

Students Course & Skills Management System.

by
Shittu Rofeeq Adeleke
IDEAS/24/87636

Supervisor: **Mr Suleiman Hayat Tuge**
Dept: **Software Engineering**

>> Baze University - IDEAS program <<

September, 2024.

Acknowledgments

I am sincerely grateful to the Almighty Lord for His countless blessings, provision, guidance, and enlightenment throughout this project. Without His grace and support, none of this would have been possible. I would also like to express my heartfelt gratitude to my parents for their unwavering support, encouragement, and guidance. Their belief in me and their constant motivation have been instrumental in my journey. Additionally, I want to thank *myself* for the *dedication and hard work* put into this project, and to everyone who contributed, directly or indirectly, to its successful completion. Your support and encouragement have made a significant difference, and *I am deeply appreciative*.

Abstract

This project involves the development of a **Student Course and Skills Management System** aimed at improving the management of student information, course enrollments, and skill tracking in educational institutions. Many current systems are inefficient and prone to errors, which this project seeks to address through a web-based solution.

The system was built using **HTML, CSS, JavaScript, and Node.js with Express**, providing both a user-friendly interface and an efficient backend. It allows users to easily manage courses, track student skills, and handle authentication.

Testing showed that the system improves data handling and reduces errors in managing student records. Overall, this project delivers a practical solution for streamlining student management tasks, with room for future improvements.

>> Table of Content <<

Acknowledgement	2
Abstract	3
Table of content	4

6. Introduction

- 6.1 Background	6
- 6.2 Problem Statement	6
- 6.3 Objectives	6
- 6.4 Scope	7
- 6.5 Significance	7
- 6.6 Methodology Overview	7

7. Literature Review

- 7.1 Introduction	8
- 7.2 Relevant Theories/Models	8
- 7.3 Related Work	8
- 7.4 Gaps in Existing Work	8

8. System Requirements

- 8.1 Functional Requirements	9
- 8.2 Non-functional Requirements	9
- 8.3 Use case Diagrams	9

9. System Design

- 9.1 System Architecture	11
- 9.2 Design Models	11
- 9.3 Database Design	12
- 9.4 User Interface Design	13
- 9.5 Technology Stack	14

10. Implementation

- 10.1 Development Environment 14
- 10.2 Code Structure 15
- 10.3 Key Algorithms/Modules 15
- 10.4 My Challenges 15

11. Testing and Evaluation

- 11.1 Testing Strategy 16
- 11.2 Test Cases 16
- 11.3 Evaluation 16
- 11.4 Limitations 16

12. Deployment

- 12.1 Deployment Environment 17
- 12.2 Deployment Process 17
- 12.3 Configuration Management 17
- 12.4 Post-Deployment Monitoring 17

13. Conclusion and Future Work

- 13.1 Summary of Work 18
- 13.2 Contributions 18
- 13.3 Future Work 18

References 19

Appendices 19

6. Introduction

6.1 Background

Managing student information, course data, and skills assessment has become a growing challenge for educational institutions. With the increasing volume of students and courses, traditional methods of data management, which often involve manual processes or disconnected systems, are no longer sufficient. These outdated approaches lead to inefficiencies, such as time-consuming data entry, errors in records, and difficulty in tracking student progress accurately. Institutions need a modern, centralised system that can streamline these processes, ensuring that student information is easily accessible, reliable, and up-to-date. This project seeks to create such a solution by addressing these challenges with a user-friendly, web-based management system.

6.2 Problem Statement

Educational institutions often face challenges with outdated, inefficient systems for managing student courses and skills. Manual processes and disconnected systems lead to errors, data inconsistencies, and extra administrative work. These issues make it hard to track student progress accurately. This project aims to solve these problems by creating a unified platform that simplifies data management, ensuring a more integrated and reliable solution.

6.3 Objectives

The main objective of this project is to develop a web-based system that allows for effective management of student courses and skills. The specific goals of the system include:

- Providing a centralised platform for storing and accessing student data.
- Streamlining course enrollment processes.

- Offering efficient tools for tracking and updating student skills and achievements.
- Improving the accuracy of data handling to reduce errors and inconsistencies.

6.4 Scope

This project will focus on building a system that facilitates:

- Course management, including the creation, editing, and deletion of courses.
- Student profile management, allowing administrators to add, update, and view student details.
- Skill tracking, where student progress and acquired skills can be documented and reviewed. While the system will handle core administrative tasks, it will not extend to advanced features such as automatic course recommendations, detailed analytics, or collaborative learning tools.

