

PRAIRIE VIEW A&M UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE  
**CINS 5143 ADVANCED DATABASE  
MANAGEMENT SYSTEMS**



**Project Paper CarGurus  
(Car Rental)**

**By:**

Purviben Anghan  
Thomas Okonkwo  
Olubunmi Adekambi  
Nneoma Okoroafor

**To:**

Dr. Mary Kim

November 23, 2021

## Table of Contents

1. Introduction .....	1
2. Mission Statements .....	2
2.1 Purpose of the system .....	2
2.2 New service development- Features .....	2
3. Overview of the system .....	3
4. System architecture .....	4
5. Requirements of the Database System.....	7
6.1 Hardware .....	7
6.2 Software .....	7
6.3 Functional Requirements.....	7
User should be able to access the home page and enter student ID to search.....	7
User should be able to select prerequisites from the list of prerequisites .....	7
User should be able to access registration form and fill in all required details .....	7
User should be able to submit form which will persist on the DB.....	7
User should be able to retrieve information using the studentID from the DB.....	7
Course Advisor/Admin should be able to modify records pertaining to Student information, building information, lectures information and main courses and prerequisites .....	7
6.4 Non-functional Requirements .....	7
7. Implementation method .....	10
8. Main queries.....	10
9. Milestone for the Project.....	12
10.Discussion and conclusion .....	12
11.Group statement.....	13
11.1 Things learnt from the project.....	13
11.2 Team Member Roles.....	13
12.References .....	14
APPENDIX .....	15

## List of Tables:

<a href="#">Table 1:Tables and Constraints</a> .....	7
--	---

## Table of Figures:

<a href="#">Figure 1: System Architecture</a> .....	3
<a href="#">Figure 2: Use Case Diagram</a> .....	4
<a href="#">Figure 3: Database System Boundary</a> .....	4
<a href="#">Figure 5: ER diagram</a> .....	7
<a href="#">Figure 6: Screenshot1</a> .....	8
<a href="#">Figure 7: Screenshot2</a> .....	8

## **1. Introduction**

Course advising is an important and time-consuming effort in academic life. Course advising has been implemented in order to fill the gap between student and the academic routine, by moving advising, evaluating, suggesting system from the traditional ways to an automated way. The aim of this paper is to implement a system which facilitates and assists academic advisors in their efforts to providing quality, accurate and consistent advising services to their students.

## **2. Mission Statements**

### ***2.1 Purpose of the system***

The purpose of the course advisor database is to reduce/eliminate interactions with student coordinators during course registration by consolidating, maintaining and suggesting suitable courses for the semester to students in Computer Science and Computer Information Systems.

### ***2.2 New service development- Features***

- A web-based system that will help manage registration of students in Computer Science and Computer Information Systems.
- An interface for Admin to record Student information
- A frontend for students to register courses, search records, check course recommendations and register course
- A consolidate report generation for both Admin and students
- Database to maintain all records

### 3. Overview of the system

The course advisor database is used to track all registration of courses by CS and CIS students, manage all entries by the Admin and provide report on relevant student activities. The database acts as a course registration management system to maintain collated from the Admin and students.

#### 3.1 *Who are the users and their capabilities?*

##### *Students*

- Reading from the table (they are only able to view the records)
- Create course record
- Students should be able to update their records

