DHA SUFFA UNIVERSITY

Department of Computer Science

Summer 2023

CS 2003L

DATABASE SYSTEMS LAB PROJECT:

Student Registration System

Github

PROJECT REPORT

Group Members:

Muhammad Asad – CS201195

Azan Ahmed Khan – SE211047

Abdul Moiz – CS192027

Table Of Contents

1. INTRODUCTION:

The Student Registration System project is designed to streamline the process of student registration, course enrollment, grading, and reporting within an educational institution. This report provides a detailed overview of the project's scope, implementation, and key components.

2. PROBLEM STATEMENT:

The manual student registration and record-keeping process at educational institutions often result in inefficiencies, errors, and a lack of data accessibility. The need for a robust and automated system to manage student information, course enrollment, and grading is essential to enhance administrative efficiency and provide students with a smoother registration experience.

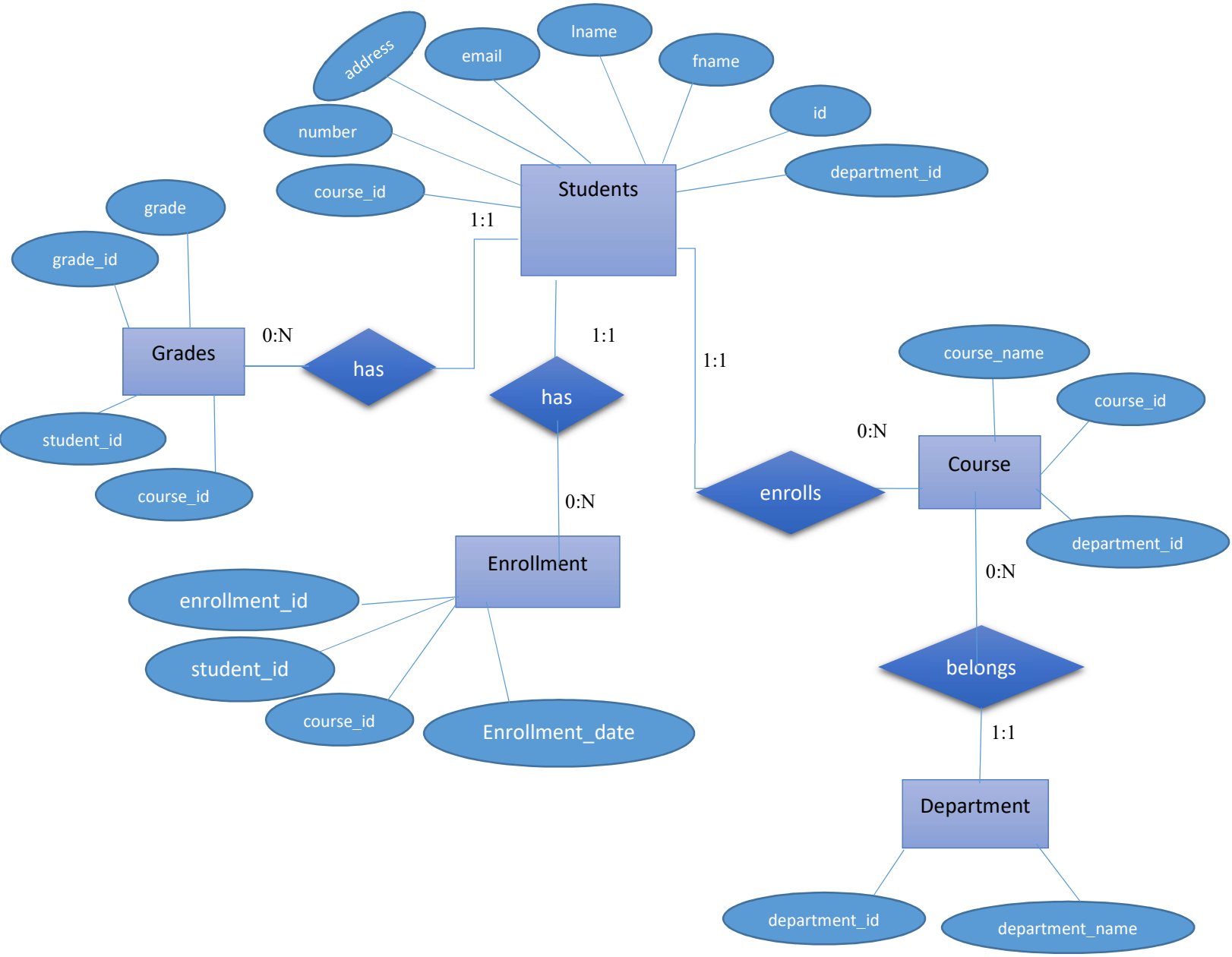
3. RECOMMENDED SOLUTION:

To address the identified problems, we recommend the development of a web-based Student Registration System. This solution will use modern web technologies and a relational database management system to store and manage student information, course details, and enrollment records.