6.5 Significance

This system simplifies and automates key administrative tasks, reducing human error and data redundancy. It ensures student information is accurate and easily accessible, improving how institutions manage courses and student data. The system also supports better decision-making and offers a scalable solution that can grow with the needs of schools and universities.

6.6 Methodology Overview

This project uses a full-stack approach, with **HTML**, **CSS**, and **JavaScript** for the responsive frontend and **Node.js** with **Express** for the backend. A **MySQL** database securely stores student and course data, ensuring quick and reliable access. The focus throughout development is on performance, security, and usability.

7. Literature Review

7.1 Introduction

Effective student management systems are essential for handling the growing complexity of student data, courses, and skills within educational institutions. This section explores existing models and their limitations, drawing insights from a minimum of 3 to 20 peer-reviewed articles.

7.2 Relevant Theories/Models

Information management systems are guided by theories that prioritise data accuracy, accessibility, and efficiency. One relevant model is the **Data Life Cycle Management** (DLM) framework, which focuses on how data is managed from its creation to eventual disposal, ensuring data integrity throughout a student's academic journey.

7.3 Related Work

Several student management systems exist, such as **Moodle** and **Blackboard**, which offer features like course enrollment and student record management. However, they can be too complex for smaller institutions or lack integrated skill tracking. Additionally, these systems are often expensive and difficult to customise for unique institutional needs.

7.4 Gaps in Existing Work

While many systems exist, there are significant gaps, especially in providing a cost-effective and streamlined solution for smaller institutions. Most current systems either have cumbersome interfaces or do not integrate course management and skill tracking effectively. This project aims to address these limitations by offering a simpler, unified platform tailored to the needs of smaller institutions.

8. System Requirements

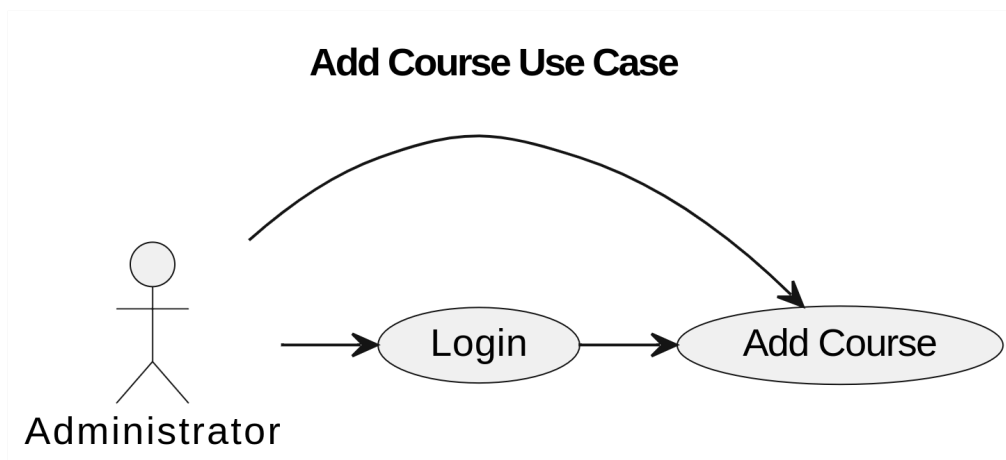
8.1 Functional Requirements

- **Course Management:** Administrators can add, update, and delete courses, while students can view and enrol in courses.
- **Skill Tracking:** The system allows tracking and updating of student skills and achievements.
- **Report Generation:** The system should generate reports on student progress, course enrollments, and skill development.