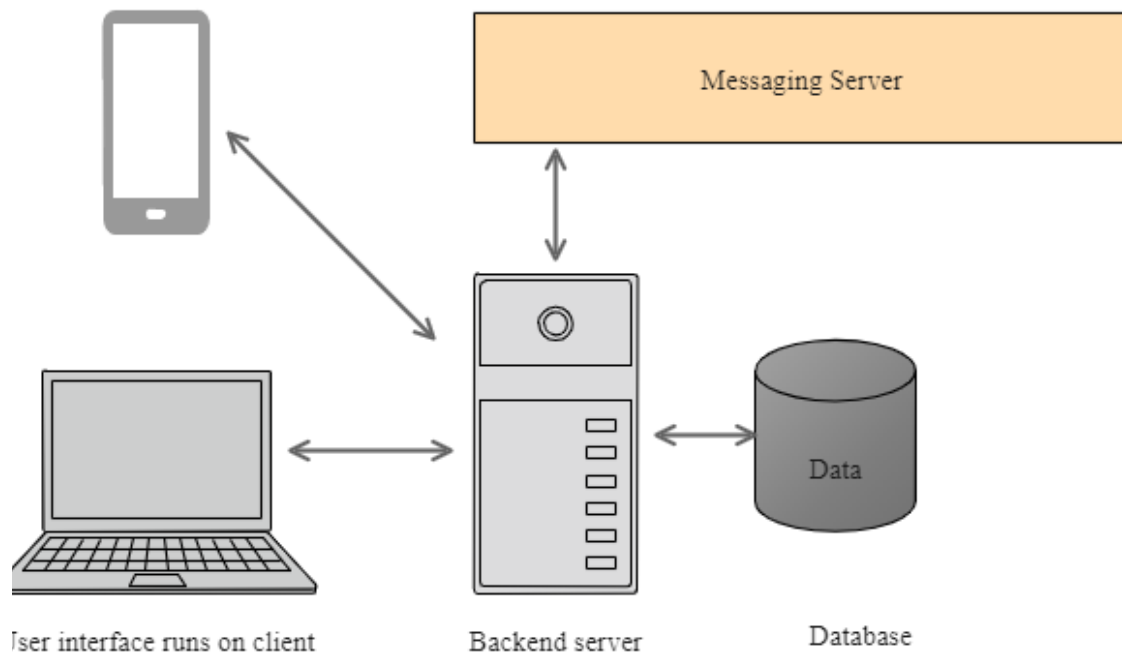
##### *Course coordinator*

- Reading from the table
- Create a new student record in the database
- Modify and update records
- Should be able to delete / render inactive

## 4. System architecture

When customer calls CarGurus, he informs the employee about the type of car he/she is looking for. Based on the given information the employee searches for the car details in the database and gets back to the customer with the available cars. The customer can select the car from the available cars and proceed to book the car or cancel the request.