4. BUSINESS RULES:

- Users must register and log in to access the system.
- Students can view and update their profiles.
- Administrators can add, edit, or delete student records.
- Course registration is available to registered students.
- Grading and reporting functionalities are accessible to authorized users.

5. ENTITY-RELATION DIAGRAM AND RELATIONSHIP MODEL:



6. CODE FOR TABLE CREATION AND INSERTION OF RECORDS:

```
-- Create a new database
CREATE DATABASE IF NOT EXISTS registration;

-- Use the newly created database
USE registration;

-- Create a table to store student information
CREATE TABLE IF NOT EXISTS students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    address VARCHAR(255),
    number VARCHAR(15) NOT NULL,
    course_id INT,
    department_id VARCHAR(2),
    FOREIGN KEY (course_id) REFERENCES course(course_id),
    FOREIGN KEY (department_id) REFERENCES department(department_id)
);

CREATE TABLE IF NOT EXISTS course (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_name VARCHAR(100) NOT NULL,
    department_id VARCHAR(2),
    FOREIGN KEY (department_id) REFERENCES department(department_id)
);

CREATE TABLE IF NOT EXISTS department (
    department_id VARCHAR(2) PRIMARY KEY,
    department_name VARCHAR(50) NOT NULL
);

CREATE TABLE IF NOT EXISTS grades (
    grade_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    course_id INT,
    grade FLOAT,
    FOREIGN KEY (student_id) REFERENCES students(id),
    FOREIGN KEY (course_id) REFERENCES course(course_id)
);

CREATE TABLE IF NOT EXISTS enrollment (
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    student_id INT,
    course_id INT,
    enrollment_date DATE,
    FOREIGN KEY (student_id) REFERENCES students(id),
    FOREIGN KEY (course_id) REFERENCES course(course_id)
);


INSERT INTO department (department_id, department_name)
VALUES
    ('CS', 'Computer Science'),
    ('ME', 'Mechanical Engineering'),
    ('EE', 'Electrical Engineering'),
    ('CE', 'Civil Engineering'),
    ('BA', 'Business Administration');

INSERT INTO course (course_name, department_id)
VALUES
    ('Introduction to Programming', 'CS'),
    ('Database Management', 'CS'),
    ('Mechanics of Materials', 'ME'),
    ('Electromagnetic Fields', 'EE'),
    ('Marketing Management', 'BA');

INSERT INTO students (fname, lname, email, address, number, course_id,
department_id)
VALUES
    ('John', 'Doe', 'johndoe@example.com', '123 Main St', '123-456-7890', 1,
'CS'),
    ('Jane', 'Smith', 'janesmith@example.com', '456 Elm St', '987-654-3210', 2,
'CS'),
    ('Alice', 'Johnson', 'alicejohnson@example.com', '789 Oak St', '555-123-
4567', 3, 'ME'),
    ('Bob', 'Williams', 'bobwilliams@example.com', '321 Pine St', '777-888-9999',
4, 'EE'),
    ('Eva', 'Brown', 'evabrown@example.com', '555 Cedar St', '111-222-3333', 5,
'BA');

INSERT INTO grades (student_id, course_id, grade)
VALUES
    (1, 1, 95),
    (2, 1, 88),
    (3, 3, 92),
    (4, 4, 89),
    (5, 5, 78);
```

```
INSERT INTO enrollment (student_id, course_id, enrollment_date)
VALUES
    (1, 1, '2023-01-15'),
    (2, 1, '2023-01-20'),
    (3, 3, '2023-02-10'),
    (4, 4, '2023-02-15'),
    (5, 5, '2023-03-01');
```

```
INSERT INTO students (fname, lname, email, address, number, course_id, department_id)
VALUES
    ('Hans', 'Schmidt', 'hans.schmidt@example.com', '123 Hauptstrasse', '123-456-7890', 1, 'CS'),
    ('Marie', 'Dubois', 'marie.dubois@example.com', '456 Rue Principale', '987-654-3210', 2, 'CE'),
    ('Franz', 'Müller', 'franz.muller@example.com', '789 Hauptweg', '555-123-4567', 3, 'BA'),
    ('Sophie', 'Lefebvre', 'sophie.lefebvre@example.com', '123 Rue de la Poste', '111-222-3333', 4, 'EE'),