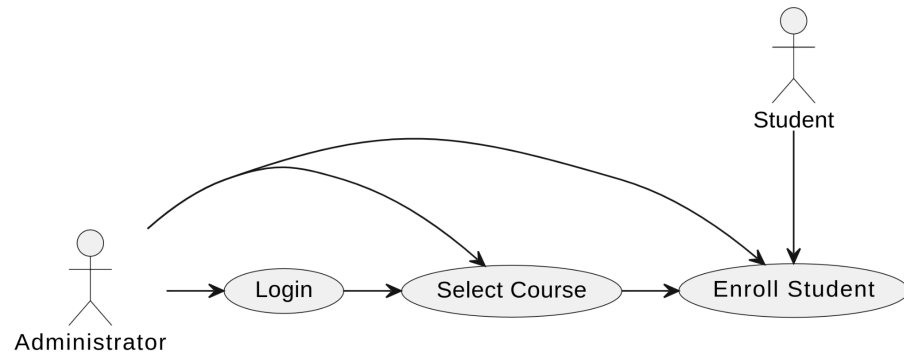
8.2 Non-functional Requirements

- **Performance:** The system should be able to support many concurrent users without significant performance issues.
- **Security:** Implement strong authentication and data protection measures to ensure user data is secure.
- **Usability:** The interface should be user-friendly for both administrators and students, making it easy to navigate and use.

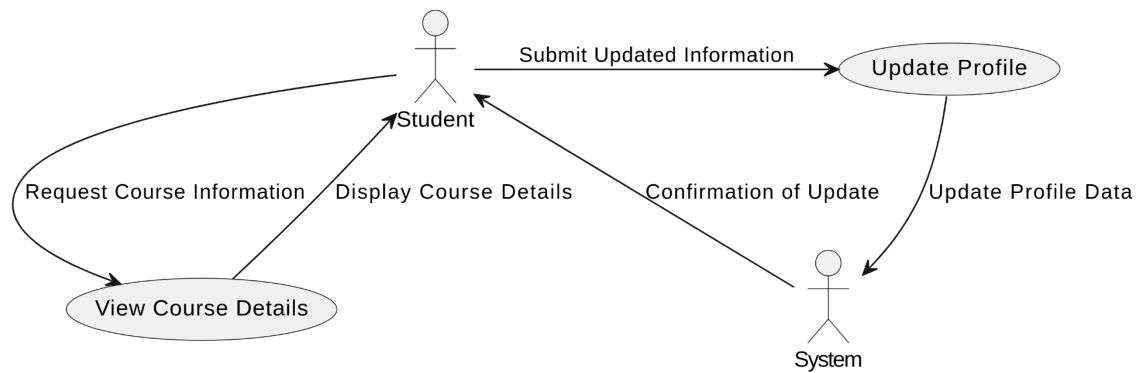
8.3 Use Case Diagrams



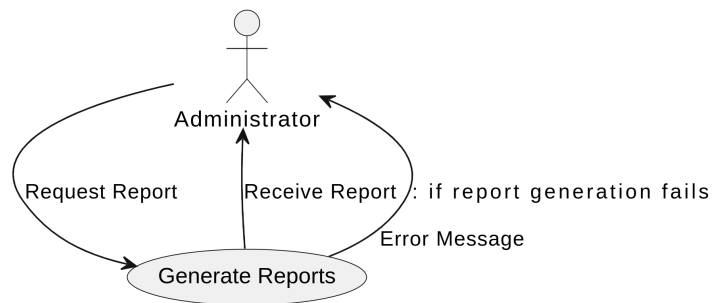
Enroll Student Use Case



Update Student Profile & View Course Details Use Case



Generate Reports Use Case



9. System Design

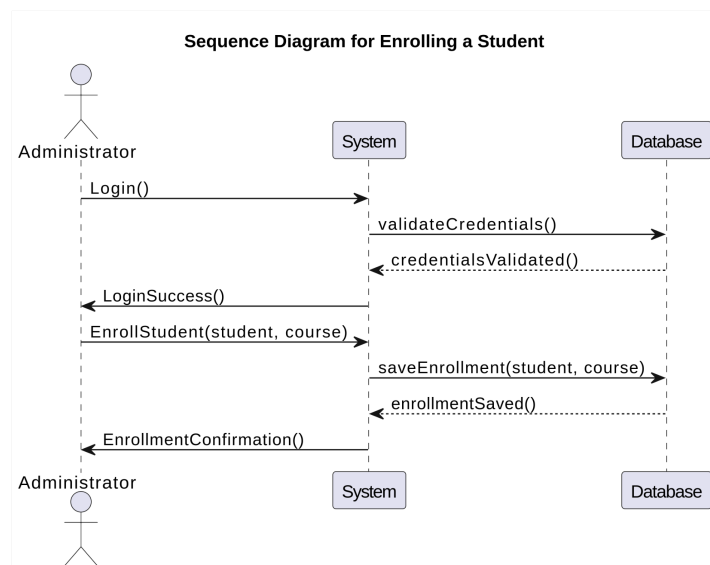
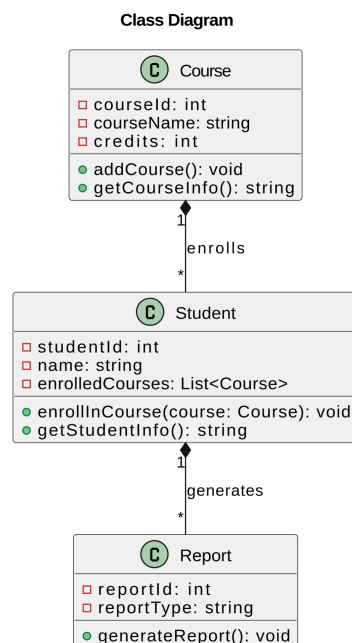
9.1 System Architecture