### System Architecture



**Figure 1: System Architecture**

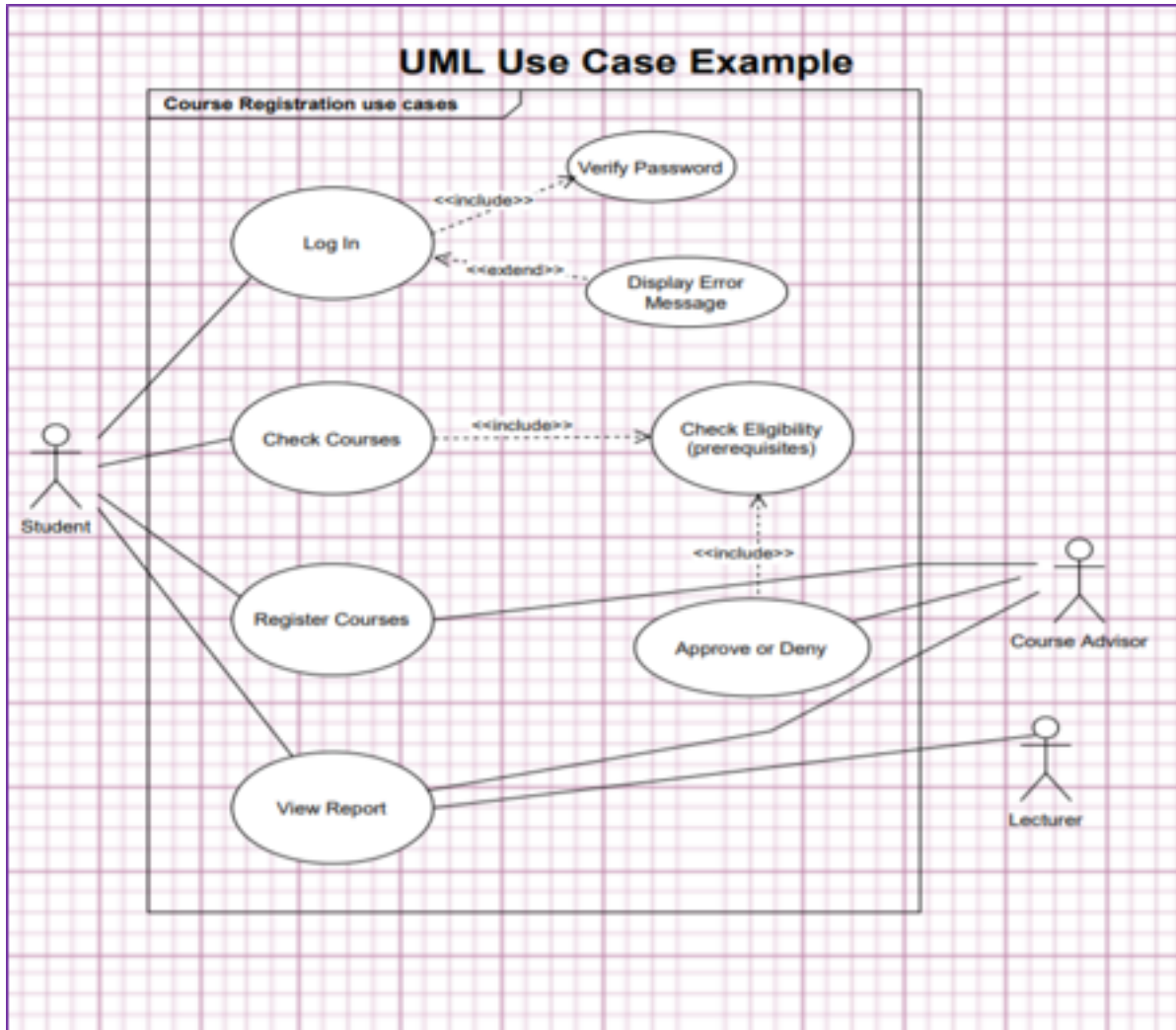


Figure 2: Use Case Diagram



## **5. Requirements of the Database System**

### ***6.1 Hardware***

Operating System: Windows 10

Hard disk: 1TB

RAM: 8 GB

Processor: Intel Core i5

Response/Speed: Internet Connection

### ***6.2 Software***

Operating System Windows 7 or higher

User Interface HTML, CSS

Server: PHP Script

Languages PHP

Web application: XAMPP Server

Database: MySQL Server 2008

### ***6.3 Functional Requirements***

User should be able to access the home page and enter student ID to search

User should be able to select prerequisites from the list of prerequisites

User should be able to access registration form and fill in all required details

User should be able to submit form which will persist on the DB

User should be able to retrieve information using the studentID from the DB

Course Advisor/Admin should be able to modify records pertaining to Student information, building information, lectures information and main courses and prerequisites

### ***6.4 Non-functional Requirements***

System should have the ability to maintain huge number of users without crashing.

System should have speedy performance / transmission of data and display accurate information.

System should have a speedy recovery time if anything goes wrong.

