COURSE ENROLLMENT SYSTEM

A MINI-PROJECT REPORT
Submitted by

KAVIARASI M 240701241

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
DECEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **"COURSE ENROLLMENT SYSTEM"** is the bonafide work of **"KAVIARASI M"** who carried out the project work under my supervision.

SIGNATURE
Mrs.S.SATHIYAVATHI
ASSISTANT PROFESSOR SG

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 Course Enrollment System is designed to provide a seamless and efficient way for students to register for courses online, replacing traditional paper-based methods. This system enables students to browse available courses, select their desired options, and enroll with just a few clicks, significantly improving convenience and accessibility. Administrators can use the platform to easily manage course information, monitor enrollment statistics, and oversee student registrations. By leveraging a secure backend database and user-friendly interface, the system ensures that all data is stored reliably and can be accessed in real-time. The project aims to reduce manual errors, save administrative time, and enhance overall transparency within the academic institution. Ultimately, the Course Enrollment System helps bridge the gap between students and administration, ensuring a smoother, faster, and more reliable enrollment experience for all stakeholders.

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 **Mrs.S.SATHIYAVATHI,** for her 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. KAVIARASI M

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Course Enrollment System streamlines student registration and course selection through a user-friendly digital platform. By automating manual tasks, it minimizes errors and accelerates administrative processes. The project offers role-based dashboards for both students and administrators, providing secure access to essential academic features. Through real-time data management and intuitive interfaces, the system enhances accountability and transparency for all stakeholders.

1.2 SCOPE OF THE WORK

This project encompasses the development of a robust application allowing students to register, browse, and enroll in courses online, while administrators can manage courses and view student enrollments. The platform integrates secure authentication, error handling, and data validation. Scalability and modularity are key aspects, making future enhancements, such as analytics or payment integration, straightforward. Compatibility with modern hardware and software environments ensures extended usability.

1.3 PROBLEM STATEMENT

Existing course registration processes in many institutions are manual or semi-digitized, causing frequent errors, slow operations, and poor accessibility for remote users. Administrators struggle to track enrollments and maintain up-to-date records, while students face delays and uncertainties in course availability. The lack of centralized control and instant communication hampers overall effectiveness. The need for a scalable, automated system is evident for modern educational requirements.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The primary aim is to provide an automated solution for student registration and course management. Objectives include simplifying the entire enrollment workflow, ensuring secure user authentication, enabling real-time access to academic data, and improving overall system efficiency. Additional goals are reducing paperwork, minimizing administrative burden, and accommodating future academic services and integrations.

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	:	JAVA(Swing/AWT)
Back - End	:	MySql
Language	:	Java,SQL
Database Driver	:	MySql Connector/J
Tools Used	:	VS code,Mysql Workbench

CHAPTER 3

MODULE DESCRIPTION

AUTHENTICATION : Login and sign-up for students & admins

STUDENT MODULE : Student registration, course search, enrollment history view

ADMIN MODULE : Add, update, delete student/course records, track all enrollments.

ENROLLMENT MODULES : Connects students and courses, records enrollment events.

DASHBOARD MODULE : Graphical views of system data for rapid decision making.

