RAJALAKSHMI
ENGINEERING
COLLEGE

SUBJECT:	OBJECT ORIENTED PROGRAMMING
CODE:	CS23333
NAME:	ANISHA G
ROLL NO:	2116240701041
YEAR:	2ND YEAR
SEC:	CSE-FA

ONLINE COURSE ENROLLMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

ANISHA G 240701041

LIPIKA J 240701288

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“ONLINE COURSE ENROLLMENT SYSTEM”** is the bonafide work of “ **ANISHA G, LIPIKA J** ” who carried out the project work under my supervision.

SIGNATURE

MR.KARTHIKEYAN

ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on _________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ABSTRACT

The Online Course Enrollment System is a comprehensive database management solution designed to streamline and automate the process of student course registration in educational institutions. The system addresses critical challenges faced by existing enrollment platforms, including limited seat availability tracking, scheduling conflicts, lack of real-time updates, and administrative overhead.

This project implements a robust database architecture with real-time seat tracking, automated waitlist management, intelligent timetable conflict detection, and course recommendations. By leveraging relational database concepts including normalization (up to 3NF/BCNF), stored procedures, triggers, and user role management, the system ensures data integrity, security, and efficient transaction processing. The implementation utilizes SQL for database operations (DDL, DML, DCL) and provides a user-friendly interface for students , significantly improving the enrollment experience while reducing manual administrative workload.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **MR.KARTHIKEYAN ,** for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. ANISHA G

2. LIPIKA J

TABLE OF CONTENTS

CHAPTER NO **TITLE** **PAGE NO**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Online Course Enrollment System is designed to revolutionize the way educational institutions manage student registrations for various courses through a digital platform. In today's fast-paced educational environment, efficient course enrollment is crucial for both student satisfaction and institutional effectiveness. This system provides a comprehensive solution that handles all aspects of the enrollment process, from initial course discovery to final registration confirmation.

1.2 SCOPE OF THE WORK

Database Design and Modeling:Complete ER diagram with entities including Students, Courses, Instructors, Enrollments, and Departments **Relational Schema Implementation**:Normalized database structure (3NF/BCNF) ensuring data integrity. **Core Database Operations**:Complex SQL queries for enrollment management, grade tracking, and capacity monitoring. **Security and User Management**:Role-based access control using DCL. **Automation**:Triggers and stored procedures for real-time updates and constraint enforcement. **Integration**:Backend database interfacing with front-end web application. **Advanced Features**:AI-based recommendations, waitlist management, and conflict detection.

1.3 PROBLEM STATEMENT

An Online Course Enrollment System aims to simplify the process of registering students for various courses through a digital platform.Existing online course enrollment systems face multiple critical challenges that hinder efficient course registration and management:

1. **Limited Seats & Unavailable Courses**: Students often encounter fully enrolled courses without visibility into future availability
2. **No Real-Time Seat Availability**: Delayed updates lead to overbooking or enrollment failures
3. **Clashing Course Timings**: Students inadvertently register for overlapping classes, discovering conflicts too late
4. **No Recommendation System**: Students lack guidance on relevant courses based on their academic history and interests
5. **Admin Overload**: Manual monitoring of enrollments, approvals, and capacity management creates bottlenecks and errors

1.4 AIM AND OBJECTIVES OF THE PROJECT

Aim: Design a database-driven enrollment system with automation and intelligent features.

Objectives: Design normalized database schema with all entities,Implement real-time seat tracking,Develop automated conflict detection,Create course recommendations,Implement automated waitlist functionality,Establish user roles and permissions,Create triggers and stored procedures ,Develop comprehensive SQL queries,Integrate with front-end application

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front – End	:	JavaScript
Back - End	:	MySql
Language	:	JavaScript,SQL

CHAPTER 3

MODULE DESCRIPTION

This application consists of two modules designed for student enrollment functionality. When the program runs, users are directed to a login interface where they can authenticate as a Student. The description of the modules are as follows:

3.1 STUDENT LOGIN MODULE

When a user logs in as a Student, they must authenticate using their student credentials (username and password). Upon successful login, students are directed to the enrollment dashboard.

Login Features:

- **User Authentication** - Secure login with student ID and password
- **Session Management** - Maintains user session throughout the enrollment process
- **Access Control** - Students can only access their own enrollment data

3.2 STUDENT ENROLLMENT MODULE

After successful login, students can access the enrollment system with the following capabilities:

Course Browsing & Search:

- View available courses with detailed information (course name, credits, instructor, schedule)
- Real-time seat availability display
- Search and filter courses by department, instructor, or timings

Course Enrollment:

- Register for courses with automatic validation
- Real-time seat availability checking before enrollment
- Timetable conflict detection to prevent overlapping schedules
- Prerequisite verification for course eligibility
- Instant enrollment confirmation upon successful registration

Enrollment Management:

- View current enrolled courses with schedule details
- Display personalized timetable showing all enrolled courses
- Drop/withdraw from courses within the allowed period
- View enrollment history and status

Waitlist Features:

- Join waitlist for courses that are full
- Track waitlist position in real-time
- Automatic enrollment notification when seat becomes available

Course Recommendations:

- Personalized course suggestions
- Recommendations based on previous enrollments and academic history
- Display highly-rated courses relevant to student's major

Enrollment Summary:

- View total enrolled credits
- Check enrollment status (active, waitlisted, completed)
- Display grades for completed courses