<b>Relations</b>	<b>Primary Key (PK)</b>	<b>Foreign Key (FK)</b>
<b>Student</b>	<b>STUDENTID</b>	
<b>Department</b>	<b>DEPARTMENTCODE</b>	
<b>Staff</b>	<b>STAFFNO</b>	
<b>Building</b>	<b>ROOMNO</b>	<b>COURSECODE</b>
<b>Schedule</b>	<b>STAFFNO</b>	<b>S T A F F N O , R O O M N O , COURSECODE</b>
<b>Course</b>	<b>COURSECODE</b>	
<b>Registration</b>	<b>STUDENTID, COURSECODE</b>	<b>STUDENTID, COURSECODE</b>

**Table 1:Tables and Constraints**

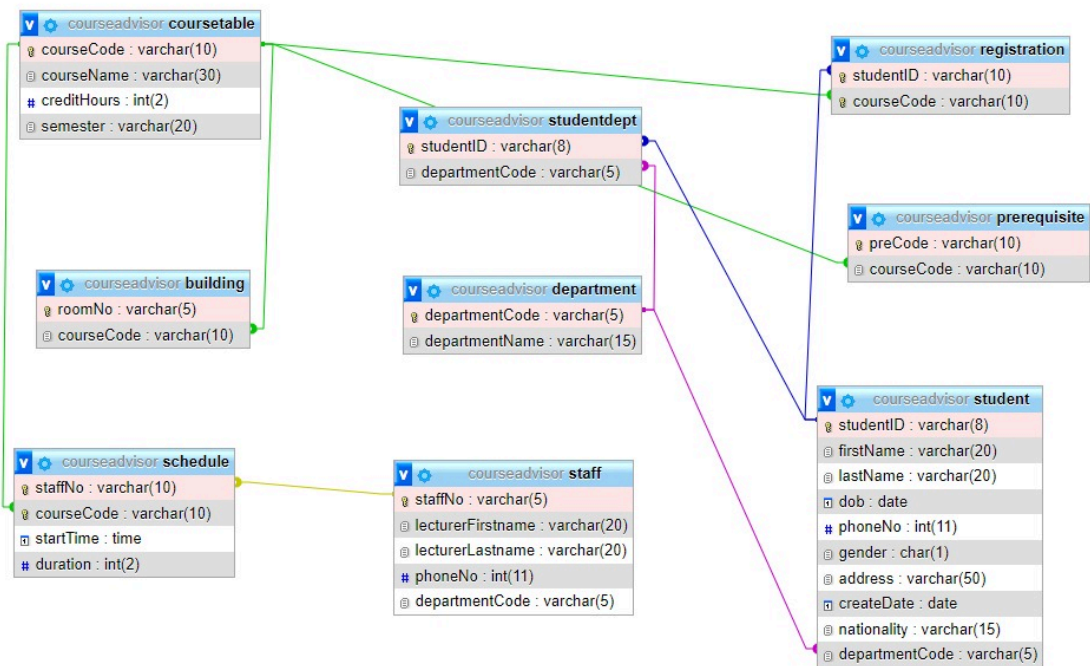


Figure 4: ER diagram

## 7. Implementation method

This project was implemented using Scrum Agile methodology. Here, an iterative approach was used in the design of the system. The functional units were built separately and integrated at the end of the project to meet the project scope within timeline and according to required specification.

## 8. Main queries

The main queries implemented in this project addresses 10 Business questions and they have been highlighted in the appendix of the paper.