    ('Lukas', 'Martin', 'lukas.martin@example.com', '456 Avenue des Champs-Élysées', '222-333-4444', 5, 'ME'),
    ('Amélie', 'Schneider', 'amelie.schneider@example.com', '890 Quai de la Seine', '777-888-9999', 1, 'CS'),
    ('Paul', 'Dupont', 'paul.dupont@example.com', '123 Quai des Orfèvres', '444-555-6666', 2, 'CE'),
    ('Julie', 'Lemoine', 'julie.lemoine@example.com', '456 Avenue de la République', '999-888-7777', 3, 'BA'),
    ('Müller', 'Schmidt', 'muller.schmidt@example.com', '789 Hauptstrasse', '123-321-1111', 4, 'EE'),
    ('Leclerc', 'Dubois', 'leclerc.dubois@example.com', '123 Rue de la Liberté', '111-555-9999', 5, 'ME'),
    ('Eva', 'Berger', 'eva.berger@example.com', '890 Strasse der Einheit', '777-999-2222', 1, 'CS'),
    ('Élise', 'Lefevre', 'elise.lefevre@example.com', '123 Avenue de Gaulle', '222-777-3333', 2, 'CE'),
    ('Thomas', 'Müller', 'thomas.muller@example.com', '456 Friedrichstrasse', '999-111-4444', 3, 'BA'),
    ('Charlotte', 'Martin', 'charlotte.martin@example.com', '123 Boulevard Haussmann', '888-333-2222', 4, 'EE'),
    ('Léa', 'Schneider', 'lea.schneider@example.com', '456 Place de la Concorde', '777-666-5555', 5, 'ME'),
    ('Leon', 'Dubois', 'leon.dubois@example.com', '123 Quai Branly', '666-111-2222', 1, 'CS'),
    ('Manon', 'Dupont', 'manon.dupont@example.com', '456 Avenue de la Bastille', '111-777-8888', 2, 'CE'),
    ('Lucas', 'Lemoine', 'lucas.lemoine@example.com', '890 Rue de la Paix', '444-999-6666', 3, 'BA'),
    ('Müller', 'Schmidt', 'muller.schmidt@example.com', '123 Hauptstrasse', '888-555-2222', 4, 'EE'),
    ('Leclerc', 'Dubois', 'leclerc.dubois@example.com', '456 Rue de la Poste', '666-333-9999', 5, 'ME');
```

```sql
INSERT INTO course (course_name, department_id)
VALUES
    ('Einführung in die Programmierung', 'CS'),
    ('Datenbankmanagement', 'CS'),
    ('Mechanik der Werkstoffe', 'ME'),
    ('Elektromagnetische Felder', 'EE'),
    ('Marketingmanagement', 'BA'),
    ('Datenstrukturen', 'CS'),
    ('Thermodynamik', 'ME'),
    ('Digitale Elektronik', 'EE'),
    ('Betriebswirtschaftslehre', 'BA'),
    ('Mathematik', 'MA'),
    ('Künstliche Intelligenz', 'CS'),
    ('Baustatik', 'CE'),
    ('Finanzmanagement', 'BA'),
    ('Maschinelles Lernen', 'CS'),
    ('Elektrische Schaltungen', 'EE'),
    ('Informatik', 'CS'),
    ('Architektur', 'CE'),
    ('Strömungsmechanik', 'ME'),
    ('Kommunikation', 'BA'),
    ('Regelungstechnik', 'EE');
```

```sql
INSERT INTO department (department_id, department_name)
VALUES
    ('IL', 'Informatik'),
    ('CL', 'Bauingenieurwesen'),
    ('DA', 'Betriebswirtschaftslehre'),
    ('MB', 'Maschinenbau'),
    ('EA', 'Elektrotechnik'),
    ('MA', 'Mathematik');
```

```sql
INSERT INTO grades (student_id, course_id, grade)
VALUES
    (1, 1, 85.5),
    (2, 1, 78.0),
    (3, 2, 92.5),
    (4, 2, 88.0),
    (5, 3, 76.5),
    (6, 3, 90.0),
    (7, 4, 87.0),
    (8, 4, 82.5),
    (9, 5, 89.5),
    (10, 5, 95.0),
    (11, 1, 77.5),
    (12, 1, 84.0),
    (13, 2, 93.5),
    (14, 2, 86.0),
    (15, 3, 75.5),
    (16, 3, 91.0),
    (17, 4, 88.5),
    (18, 4, 83.0),
    (19, 5, 90.5),
    (20, 5, 94.0);
```

```sql

INSERT INTO enrollment (student_id, course_id, enrollment_date) VALUES
(11, 1, '2023-09-01'),
(12, 2, '2023-09-02'),
(13, 3, '2023-09-03'),
(14, 4, '2023-09-04'),
(15, 1, '2023-09-05'),
(16, 2, '2023-09-06'),
(17, 3, '2023-09-07'),
(18, 4, '2023-09-08'),
(19, 1, '2023-09-09'),
(20, 2, '2023-09-10'),
(21, 3, '2023-09-11'),
(22, 4, '2023-09-12'),
(23, 1, '2023-09-13'),
(24, 2, '2023-09-14'),
(25, 3, '2023-09-15'),
(26, 4, '2023-09-16'),
(27, 1, '2023-09-17'),
(28, 2, '2023-09-18'),
(29, 3, '2023-09-19'),
(30, 4, '2023-09-20');
```