CHAPTER 4

SAMPLE CODING

Sample 1 ADMIN DASHBOARD

```
package com.courseenrollment.gui;

import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.awt.*;

import java.sql.Connection;

import java.sql.PreparedStatement

import java.sql.ResultSet;

import java.sql.SQLException;

import com.courseenrollment.database.DatabaseConnection;

public class AdminDashboard extends JFrame {

    private int adminId;

    private String adminName;

    private DefaultTableModel studentsModel, coursesModel, enrollmentsModel;

    public AdminDashboard(int adminId, String adminName) {

        this.adminId = adminId;

        this.adminName = adminName;

        setTitle("Admin Dashboard - " + adminName);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setBounds(100, 100, 1200, 800);
        initializeComponents();
        loadAllData();
        setLocationRelativeTo(null);
    }
     private void loadAllData() {
        loadStudentsData();
        loadCoursesData();
        loadEnrollmentsData();
    }
 private void loadStudentsData() {
        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT student_id, CONCAT(first_name, ' ', last_name) as
name, email, phone, registration_date FROM students";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();
            studentsModel.setRowCount(0);
            while (resultSet.next()) {
                studentsModel.addRow(new Object[]{
                    resultSet.getInt("student_id"),
                    resultSet.getString("name"),
                    resultSet.getString("email"),
```

```
                resultSet.getString("phone"),
                resultSet.getTimestamp("registration_date")
            });
        }
    } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error loading students: " +
e.getMessage());
    }
}
private void loadCoursesData() {
    try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT course_id, course_code, course_name, credits,
max_capacity, course_fee FROM courses";
        PreparedStatement statement = connection.prepareStatement(query);
        ResultSet resultSet = statement.executeQuery();
        coursesModel.setRowCount(0);
        while (resultSet.next()) {
            coursesModel.addRow(new Object[]{
                resultSet.getInt("course_id"),
                resultSet.getString("course_code"),
                resultSet.getString("course_name"),
                resultSet.getInt("credits"),
```

```
                resultSet.getInt("max_capacity"),
                "$" + resultSet.getBigDecimal("course_fee")
            });
        }
    } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error loading courses: " +
e.getMessage());
    }
  }
  private void loadEnrollmentsData() {
    try (Connection connection = DatabaseConnection.getConnection()) {
 String query = "SELECT s.student_id, CONCAT(s.first_name, ' ', s.last_name) as
student_name, " +
                "c.course_code, c.course_name, e.enrollment_date " +
                "FROM enrollments e " +
                "JOIN students s ON e.student_id = s.student_id " +
                "JOIN courses c ON e.course_id = c.course_id " +
                "ORDER BY s.student_id, c.course_code";
        PreparedStatement statement = connection.prepareStatement(query);
        ResultSet resultSet = statement.executeQuery();
        enrollmentsModel.setRowCount(0);
        while (resultSet.next()) {
```

```
            enrollmentsModel.addRow(new Object[]{
                resultSet.getInt("student_id"),
                resultSet.getString("student_name"),
                resultSet.getString("course_code"),
                resultSet.getString("course_name"),
                resultSet.getTimestamp("enrollment_date")
            });
        }
    } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error loading enrollments: " +
e.getMessage());  }
  }
}
```

Sample 1 depicts the display code, that gets the data from the database i.e. being stored there and represented on students interested courses with the enrolled coursesi.e. being already specified.