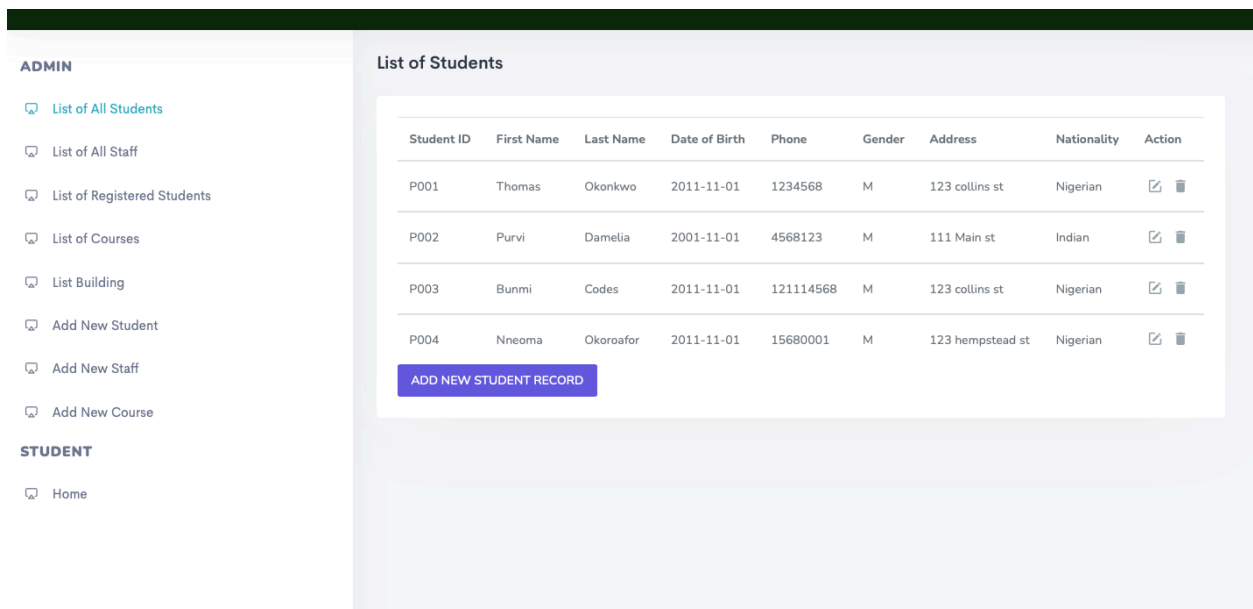


Figure 5: Screenshot1

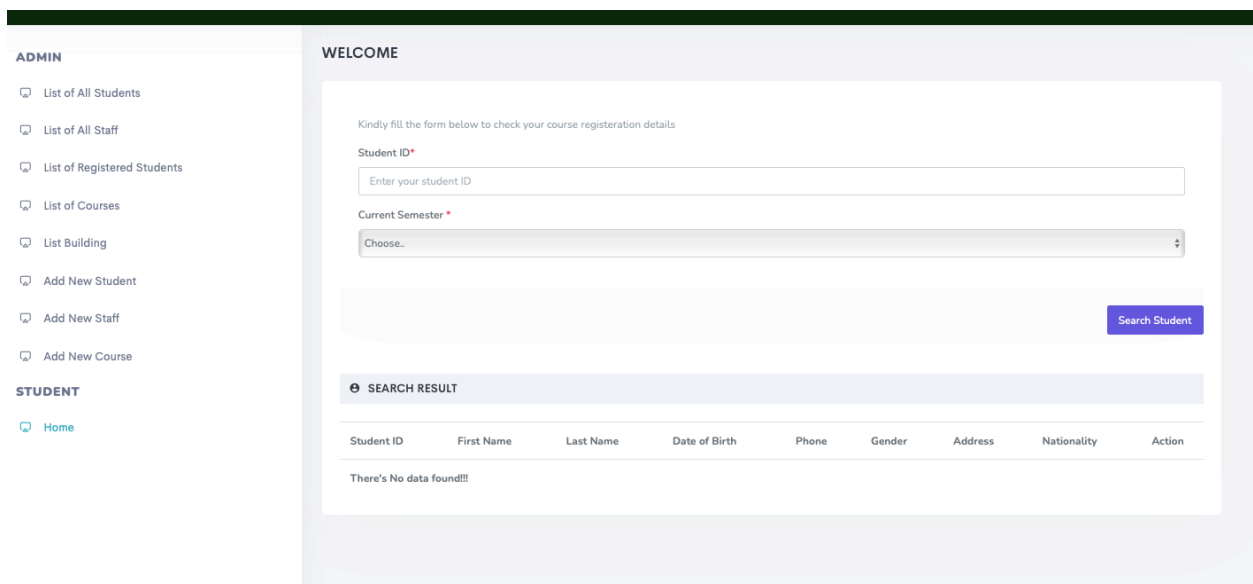


Figure 6: Screenshot2

The screenshot displays a user profile interface. On the left is a sidebar with navigation links under 'ADMIN' and 'STUDENT' categories. The main content area is titled 'WELCOME' and shows the user's name 'Thomas Okonkwo' with a profile picture and ID 'P001'. Below this is a 'PERSONAL INFORMATION' section with fields for Date of Birth, Phone Number, Gender, Nationality, and Address. A 'REGISTERED COURSES' section follows, containing a table with three rows of course data. At the bottom of the courses section is an 'ADD COURSE' button.


**ADMIN**

- List of All Students
- List of All Staff
- List of Registered Students
- List of Courses
- List Building
- Add New Student
- Add New Staff
- Add New Course

**STUDENT**

- Home

**WELCOME**

 **Thomas Okonkwo**  
P001

**PERSONAL INFORMATION**

DATE OF BIRTH :  
2011-11-01

PHONE NUMBER :  
1234568

GENDER:  
M

NATIONALITY :  
Nigerian

ADDRESS :  
123 collins st

**REGISTERED COURSES**

Course Code	Course Name	Credit Hours	Semester
CINS5310	Graduate Research	3	Fall
COMP5303	Programming Languages	3	Fall
COMP5305	Design and Analysis of Algorit	3	Fall

**ADD COURSE**

Figure 7: Screenshot3

Activity	Week 1	Week 2	Week 3	Week 4	Week 5	Responsibility
Design UI for Frontend						Bunmi & Purvi
Design Backend (Database)						Nneoma & Thomas
Link frontend to Database						All Members
Populate Database with dummy data						Nneoma & Thomas
Test, fixes, and review						All Members
Documentation and slides						All Members
Presentation and Demo (Rehearsal)						All Members
Presentation and Demo						All Members

## 9. Milestone for the Project

1. Identify Project- At this phase we are supposed to discover, design and know what kind of system that is required to make this project a success. We have explained the purpose of the database system and analyzed its requirements.
2. Database System Design- At this stage, we design the database schema, create all the relations and constraints for the database. We have come up with our ideas to design the flow of our system and various components involved.