7. NORMALIZED TABLES:

1. Students Table:

- Student ID (Primary Key)
- First Name
- Last Name
- Email
- Address
- Phone Number
- Course ID (Foreign Key referencing the Courses Table)
- Department ID (Foreign Key referencing the Departments Table)

2. Courses Table:

- Course ID (Primary Key)
- Course Name
- Department ID (Foreign Key referencing the Departments Table)

3. Departments Table:

- Department ID (Primary Key)
- Department Name

4. Grades Table (This table can store grades for each student in each course):

- Grade ID (Primary Key)
- Student ID (Foreign Key referencing the Students Table)
- Course ID (Foreign Key referencing the Courses Table)
- Grade (e.g., A, B, C, etc.)

5. Enrollments Table (This table can track when a student enrolls in a course):

- Enrollment ID (Primary Key)
- Student ID (Foreign Key referencing the Students Table)
- Course ID (Foreign Key referencing the Courses Table)
- Enrollment Date

8. DATA DICTIONARY:

Students Table

Field Name	Data Type	Length	Description
id (PK)	INT		Primary key for student records
fname	VARCHAR(50)	50	First name of the student
lname	VARCHAR(50)	50	Last name of the student
email	VARCHAR(100)	100	Email address of the student
address	VARCHAR(255)	255	Student's address
number	VARCHAR(15)	15	Phone number of the student
course_id (FK)	INT		Foreign key referencing Courses Table
department_id (FK)	VARCHAR(2)	2	Foreign key referencing Departments Table

Course Table

Field Name	Data Type	Length	Description
course_id (PK)	INT		Primary key for course records

course_name	VARCHAR(100)	100	Name of the course
department_id (FK)	VARCHAR(2)	2	Foreign key referencing Departments Table

Department Table

Field Name	Data Type	Length	Description
department_id (PK)	VARCHAR(2)	2	Primary key for department records
department_name	VARCHAR(50)	50	Name of the department

Grades Table

Field Name	Data Type	Length	Description
grade_id (PK)	INT		Primary key for grade records
student_id (FK)	INT		Foreign key referencing Students Table
course_id (FK)	INT		Foreign key referencing Courses Table
grade	FLOAT		Grade received by the student