CHAPTER 5

SCREEN SHOTS

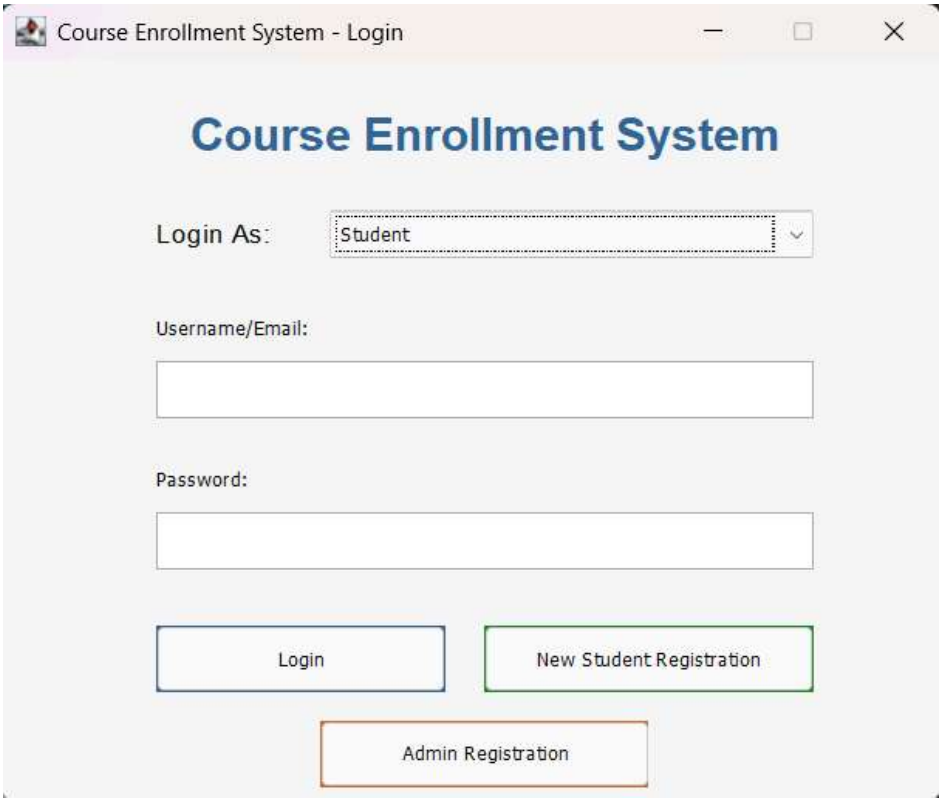

Fig 5.1 Login page

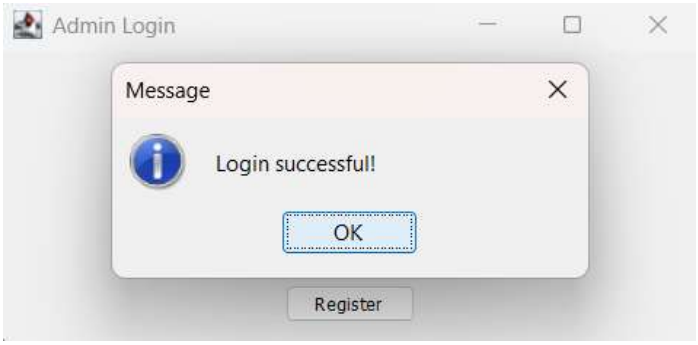

Fig 5.2 Admin Login

Fig 5.3 Admin Dashboard

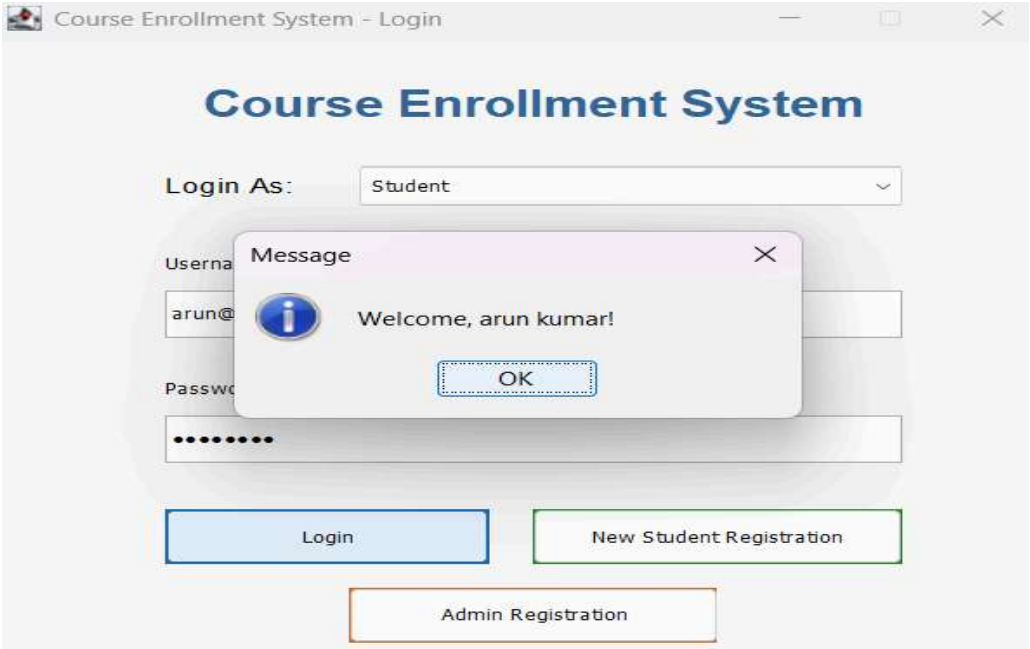

Fig 5.4 Student Login

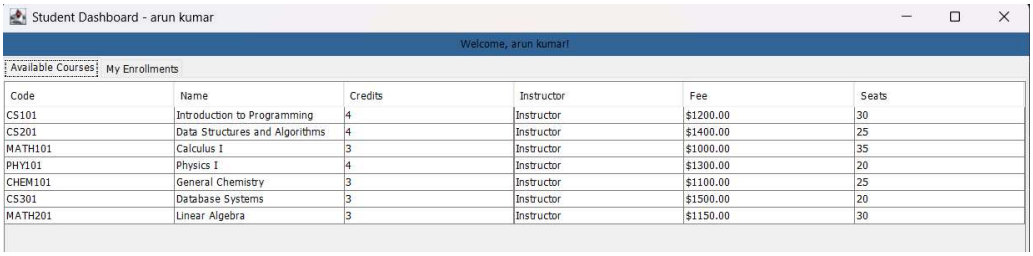

Fig 5.5 Available Courses

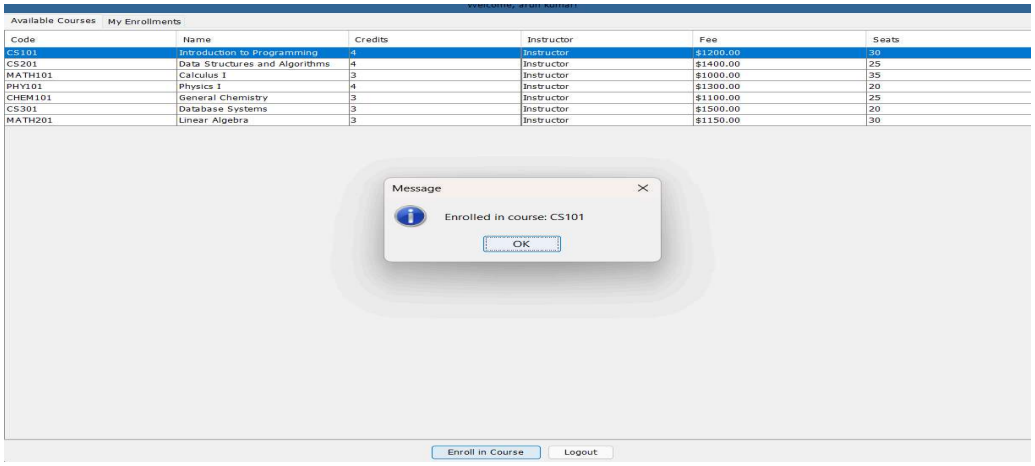

Fig 5.6 Enrollment

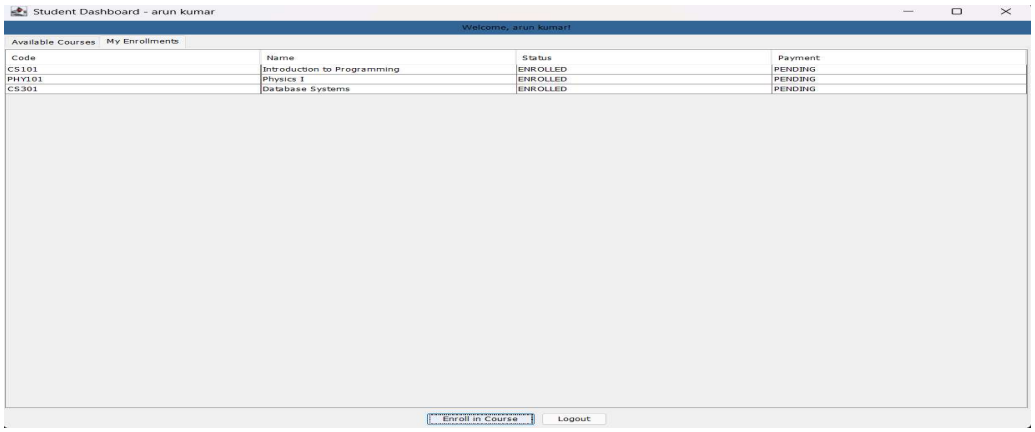

Fig 5.6 Enrolled Courses

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

This project delivers a reliable, scalable, and secure framework for automating student course enrollment processes. It substantially improves campus administrative efficiency and user experience. Planned future enhancements include adding mobile access, reporting analytics, support for online payments, and integration with institutional ERP and LMS platforms for complete academic automation.

REFERENCES

1. https://www.w3schools.com/java/
2. https://www.w3schools.com/sql/ w3schools.com
3. https://www.geeksforgeeks.org/java/
4. https://www.w3schools.com/sql/
5. https://dev.mysql.com/doc/
6. https://code.visualstudio.com/doc