CHAPTER 4

SAMPLE CODING

DB CONNECTION:

```
 package com.enrollment.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

    // MySQL connection details

    private static final String JDBC_DRIVER =
"com.mysql.cj.jdbc.Driver";

    private static final String DB_URL =
"jdbc:mysql://localhost:3306/course_enrollment_db";

    private static final String USERNAME = "root";

    private static final String PASSWORD = "dbms2006";

    // Connection parameters
    private static final String PARAMS = "useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true;

    private static final String FULL_URL = DB_URL + PARAMS

    static {

        try {

            // Explicitly load MySQL JDBC Driver

            Class.forName(JDBC_DRIVER);

            System.out.println("✓ MySQL JDBC Driver loaded successfully!");

        } catch (ClassNotFoundException e) {

            System.err.println("✗ MySQL JDBC Driver not found!");

            System.err.println("Make sure mysql-connector-j-8.3.0.jar is in WEB-INF/lib/");

            e.printStackTrace();

            throw new ExceptionInInitializerError(e);

        }

    }
    public static Connection getConnection() throws SQLException {

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(FULL_URL,
USERNAME, PASSWORD);

            System.out.println("✓ Database connection successful!");

            System.out.println("Connected to: " + DB_URL);

            return conn;

        } catch (SQLException e) {

            System.err.println("✗ Database connection failed!");

            System.err.println("URL: " + FULL_URL);

            System.err.println("Username: " + USERNAME);

            System.err.println("Error Code: " + e.getErrorCode());

            System.err.println("SQL State: " + e.getSQLState());

            System.err.println("Message: " + e.getMessage());

            throw e;

        }
    }  public static void closeConnection(Connection conn) {
        if (conn != null) {
            try {
                conn.close();
                System.out.println("Database connection closed.");
            } catch (SQLException e) {
                System.err.println("Error closing connection: " +
e.getMessage());
                e.printStackTrace();
            }
        }
    }

    // Test database connection
    public static void main(String[] args) {
        System.out.println("Testing database connection...");
        System.out.println("===================================");
        try {

         Connection conn = getConnection();

            if (conn != null && !conn.isClosed()) {

                System.out.println("✓ Connection test SUCCESSFUL!");

                System.out.println("Database: " + conn.getCatalog());

                closeConnection(conn);

            }

        } catch (SQLException e) {

            System.err.println(" ✗ Connection test FAILED!");

            System.err.println("\nTroubleshooting steps:");

            System.err.println("1. Make sure MySQL server is running");

            System.err.println("2. Verify database 'course_enrollment_db'
exists");

            System.err.println("3. Check username and password are
correct");

            System.err.println("4. Ensure mysql-connector-j-8.3.0.jar is in
WEB-INF/lib/");
```

```
            e.printStackTrace();
        }}}
```

BACKEND MYSQL CODE:

```
USE course_enrollment_db;

-- Drop existing tables

SET FOREIGN_KEY_CHECKS = 0;

DROP TABLE IF EXISTS waitlist;

DROP TABLE IF EXISTS enrollments;

DROP TABLE IF EXISTS courses;

DROP TABLE IF EXISTS students;

DROP TABLE IF EXISTS users;

SET FOREIGN_KEY_CHECKS = 1;



-- Create Users table

CREATE TABLE users (

    user_id INT AUTO_INCREMENT PRIMARY KEY,

    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,

    email VARCHAR(100) UNIQUE NOT NULL,

    role VARCHAR(20) NOT NULL CHECK (role IN ('STUDENT', 'ADMIN')),

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;



-- Create Students table

CREATE TABLE students (

    student_id INT AUTO_INCREMENT PRIMARY KEY,

    user_id INT UNIQUE NOT NULL,

    first_name VARCHAR(50) NOT NULL,

    last_name VARCHAR(50) NOT NULL,

    date_of_birth DATE,

    phone VARCHAR(15),

    address VARCHAR(200),

    CONSTRAINT fk_student_user FOREIGN KEY (user_id)
```

```
        REFERENCES users(user_id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Courses table
CREATE TABLE courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_code VARCHAR(20) UNIQUE NOT NULL,
    course_name VARCHAR(100) NOT NULL,
    description VARCHAR(500),
    credits INT NOT NULL,
    max_capacity INT NOT NULL,
    enrolled_count INT DEFAULT 0,
    instructor_name VARCHAR(100),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Enrollments table
```

```
CREATE TABLE enrollments (
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT NOT NULL,
    course_id INT NOT NULL,
    enrollment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) DEFAULT 'ACTIVE',
    grade VARCHAR(2),
    CONSTRAINT fk_enrollment_student FOREIGN KEY (student_id)
        REFERENCES students(student_id) ON DELETE CASCADE,
    CONSTRAINT fk_enrollment_course FOREIGN KEY (course_id)
        REFERENCES courses(course_id) ON DELETE CASCADE,
    CONSTRAINT unique_enrollment UNIQUE (student_id, course_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;


-- Create Waitlist table
CREATE TABLE waitlist (
    waitlist_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT NOT NULL,

    course_id INT NOT NULL,

    added_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    position INT NOT NULL,

    status VARCHAR(20) DEFAULT 'WAITING',

    CONSTRAINT fk_waitlist_student FOREIGN KEY (student_id)

        REFERENCES students(student_id) ON DELETE CASCADE,

    CONSTRAINT fk_waitlist_course FOREIGN KEY (course_id)

        REFERENCES courses(course_id) ON DELETE CASCADE,

    CONSTRAINT unique_waitlist UNIQUE (student_id, course_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;


-- Create Triggers

DELIMITER //


CREATE TRIGGER trg_enrollment_insert

AFTER INSERT ON enrollments
```

```
FOR EACH ROW
BEGIN
    IF NEW.status IN ('ACTIVE', 'ENROLLED') THEN
        UPDATE courses SET enrolled_count = enrolled_count + 1
        WHERE course_id = NEW.course_id;
    END IF;
END//

CREATE TRIGGER trg_enrollment_delete
AFTER DELETE ON enrollments
FOR EACH ROW
BEGIN
    IF OLD.status IN ('ACTIVE', 'ENROLLED') THEN
        UPDATE courses SET enrolled_count =
GREATEST(enrolled_count - 1, 0)
        WHERE course_id = OLD.course_id;
    END IF;
```

```
END//


CREATE TRIGGER trg_enrollment_update
AFTER UPDATE ON enrollments
FOR EACH ROW
BEGIN
   IF OLD.status IN ('ACTIVE', 'ENROLLED') AND NEW.status NOT IN ('ACTIVE', 'ENROLLED') THEN
       UPDATE courses SET enrolled_count = GREATEST(enrolled_count - 1, 0)
       WHERE course_id = NEW.course_id;
   ELSEIF OLD.status NOT IN ('ACTIVE', 'ENROLLED') AND NEW.status IN ('ACTIVE', 'ENROLLED') THEN
       UPDATE courses SET enrolled_count = enrolled_count + 1
       WHERE course_id = NEW.course_id;
   END IF;
END//
```