Enrollment Table

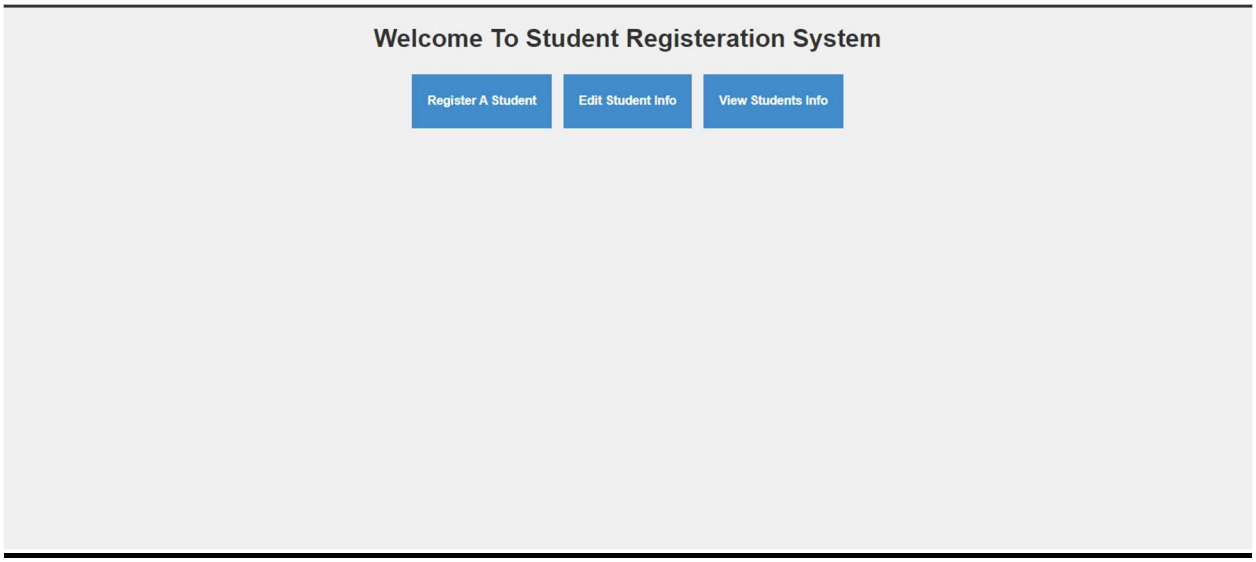
Field Name	Data Type	Length	Description

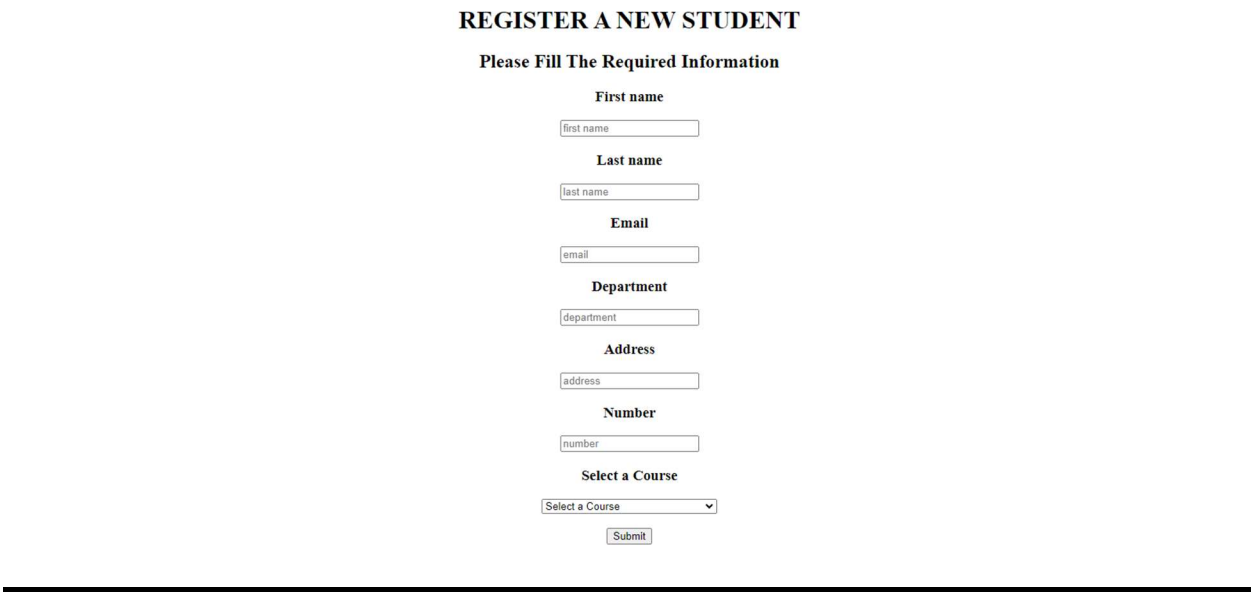
enrollment_id (PK)	INT		Primary key for enrollment records
student_id (FK)	INT		Foreign key referencing Students Table
course_id (FK)	INT		Foreign key referencing Courses Table
enrollment_date	DATE		Date when the student enrolled