3. Interface Design- At the phase we develop the web interface for the project.
4. Linking Interface with database- At the phase we link the database with the
5. Testing and Debugging- Testing and debugging the errors in the database and user interface functionalities.
6. Documentation and presentation - At this phase, we collected all screenshots, scripts and paper work needed for our presentation. Also, we write a detailed report of all the work that went into creating this project.

## 10. Discussion and conclusion

According to the milestones and timelines shared above, this project was completed and the objectives were met. These are some of the objectives of this project;

- Design and implement close to real-world Database Systems

- Acquire skill sets needed for career in IT areas such as MySQL, php and HTML as well as communication skills

The following assumptions were also made in achieving this project

- Staff, Buildings and Students were already pre-registered in the system
- Only Admin/Course Advisor has the right to delete, update, or insert new records
- New Students will have to reach out to the Course Advisor/Admin for onboarding.

The project affords students the opportunity to see the current status of their course registration and be able to add more courses where required.

Finally, this project was a great insight into how websites work, retrieve and commit data from and into the database, it was eye opening exposing the technology behind majority of the websites in real world.

## **11.Group statement**

### ***11.1 Things learnt from the project***

During the course of this project, we learnt how to apply various complex concepts like setting up a database system, using best practices for creating a database, the 3NF concept to create ER diagram and schema diagram. We also learnt to design the database system architecture and rules that apply for different business cases. We were able to create view and trigger for our database system. We also applied what we learnt in class on how to connect and link the database system with front end web interface using XAMPP server.