The system follows a **client-server architecture** where the client (user's browser) interacts with the server to request and display data. The frontend, built with **HTML**, **CSS**, and **JavaScript**, communicates with the backend through **Node.js** and **Express**. The server processes these requests, interacts with the **MySQL** database to fetch or update data, and sends responses back to the client.

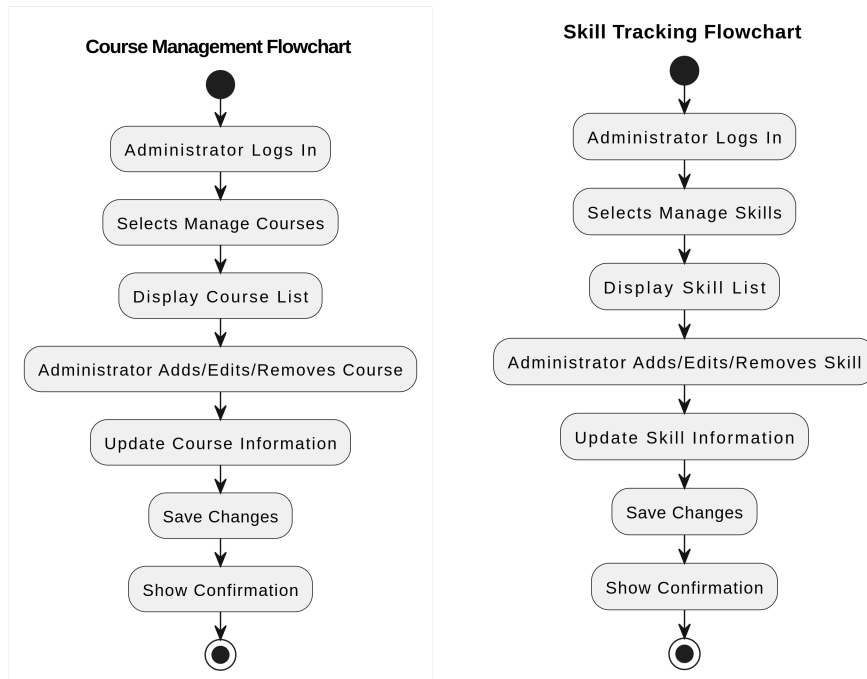
9.2 Design Models

Design models such as **UML diagrams** and **flowcharts** will be used to represent the system's structure and processes. For example:

- **UML Diagrams** will show the system's components and their interactions.

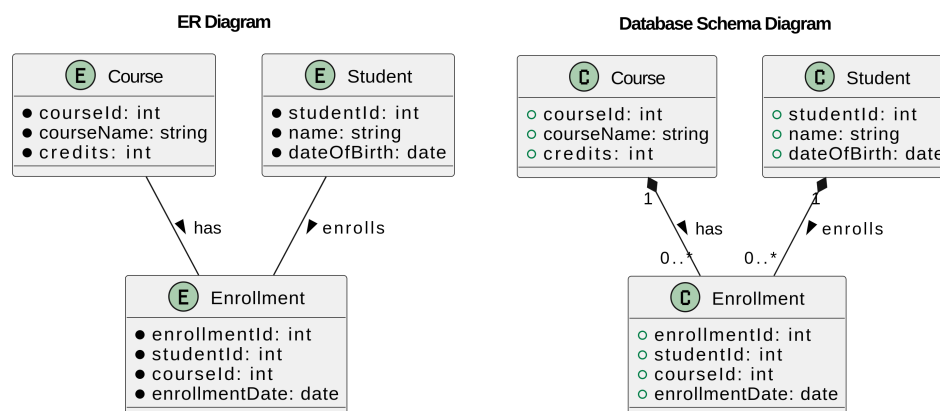


- **Flowcharts** will illustrate how different functions, like course management or skill tracking, are handled within the system.