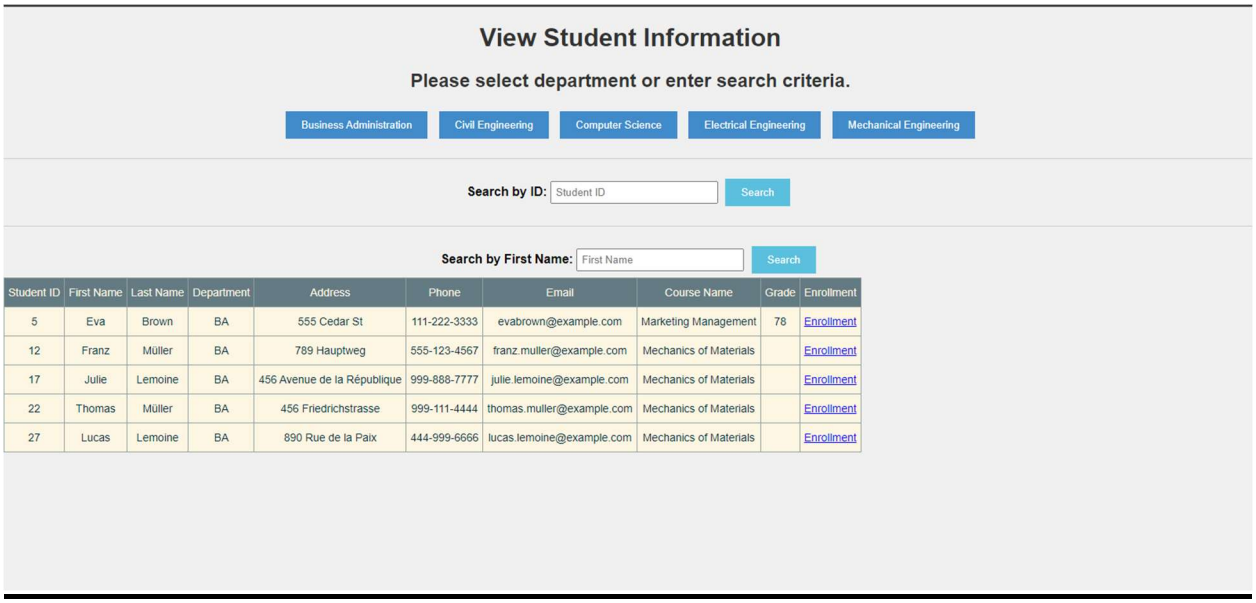
9. REPORTING QUERIES WITH CODE AND SCREENSHOTS:

```
-- queries used in PHP scripts
INSERT INTO enrollment (student_id, course_id, enrollment_date) VALUES
(LAST_INSERT_ID(), ?, ?);

insert into students(fname,lname,email,department,address,number,
course_id)values(?, ?, ?, ?, ?, ?, ?);

// Query to fetch student information including department and course details
    $query = "SELECT students.id, students.fname, students.lname,
students.department, students.address, students.number, students.email,
course.course_name, grades.grade
              FROM students
              LEFT JOIN course ON students.course_id = course.course_id
              LEFT JOIN grades ON students.id = grades.student_id
              WHERE students.id='$inp_search_id'";

// Query used to edit records
$query = "SELECT students.id, students.fname, students.lname,
students.department, students.address, students.number, students.email,
course.course_name, grades.grade
                  FROM students
                  LEFT JOIN course ON students.course_id = course.course_id
                  LEFT JOIN grades ON students.id = grades.student_id
                  WHERE students.id='$input_id_var' ";
//  Update Queries
$update_query = "UPDATE students SET department_id = '$new_department_id' WHERE
id = $student_id";
```