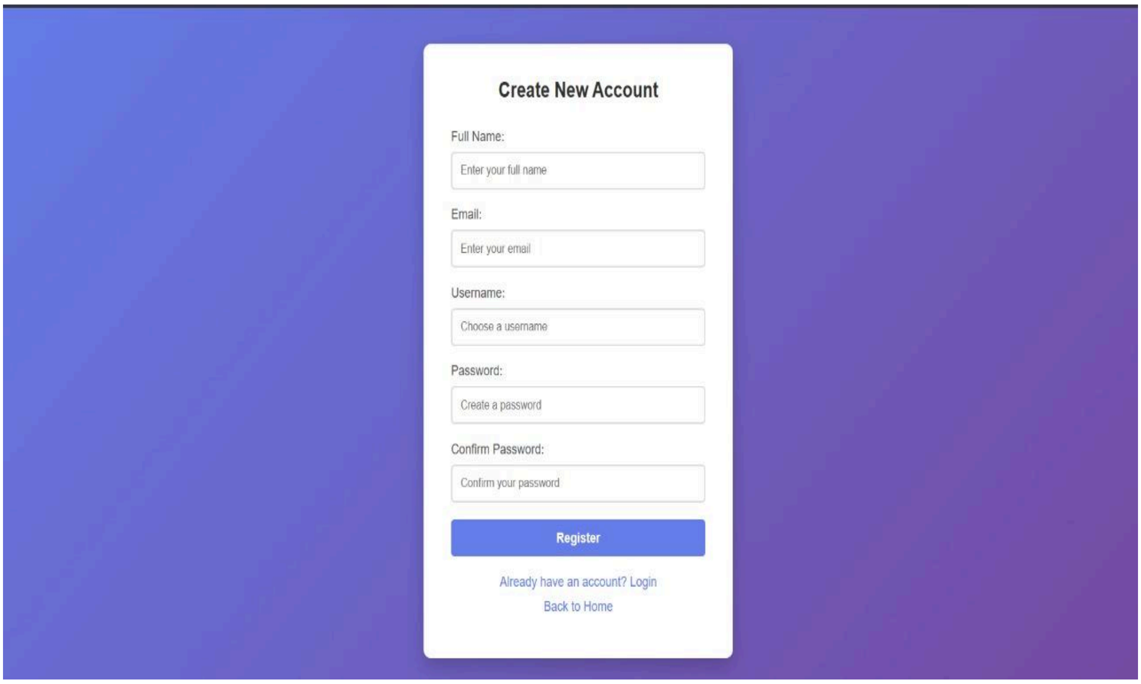
```
CREATE TRIGGER trg_waitlist_position

BEFORE INSERT ON waitlist

FOR EACH ROW

BEGIN

    DECLARE max_pos INT;

    SELECT COALESCE(MAX(position), 0) INTO max_pos

    FROM waitlist WHERE course_id = NEW.course_id AND status =
'WAITING';

    SET NEW.position = max_pos + 1;

END//


DELIMITER ;


-- Insert Users

INSERT INTO users (username, password, email, role) VALUES

('admin', 'admin123', 'admin@university.edu', 'ADMIN'),

('john_doe', 'student123', 'john@student.edu', 'STUDENT'),
('jane_smith', 'student123', 'jane@student.edu', 'STUDENT'),

('bob_wilson', 'student123', 'bob@student.edu', 'STUDENT');


-- Insert Students

INSERT INTO students (user_id, first_name, last_name, date_of_birth, phone, address) VALUES

(2, 'John', 'Doe', '2000-05-15', '1234567890', '123 Main St'),

(3, 'Jane', 'Smith', '1999-08-20', '9876543210', '456 Oak Ave'),

(4, 'Bob', 'Wilson', '2001-03-10', '5551234567', '789 Pine Rd');


-- Insert Courses (FIXED - removed duplicates)

INSERT INTO courses (course_code, course_name, description, credits, max_capacity, instructor_name, enrolled_count) VALUES

('CS101', 'Introduction to Computer Science', 'Basic programming concepts', 3, 30, 'Dr. Smith', 4),

('CS102', 'Programming Fundamentals', 'Learn programming basics with Python', 3, 35, 'Dr. Anderson', 15),

('CS201', 'Data Structures', 'Advanced data structures', 4, 25, 'Dr. Johnson', 8),
('CS202', 'Data Structures and Algorithms', 'Advanced algorithms and complexity analysis', 4, 30, 'Dr. Martinez', 20),

('CS301', 'Database Systems', 'Database design and SQL', 3, 20, 'Dr. Williams', 3),

('CS303', 'Operating Systems', 'Process management and system calls', 4, 25, 'Dr. Chen', 18),

('CS401', 'Machine Learning', 'Introduction to ML algorithms', 3, 20, 'Dr. Taylor', 12),

('CS402', 'Computer Networks', 'Network protocols and architecture', 3, 25, 'Dr. White', 10),

('CS403', 'Software Engineering', 'Software development lifecycle and methodologies', 3, 30, 'Dr. Brown', 16),

('CS404', 'Web Development', 'Full-stack web application development', 4, 28, 'Dr. Garcia', 22),

('CS405', 'Mobile App Development', 'iOS and Android app development', 3, 25, 'Dr. Lee', 19),

('MATH101', 'Calculus I', 'Differential calculus', 4, 40, 'Dr. Brown', 16),

('MATH201', 'Linear Algebra', 'Vectors, matrices and transformations', 4, 35, 'Dr. Davis', 22),

('MATH301', 'Discrete Mathematics', 'Logic, sets, and graph theory', 3, 30, 'Prof. Garcia', 15),
('MATH302', 'Probability and Statistics', 'Statistical analysis and probability theory', 4, 32, 'Dr. Kim', 20),

('ENG101', 'English Composition', 'Writing skills', 3, 35, 'Prof. Davis', 20),

('ENG102', 'Literature and Composition', 'Critical reading and writing', 3, 30, 'Prof. Wilson', 18),

('ENG201', 'Technical Writing', 'Professional communication skills', 3, 30, 'Prof. Moore', 20),

('PHY101', 'Physics I', 'Mechanics', 4, 30, 'Dr. Martinez', 9),

('PHYS101', 'Physics I', 'Mechanics and thermodynamics', 4, 40, 'Dr. Lee', 28),

('PHYS201', 'Physics II', 'Electricity and magnetism', 4, 35, 'Dr. Lee', 24),

('CHEM101', 'General Chemistry I', 'Chemical principles and reactions', 4, 35, 'Dr. Rodriguez', 30),

('CHEM201', 'Organic Chemistry', 'Carbon compounds and reactions', 4, 30, 'Dr. Rodriguez', 22),

('BIO101', 'Biology I', 'Cell structure and function', 4, 40, 'Dr. Kim', 32),

('HIST101', 'World History', 'Ancient to modern civilizations', 3, 35, 'Prof. Thompson', 25),
```

```sql
('PSYCH101', 'Introduction to Psychology', 'Human behavior and mental processes', 3, 40, 'Dr. Martinez', 35),

('ECON101', 'Microeconomics', 'Supply, demand and market systems', 3, 35, 'Prof. Jackson', 28),

('ART101', 'Introduction to Art', 'Art history and appreciation', 3, 30, 'Prof. Davis', 20),

('MUS101', 'Music Theory', 'Fundamentals of music composition', 3, 25, 'Prof. Anderson', 18);

-- Insert Sample Enrollments

INSERT INTO enrollments (student_id, course_id, status) VALUES

(1, 1, 'ENROLLED'),

(2, 1, 'ENROLLED');

-- Show success

SELECT '✓ DATABASE SETUP COMPLETE!' as Status;

SHOW TABLES;

SELECT 'Total Users:' as Info, COUNT(*) as Count FROM users;

SELECT 'Total Students:' as Info, COUNT(*) as Count FROM students;

SELECT 'Total Courses:' as Info, COUNT(*) as Count FROM courses;
```