9.3 Database Design

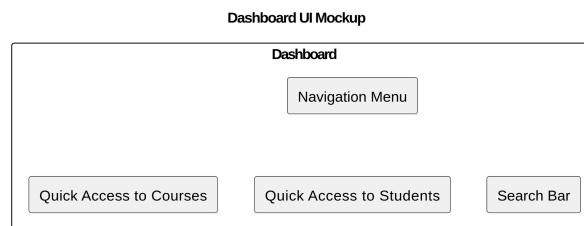
The database design is represented with **Entity-Relationship (ER) diagrams** that outline the schema and relationships between different entities, such as students, courses, and skills. The schema includes tables for storing user information, course details, and skill records, with primary and foreign keys to maintain data integrity.



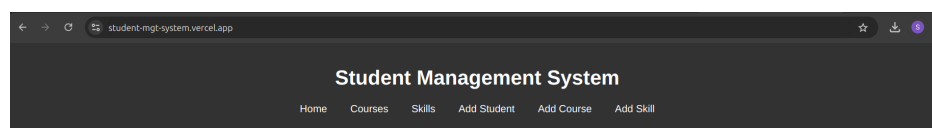
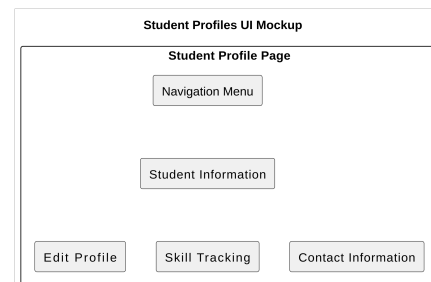
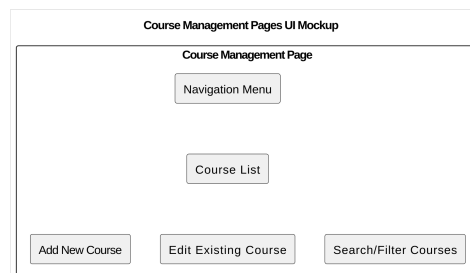
9.4 User Interface Design

The user interface design focuses on creating a clean and intuitive layout. Key UI components include:

- **Dashboard:** For quick access to courses and student information.



- **Course Management Pages:** For adding and editing course details.
- **Student Profiles:** For viewing and updating student information and skills. Illustrations and mockups of these components will help visualise the final look and feel of the system.



Welcome to the Student Management System

Introduction

This system allows you to manage students, courses, and skills efficiently. Use the navigation links above to add new students, courses, or skills to the system.

System Overview

Our student management system provides the following features:

- **Add and manage students:** Enter student details and keep track of their courses and skills.
- **Add and manage courses:** Create and organize courses that students can enroll in.
- **Add and manage skills:** Track skills that students can acquire or improve.

Add Student

Name Courses Skills

9.5 Technology Stack

The project uses the following technologies:

- **Frontend: HTML, CSS, and JavaScript** for creating a responsive and interactive user interface.
- **Backend: Node.js and Express** for handling server-side logic and API requests.
- **Database: MySQL** for storing and managing data securely.

10. Implementation

10.1 Development Environment

For development, I used **Visual Studio Code** as my primary IDE, which provided a versatile and user-friendly environment for coding. I worked primarily in the **Ubuntu terminal** for command-line tasks. Additionally, I utilised **Node.js** and **Express** for the backend, and **MySQL** for database management. For debugging and code structuring, I relied on various online resources and tools to guide me through the process.

10.2 Code Structure

The codebase is organised into distinct sections:

- **Frontend Code:** Includes **HTML**, **CSS**, and **JavaScript** files for creating the user interface.
- **Backend Code:** Contains **Node.js** and **Express** code for handling server-side logic and API routes.
- **Database Scripts:** Includes SQL scripts for setting up and managing the **MySQL** database schema.
- **Configuration Files:** Contains files like **package.json** for dependencies and **.env** for environment variables.

10.3 Key Algorithms/Modules

- **Authentication Module:** Manages user login and registration, ensuring secure access.
- **Course Management:** Handles CRUD operations for courses, including adding, updating, and deleting.
- **Skill Tracking:** Updates and retrieves student skill records.
- **Report Generation:** Compiles data into reports for administrative use.