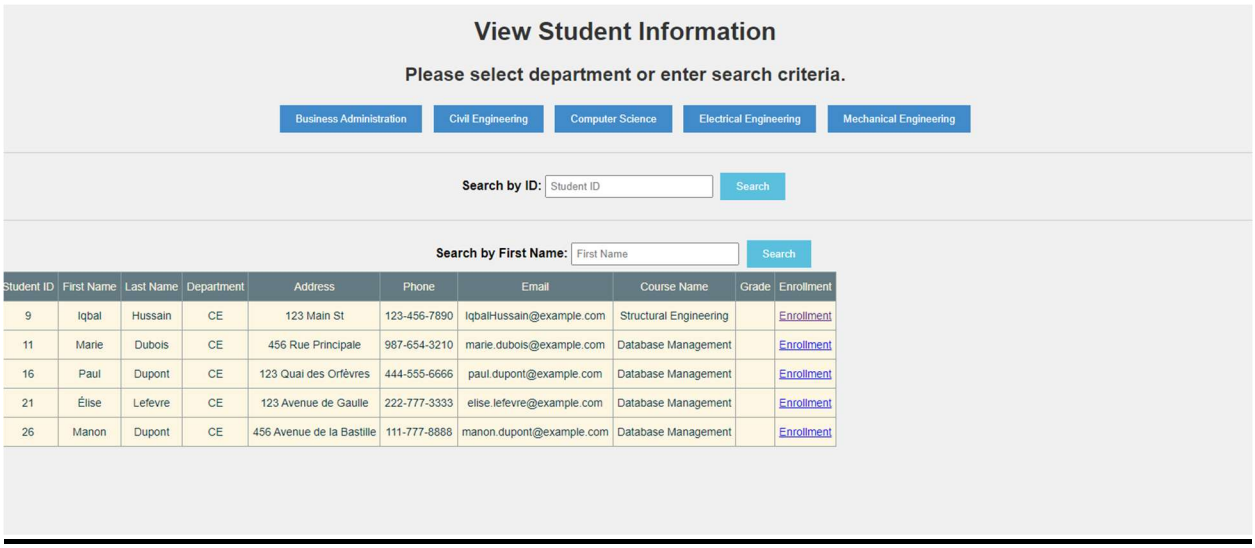
```
$sql = "UPDATE students SET fname = '$fname', lname = '$lname', email = '$email',
department_id = '$department_id', address = '$address', number = '$number' WHERE
id = $id";
$query = "SELECT * FROM students WHERE id='$input_id_var' ";

// used in inserting records
("INSERT INTO students (fname, lname, email, department_id, address, number,
course_id) VALUES (?, ?, ?, ?, ?, ?, ?)");
("INSERT INTO enrollment (student_id, course_id, enrollment_date) VALUES
(LAST_INSERT_ID(), ?, ?)");

 // Query to fetch enrollment information for the selected student
    $query = "SELECT enrollment.enrollment_id, enrollment.course_id,
course.course_name, enrollment.enrollment_date
              FROM enrollment
              LEFT JOIN course ON enrollment.course_id = course.course_id
              WHERE enrollment.student_id='$student_id'";

// Function to fetch and display students by department
function fetchStudentsByDepartment($con, $department_id)
    $query = "SELECT students.id, students.fname, students.lname,
students.department_id, students.address, students.number, students.email,
course.course_name, grades.grade
              FROM students
              LEFT JOIN course ON students.course_id = course.course_id
              LEFT JOIN grades ON students.id = grades.student_id
              WHERE students.department_id='$department_id'";

// Function to fetch and display students by ID or first name
function searchStudents($con, $search_id, $search_fname)
    $query = "SELECT students.id, students.fname, students.lname,
students.department_id, students.number, students.email, course.course_name,
grades.grade
              FROM students
              LEFT JOIN course ON students.course_id = course.course_id
              LEFT JOIN grades ON students.id = grades.student_id
              WHERE students.id='$search_id' OR students.fname LIKE
'%$search_fname%'";
```


Welcome To Student Registeration System
Register A Student
Edit Student Info
View Students Info


REGISTER A NEW STUDENT
Please Fill The Required Information
First name
first name
Last name
last name
Email
email
Department
department
Address
address
Number
number
Select a Course
Select a Course
Submit



Student ID	First Name	Last Name	Department	Address	Phone	Email	Course Name	Grade	Enrollment
5	Eva	Brown	BA	555 Cedar St	111-222-3333	evabrown@example.com	Marketing Management	78	Enrollment
12	Franz	Müller	BA	789 Hauptweg	555-123-4567	franz.muller@example.com	Mechanics of Materials		Enrollment
17	Julie	Lemoine	BA	456 Avenue de la République	999-888-7777	julie.lemoine@example.com	Mechanics of Materials		Enrollment
22	Thomas	Müller	BA	456 Friedrichstrasse	999-111-4444	thomas.muller@example.com	Mechanics of Materials		Enrollment
27	Lucas	Lemoine	BA	890 Rue de la Paix	444-999-6666	lucas.lemoine@example.com	Mechanics of Materials		Enrollment