CHAPTER 5

SCREEN SHOTS

Fig 5.1 Registration page



Create New Account

Full Name:

Enter your full name

Email:

Enter your email

Username:

Choose a username

Password:

Create a password

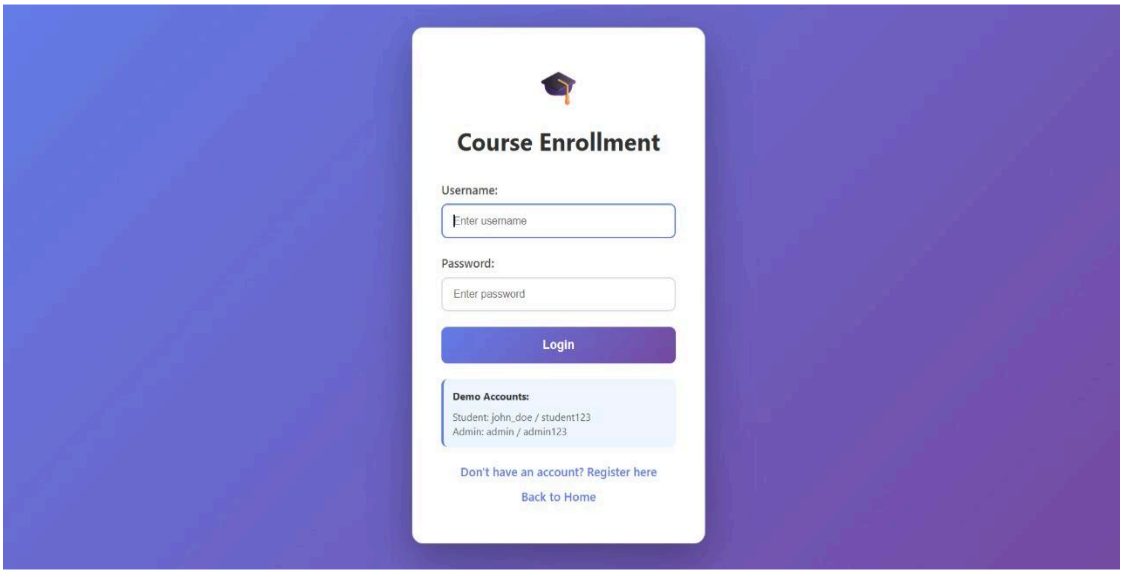
Confirm Password:

Confirm your password

Already have an account? Login

Back to Home

Fig 5.2 Login page



Course Enrollment

Username:

Enter username

Password:

Enter password

Demo Accounts:

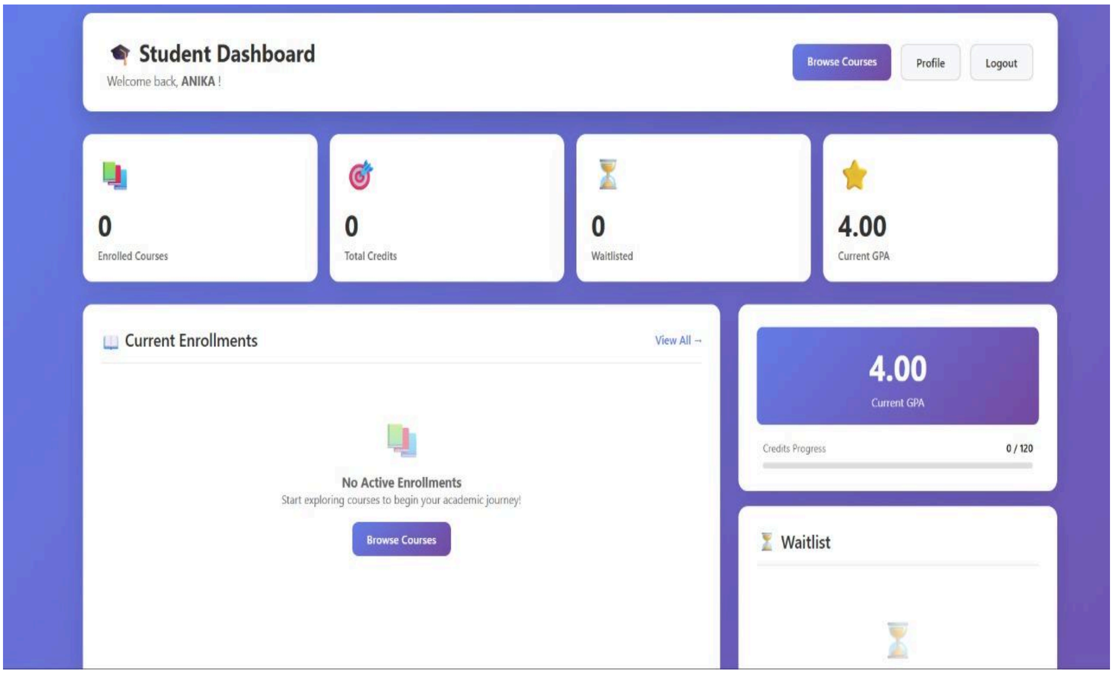
Student: john_doe / student123

Admin: admin / admin123

Don't have an account? Register here

Back to Home

Fig 5.3 Student DashBoard



No Active Enrollments

Start exploring courses to begin your academic journey!

