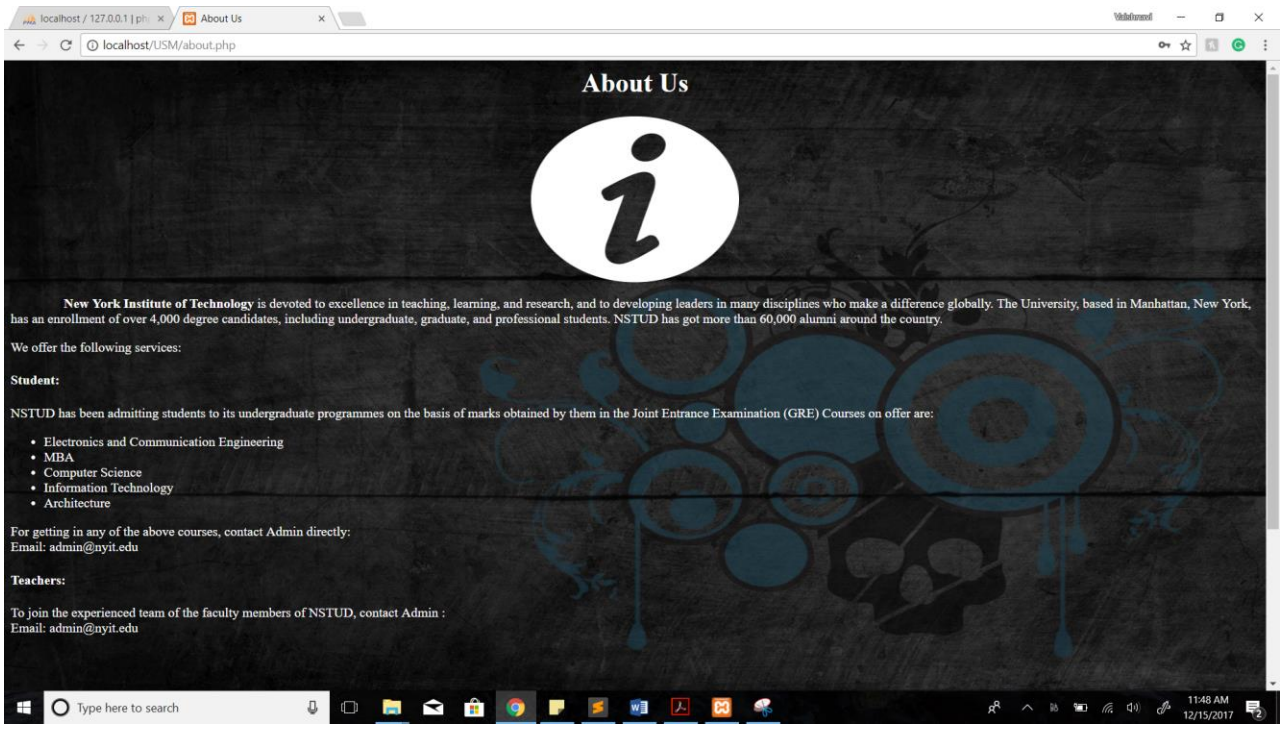

localhost / 127.0.0.1 | ph
Enroll Course
localhost/USM/student.php
Courses to Enroll Into
Course Title Database
Enroll
Course Added Succesfully
Type here to search
11:45 AM
12/15/2017


localhost / 127.0.0.1 | ph
Faculty
localhost/USM/login_teacher.php
Faculty login
Email ennore@nyit.edu
Password
Login
Go Back
Type here to search
11:47 AM
12/15/2017


localhost / 127.0.0.1 | ph
Faculty
localhost/USM/login_teacher.php
Faculty login
Email
Password
Password
Login
Go Back
Login Successful
Type here to search
11:47 AM
12/15/2017


localhost / 127.0.0.1 | ph
Teacher
localhost/USM/teacher.php
Welcome
Prof Ennore
What would you like to do?
View all students under you
View Teaching Courses
Grade Student
Logout
Type here to search
11:47 AM
12/15/2017


localhost / 127.0.0.1 | ph
View Course
localhost/USM/teacher_courses.php
All Course
course_id
course name
course sem
course cred
0
Desgin
3
3
Type here to search
11:47 AM
12/15/2017


localhost / 127.0.0.1 | ph
About Us
localhost/USM/about.php
About Us
New York Institute of Technology is devoted to excellence in teaching, learning, and research, and to developing leaders in many disciplines who make a difference globally. The University, based in Manhattan, New York, has an enrollment of over 4,000 degree candidates, including undergraduate, graduate, and professional students. NSTUD has got more than 60,000 alumni around the country.
We offer the following services:
Student:
NSTUD has been admitting students to its undergraduate programmes on the basis of marks obtained by them in the Joint Entrance Examination (GRE) Courses on offer are:
Electronics and Communication Engineering
MBA
Computer Science
Information Technology
Architecture
For getting in any of the above courses, contact Admin directly:
Email: admin@nyit.edu
Teachers:
To join the experienced team of the faculty members of NSTUD, contact Admin :
Email: admin@nyit.edu
Type here to search
11:48 AM
12/15/2017

10. User Guide

Integrated University Department Information System offers many flexible and convenient features, allowing admin, student and teacher to maximize time and efficiency. IUDIS gives the all detailed information about student and teacher. It will track how many class are available in university and whether that particular class is filled or not. It will provide the total student count in a University. It keeps the record of the students and teachers. Software is customizing for any IUDIS.

Following is a brief description about the working of the system:-

- Homepage – Contains Admin, Teacher and Student module which allows to authenticate user and allow them to login into their module
- Add a student - Allows the Admin/Teacher to add student in particular department.
- Add a Teacher – Allows the Admin to add Teacher in particular department.
- Add class- Allows admin to add class in particular department.
- Calculate Fees - This feature allows to calculate fees for student.

11. Specify in detail each work:

Modules / **Team Members Name**	**Database Design**	**Project Report**	**Implementation and coding**
Vaishnavi Bhambure (1156082)	-Schema Mapping -Normalization - Design of ER -Diagram -Schema Mapping	-Introduction -Relational Database Design -ER data Model Design -Logical Design of Database	-Database Connection -Implementation/ Development of Modules -Query Design -Form Design -Implementation and synchronization with database design