### ***11.2 Team Member Roles***

#### **NNEOMA (Ideas for web interface and database tables, presentation assistance)**

- Function as team leader.
- Research on database system requirements and implementation.
- Assist on Project presentation and paper.
- Make sure tasks are completed at scheduled timelines.
- Help team members with duties.

#### **THOMAS (Database and web design)**

- Work on completing database implementation.
- Work on SQL queries and implementation.
- Research solutions for creating schema design, architecture.
- Assist team members on creating tables for the database and connect the database with front end interface.

#### **OLUBUNMI (web interface development and testing)**

- Work on complete web interface implementation.

- Work on SQL queries and implementation.
- Research solutions for creating schema design, architecture.
- Assist team members on creating tables for the database and connect the database with front end interface.

#### **PURVI (Presentation, paper work and database tables)**

- Work on project paper.
- Project presentation.
- Brainstorm ideas on database system requirements, ER diagram, system architecture and implementation.
- Assisting team members in other aspects.

## **12.References**

<https://www.geeksforgeeks.org/how-to-insert-form-data-into-database-using-php/>

<https://www.w3schools.com/php/>



# APPENDIX

## Main Queries

### 1. View Student records with department

```
SELECT s.studentID, s.firstName, s.lastName, s.departmentCode, d.departmentName FROM
student s, department d WHERE s.departmentCode = d.departmentCode;
```

### 2. View all Staff records with department

```
SELECT s.staffNo, s.lecturerFirstName, s.lecturerLastName, s.departmentCode,
d.departmentName FROM staff s, department d WHERE s.departmentCode =
d.departmentCode;
```

### 3. Display All Registered Students record

```
"SELECT c.courseCode, c.courseName, c.creditHours, c.semester FROM coursetable c,
registration r WHERE c.courseCode != r.courseCode AND r.studentID = '$studentID' AND
c.semester = 'Fall'";
```

### 4. Register new Student

```
"INSERT INTO student (studentID, firstName, lastName, dob, phoneNo, gender, address,
createDate, nationality, departmentCode) VALUES
('$studentID','$firstName','$lastName','$dob','$phoneNo','$gender','$address', '$createDate',
'$nationality', '$departmentCode')";
```

### 5. Display courses that have not been registered for student with ID P001

```
SELECT c.courseCode, c.courseName, c.creditHours, c.semester FROM coursetable c,
registration r WHERE c.courseCode != r.courseCode AND r.studentID = "P001" AND
c.semester = "Fall";
```

### 6. Display details of all students that have registered for a course

```
SELECT s.studentID, s.firstName, s.lastName, s.DOB, s.phoneNo, s.address, d.departmentName
as Department FROM registration r, student s, department d WHERE s.studentID= r.studentID
AND s.departmentCode= d.departmentCode;
```

### 7.Display Enrolled Student

```
"SELECT c.courseCode, c.courseName, c.creditHours, c.semester FROM coursetable c,
registration r WHERE c.courseCode != r.courseCode AND r.studentID = '$studentID' AND
c.semester = 'Fall'";
```

#### 8. Update staff Information

```
"UPDATE staff SET staffNo = '$staffNo', lecturerFirstname = '$lecturerFirstname',
lecturerLastname = '$lecturerLastname', phoneNo = '$phoneNo', departmentCode =
'$departmentCode' WHERE staffNo = '$staffNo'";
```

#### 9. Register new Student

```
"INSERT INTO student (studentID, firstName, lastName, dob, phoneNo, gender, address,
createDate, nationality, departmentCode) VALUES
('$studentID','$firstName','$lastName','$dob','$phoneNo','$gender','$address', '$createDate',
'$nationality', '$departmentCode')";
```

#### 10. Get staff schedule

```
SELECT s.staffNo, n.lecturerFirstname, n.lecturerLastname, s.courseCode, s.startTime,
s.duration From schedule s, staff n WHERE s.staffNo=n.staffNo;
```