Fig 5.4 Available Course

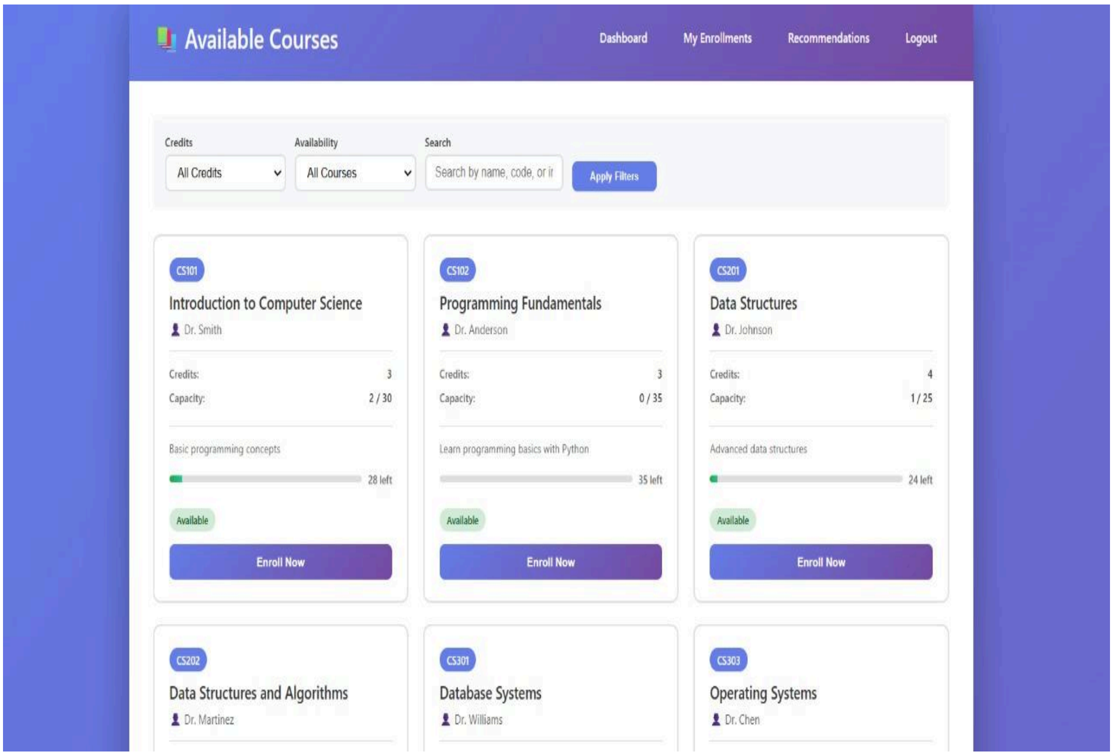
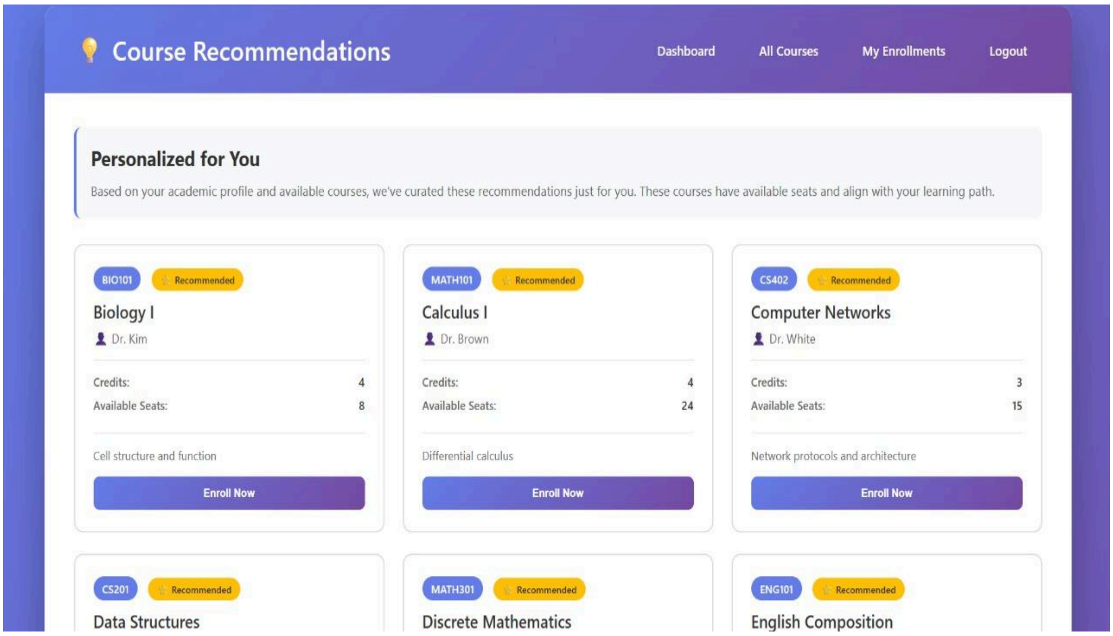


Fig 5.5 Course Recommendation



Course Recommendations

Dashboard | All Courses | My Enrollments | Logout

Personalized for You

Based on your academic profile and available courses, we've curated these recommendations just for you. These courses have available seats and align with your learning path.

BIO101 Recommended
Biology I
Dr. Kim
Credits: 4
Available Seats: 8
Cell structure and function
Enroll Now

MATH101 Recommended
Calculus I
Dr. Brown
Credits: 4
Available Seats: 24
Differential calculus
Enroll Now

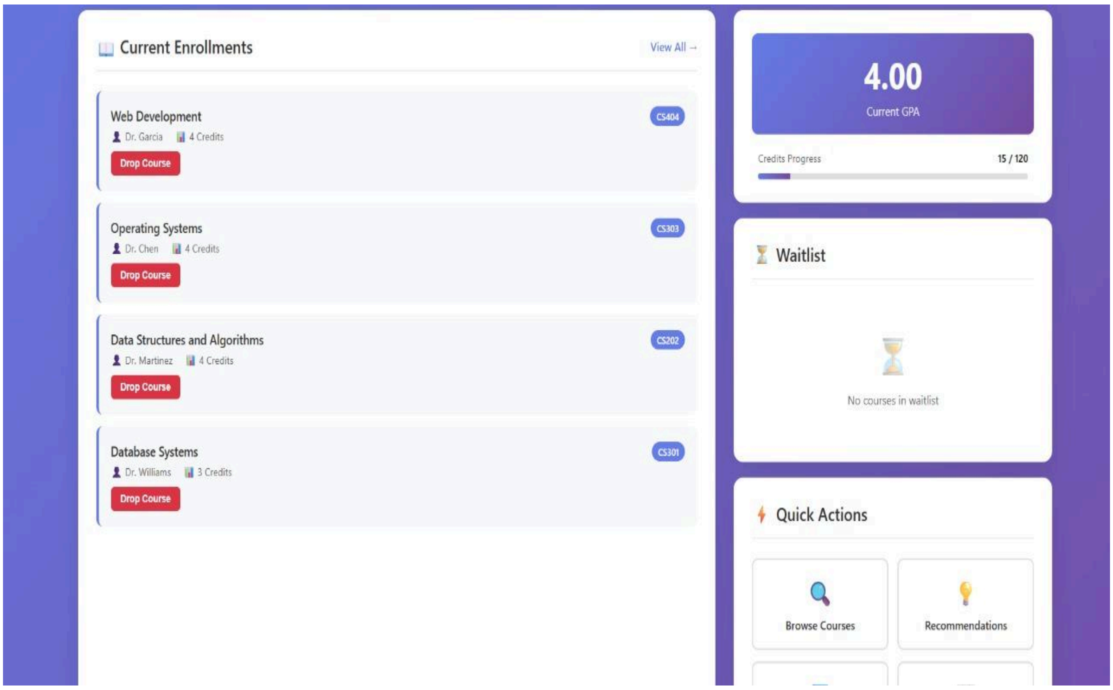
CS402 Recommended
Computer Networks
Dr. White
Credits: 3
Available Seats: 15
Network protocols and architecture
Enroll Now