Student ID	First Name	Last Name	Department	Address	Phone	Email	Course Name	Grade	Enrollment
9	Iqbal	Hussain	CE	123 Main St	123-456-7890	IqbalHussain@example.com	Structural Engineering		Enrollment
11	Marie	Dubois	CE	456 Rue Principale	987-654-3210	marie.dubois@example.com	Database Management		Enrollment
16	Paul	Dupont	CE	123 Quai des Orfèvres	444-555-6666	paul.dupont@example.com	Database Management		Enrollment
21	Élise	Lefevre	CE	123 Avenue de Gaulle	222-777-3333	elise.lefevre@example.com	Database Management		Enrollment
26	Manon	Dupont	CE	456 Avenue de la Bastille	111-777-8888	manon.dupont@example.com	Database Management		Enrollment

Search by ID: Student ID Search

Search by First Name: First Name Search

Student ID	First Name	Last Name	Department	Address	Phone	Email	Course Name	Grade	Enrollment
1	John	Doe	CS		123-456-7890	johndoe@example.com	Introduction to Programming	95	Enrollment
2	John	Smith	CS		987-654-3210	janesmith@example.com	Database Management	88	Enrollment
3	Alice	Johnson	ME		555-123-4567	alicejohnson@example.com	Mechanics of Materials	92	Enrollment
4	Bob	Williams	EE		777-888-9999	bobwilliams@example.com	Electromagnetic Fields	89	Enrollment
5	Eva	Brown	BA		111-222-3333	evabrown@example.com	Marketing Management	78	Enrollment
8	Abdul	Rafay			03114895725	rafay13@protonmail.com	Marketing Management		Enrollment
9	Iqbal	Hussain	CE		123-456-7890	IqbalHussain@example.com	Structural Engineering		Enrollment
10	Hans	Schmidt	CS		123-456-7890	hans.schmidt@example.com	Introduction to Programming		Enrollment
11	Marie	Dubois	CE		987-654-3210	marie.dubois@example.com	Database Management		Enrollment
12	Franz	Müller	BA		555-123-4567	franz.muller@example.com	Mechanics of Materials		Enrollment
13	Sophie	Lefebvre	EE		111-222-3333	sophie.lefebvre@example.com	Electromagnetic Fields		Enrollment
14	Lukas	Martin	ME		222-333-4444	lukas.martin@example.com	Marketing Management		Enrollment
15	Amélie	Schneider	CS		777-888-9999	amelie.schneider@example.com	Introduction to Programming		Enrollment
16	Paul	Dupont	CE		444-555-6666	paul.dupont@example.com	Database Management		Enrollment
17	Julie	Lemoine	BA		999-888-7777	julie.lemoine@example.com	Mechanics of Materials		Enrollment

View Student Information

Please select department or enter search criteria.

Business Administration | Civil Engineering | Computer Science | Electrical Engineering | Mechanical Engineering

Search by ID: Student ID Search

Search by First Name: Syed Search

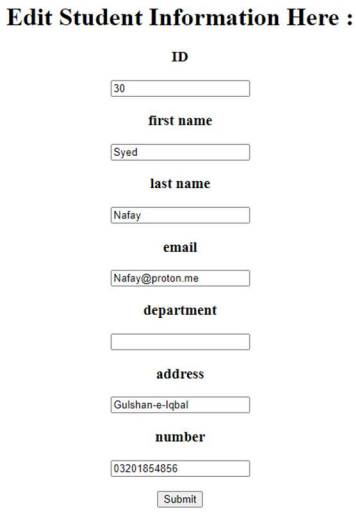
Student ID	First Name	Last Name	Department	Address	Phone	Email	Course Name	Grade	Enrollment
30	Syed	Nafay			03201854856	Nafay@proton.me	Mechanics of Materials		Enrollment

Edit Student Information

Please search for a student by ID

30

Submit



---

10. CONCLUSION:

The Student Registration System project successfully addresses the challenges of manual registration and record-keeping in educational institutions. It offers a user-friendly interface for students and administrators, efficient data management, and robust reporting capabilities.