10.4 Challenges

One major challenge was learning **Node.js** and **Express** for the backend, which was new and quite overwhelming at first. I faced difficulties with debugging, code structuring, and getting everything to work together smoothly. To overcome these challenges, I sought help from online resources, used **AI tools** for debugging, and relied on various forums and documentation to get my code functioning correctly. These resources were invaluable in helping me fix issues and structure my code effectively.

11. Testing and Evaluation

11.1 Testing Strategy

- **Unit Testing:** Test individual parts of the code to make sure they work.
- **Integration Testing:** Check how different parts of the system work together.
- **System Testing:** Ensure the whole system functions correctly.
- **User Testing:** Get feedback from real users to see if it meets their needs.

11.2 Test Cases

- **User Authentication:** Check login with correct and incorrect credentials.
Result: Worked fine for correct credentials; handled errors for incorrect ones.
- **Course Enrollment:** Test adding and viewing courses.
Result: Successfully added and displayed courses.
- **Skill Tracking:** Verify updating and retrieving skill data.
Result: Data updated and retrieved correctly.

11.3 Evaluation

- **Performance:** The system works well with good number of student
- **Usability:** Easy to use; user feedback was positive.
- **Functionality:** Main features work as expected with no major issues.

11.4 Limitations

- **Scalability:** Performance might drop with very high user numbers.
- **Features:** Lacks advanced features like predictive analytics.
- **Customization:** Extra development needed for specific institutional needs.

12. Deployment

12.1 Deployment Environment

The system is deployed using **Vercel**, which hosts the web application. The source code is stored and managed on my **GitHub** account..

—> Deployed Link: <https://student-mgt-system.vercel.app/>

12.2 Deployment Process

1. **Code Push:** Push code changes to the **GitHub** repository.
2. **Automatic Deployment:** Vercel automatically deploys the latest code from the GitHub repo.
3. **Review:** Check the live site on Vercel to ensure everything is working as expected.

12.3 Configuration Management

- **Version Control:** Managed via **GitHub**, where all code changes are tracked and versioned.
- **Configuration:** Environment variables and settings are configured in Vercel for the deployment.

12.4 Post-Deployment Monitoring

- **Monitoring:** Regularly check the live application on Vercel for any issues.
- **Maintenance:** Update the GitHub repository with any bug fixes or improvements, which will automatically be deployed by Vercel.

13. Conclusion and Future Work

13.1 Summary of Work

This project successfully developed a web-based system for managing student courses and skills. We created a user-friendly interface for course management, skill tracking, and report generation. The system is functional, reliable, and easy to use, meeting the initial objectives.

13.2 Contributions

This project makes a real difference for educational institutions by simplifying how they manage student information. It combines course management and skill tracking into one easy-to-use platform, which helps cut down on errors and saves time for administrators. By streamlining these processes, the system helps institutions keep their data accurate and accessible, making day-to-day operations smoother and more efficient.

13.3 Future Work

Future improvements could include adding advanced features like predictive analytics for skill development, integrating a recommendation system for courses, and enhancing scalability to handle more users. Customization options for different institutional needs could also be explored.

14. References

1. Websites:

- Node.js Foundation. (2024). *Node.js Documentation*. <https://nodejs.org/en/docs/>
- Express.js Foundation. (2024). *Express.js Documentation*. <https://expressjs.com/>
- Johnson, M. (2021, July 14). How to deploy a web app with Vercel. *Web Development Today*. <https://www.webdevtoday.com/vercel-deploy>

2. YouTube resources

- Code with Yousuf (2023) Connecting Node.js to a MySQL Database | Node.js Connect MySQL with Node app <https://youtu.be/JWia8kUPrRs?si=1f4TfcGVzMTxdjyB>

15. Appendices

Appendix A: Code Snippets and Diagrams

- **Code Snippets:** Relevant code snippets demonstrating key functionalities or complex logic in the project can be accessed on my GitHub repository at <https://github.com/rofeeqshittu/student-mgt-system>.

Appendix B: User Manual and Installation Guide

- **User Manual:** All instructions for using the system are available directly on the deployed website's homepage.
- **Installation Guide:** As the system is hosted online, no installation is required. Users only need a browser and access to the internet to use the system. No special setup or commands are needed.