CS201 Recommended
Data Structures

MATH301 Recommended
Discrete Mathematics

ENG101 Recommended
English Composition

Fig 5.6 Course Enrollments



Current Enrollments — View All →

Web Development — CS404
Dr. Garcia · 4 Credits
Drop Course

Operating Systems — CS303
Dr. Chen · 4 Credits
Drop Course

Data Structures and Algorithms — CS202
Dr. Martinez · 4 Credits
Drop Course

Database Systems — CS301
Dr. Williams · 3 Credits
Drop Course

4.00
Current GPA

Credits Progress 15 / 120

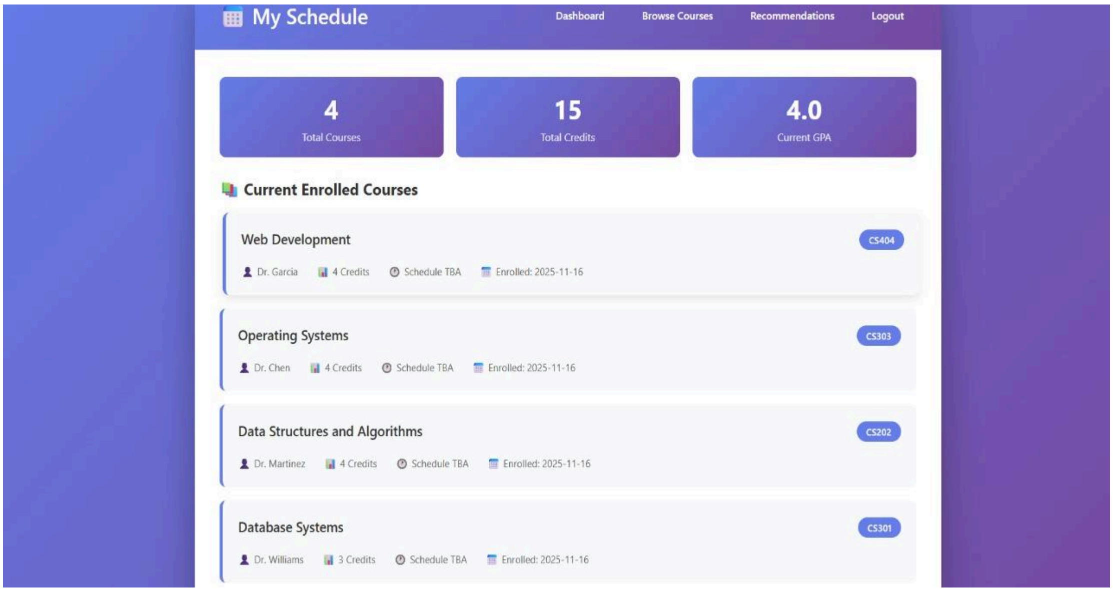
Waitlist

No courses in waitlist

Quick Actions

Browse Courses | Recommendations

Fig 5.7 My Schedule



My Schedule

4
Total Courses

15
Total Credits

4.0
Current GPA

Current Enrolled Courses

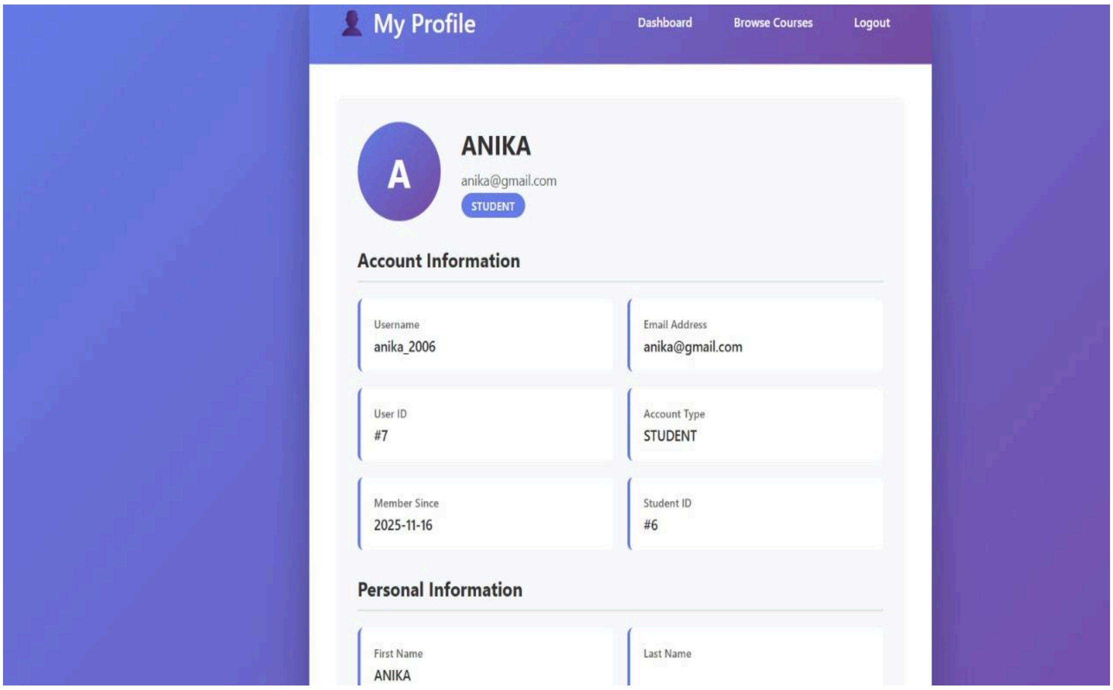
Web Development CS404
Dr. Garcia 4 Credits Schedule TBA Enrolled: 2025-11-16

Operating Systems CS303
Dr. Chen 4 Credits Schedule TBA Enrolled: 2025-11-16

Data Structures and Algorithms CS202
Dr. Martinez 4 Credits Schedule TBA Enrolled: 2025-11-16

Database Systems CS301
Dr. Williams 3 Credits Schedule TBA Enrolled: 2025-11-16

Fig 5.8 My Profile



My Profile

ANIKA
anika@gmail.com
STUDENT

Account Information

Username
anika_2006

Email Address
anika@gmail.com

User ID
#7

Account Type
STUDENT

Member Since
2025-11-16

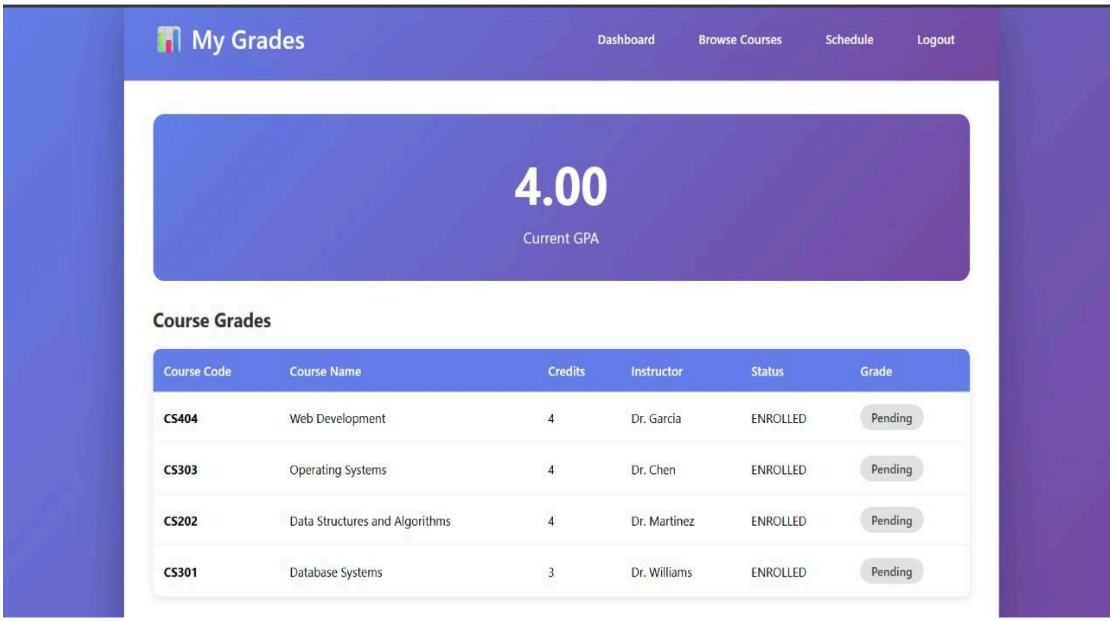
Student ID
#6

Personal Information

First Name
ANIKA

Last Name

Fig 5.9 My Grades



4.00

Current GPA

Course Grades

Course Code	Course Name	Credits	Instructor	Status	Grade
CS404	Web Development	4	Dr. Garcia	ENROLLED	Pending
CS303	Operating Systems	4	Dr. Chen	ENROLLED	Pending
CS202	Data Structures and Algorithms	4	Dr. Martinez	ENROLLED	Pending
CS301	Database Systems	3	Dr. Williams	ENROLLED	Pending

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

In such a way, with the help of our project, customers will be able to check the list of bookings and can register themselves to avail a cab. The booking system clearly represents the available data of the customers for booking and the number of bookings using a booking log and management becomes easier. In future people will be able to book cabs according to the data available in the system and with respect to the availability. Hence this project makes the user and other advantages to be benefitted in all possible ways.

REFERENCES

1. https://www.w3schools.com/sql/
2. https://dev.mysql.com/doc/
3. https://developer.mozilla.org/en-US/docs/Web/JavaScript
4. https://www.geeksforgeeks.org/online-course-registration-system/
5. https://www.javatpoint.com/dbms-tutorial
6. https://www.tutorialspoint.com/dbms/