Digital E-learning with Gamification

Submitted By

Irfan Kafil Areyan

ID: 221-35-991

Supervised By

Dr. Nusrat Jahan

Head,

Department of Information Technology & Management, Daffodil International University

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

Department of Software Engineering

Faculty of Science and Information Technology

Supervisor Approval Form

Fall 2025	B.Sc. In SWE	Campus: DSC

Student Name	Student ID
Irfan Kafil Areyan	221-35-991

Project/Thesis Information	
Project/Thesis Title	Digital E-learning Platform
Type of work	Development-based Project

Supervisor information	
Supervisor Name	Dr. Nusrat Jahan
Supervisor Initial	NJ
Completed Credit till now	133
How many credits in this semester	6
Amount (Due)	BDT 0.00
Supervisor Consent	☑ Yes ☐ No

Supervisor Signature 

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF PROJECT AND COPYRIGHT

Author's Full Name	:	Irfan Kafil Areyan
Date of Birth	:	20-10-2002
Title	:	Digital E-learning with Gamification
Academic Session	:	2022-2025

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my project to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

(Student's Signature)

(Supervisor's Signature)

221-35-991
Student ID
Date: 13 December, 2025

Dr. Nusrat Jahan
Name of Supervisor
Date: 13 December, 2025

NOTE: * If the Project is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this project and in my/our* opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

(Supervisor's Signature)

Full Name : **Dr. Nusrat Jahan**

Position : Head

Date : 13 December, 2025

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

(Student's Signature)

Full Name : **Irfan Kafil Areyan**
ID Number : 221-35-991
Date : 13 December, 2025

Digital E-learning with Gamification

IRFAN KAFIL AREYAN

Project submitted in fulfillment of the requirements
For the award of the degree of
Bachelor of Science/Master of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

DECEMBER 2025

ACKNOWLEDGEMENTS

First of all, I am grateful to The Almighty Allah for making me eligible to complete this project.

Then I would like to thank my supervisor “Dr. Nusrat Jahan”, “Head”, Department of Information Technology & Management. I am extremely grateful and indebted to her for her expert, sincere and valuable guidance and encouragement extended to me.

I would also wish to extend my gratitude to them who took part in the survey in this project. The validation survey would not have been conducted successfully without their strong involvement and contribution.

At this point, I wish to thank the members of the faculty of the Department of Software Engineering who helped and inspired me with their support.

And lastly, I would like to make a cheer to my parents, for their unconditional support, love and without this I would not have reached this far.

DEDICATION

I therefore declare that I have done this project under the oversight of "**Dr. Nusrat Jahan**", **"Head"**, Department of Information Technology & Management, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

ABSTRACT

This project outlines building a modern, user-friendly online learning platform. The platform is engineered to facilitate a dynamic digital learning environment by providing distinct, role-based interfaces for students and instructors. Core functionalities include user authentication, course enrollment and management, progress tracking, and assessment through quizzes and assignments. A key feature allows instructors to monitor individual student performance via a dedicated search, enabling targeted support. The system aims to streamline the educational process, making it more accessible, interactive, and data-driven for all users.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

The education today is evolving. As additional learning continues to be conducted over the Internet, learning institutions require online services that are both practical and user-friendly. Although there are numerous systems, most of them are cumbersome and cumbersome- mimicking a complex toolbox in which it is difficult to locate the appropriate tool. Educators are overwhelmed with statistics and cannot find a straight forward answer to a crucial question: How this or that student is doing? They also pass up time in switching between screens to see the grades and progress of a student. This project aims to fix that. We are creating a new people-centered learning platform that will be user friendly. To students it will represent a clean and simple center through which they can search their courses, get access to materials and view their progress. To the teachers, it will ease the course administration and most importantly provide teachers with a simple method of searching the performance of any given student with just a mere search. Simply put, we are not merely creating any other system; we are creating a superior and more accommodating digital learning and teaching house.

1.1.1 Context and Relevance

The current project is placed in the rapidly developing sphere of Technology-Enhanced Learning (TEL) and EdTech (Educational Technology) industry around the world. The wider scope of research is to determine how digital technologies and platforms can be used to enhance the accessibility, efficiency, and customization of education. The problematic issues in this respect are: User Engagement: To reduce low completion rates, the digital platforms should be easy to use and use. Data-Driven Instruction: Although information is plentiful, most systems do not provide easy mechanisms of helping teachers to turn this information into actionable data to help at-risk students. Disjointed Ecosystems: Universities tend to have several fragmented systems, and the experience is cumbersome to both the students and the faculty. This project is a direct response to such challenges with the creation of a centralized LMS which is specific to a Learning Center of a university. It opens up a major prospect to improve the process of education by automating the administration part of the work process, offering the educators direct means to monitor student performance, and offering the students more supportive and unified digital environment to learn in.

1.1.2 Problem Identification

The idea of this project is that it is a fundamental issue of contemporary digital education: there is no unified learning platform that will successfully connect the course administration and student learning analytics. Although a typical educational establishment may implement a regular Learning Management System (LMS) like Moodle or Canvas, they have a few highly problematic limitations that slow down their operation: Fragmentation of platforms: Traditional LMS solutions are used as fragmented tools and are not integrated systems. Critical elements such as course enrollment, content delivery and performance monitoring work independently, and they demand users to use various interfaces which provides a fragmented experience to either the teachers or the students. Poor Accessibility: Although a lot of data is gathered about a student performance, the current systems do not have easy-to- understand ways of converting such data into useful information. Teachers experience the burden of an arduous system of generating several reports and compiling data, which do not allow identifying students at risk in time and provide the required academic assistance. Generic Implementation Strategy: The LMS commercial systems assume the philosophy of universal design which does not support a particular institutional workflow and teaching methodology. This leads to complexity of interfaces, unused function and high cognitive loads in end users leading to reduced engagement and adoption.

1.1.3 Purpose and Justification

Importance and Value Addition

The current project has that urgent significance as it has a direct impact on the main purpose of the university student success and teaching excellence. It transforms the digital learning infrastructure into a non-oppressive storyteller to a supportive ecosystem.

The project adds value in three major ways:

For Students: It is valuable because it provides centralized, streamlined, and transparency of the learning journey. Students experience the advantage of having one easy-to-use portal to all their blended learning solutions: enrolling, tracking their progress, etc. This lessens mental strain and administrative tension and offers opportunities to study more and less to navigate complicated systems.

For Faculty (Teachers): It is a valuable addition since it saves time, and facilitates data-based teaching. The performance dashboard and student search are dedicated and turn raw data into actionable insight and immediate wisdom. The time saved by instructors on manual data compilation can be used to create effective pedagogical decisions, which enable them to spot students who require assistance, provide them with feedback in a manner that is personally relevant and effective, and reduce their time taken to identify them, making them more efficient teachers

For the Institution (DIU): The project will create a strategic value by making the university a trusted entity in the area of innovative education. The platform that is tailor-made and efficient shows the intent to use technology in actual educational enhancement. It enhances operational effectiveness, offers an excellent user experience that enhances satisfaction, and will be a scalable model that can be used in future digital projects to keep the BLC ahead of blended learning.

1.1.4 Scope

The project will provide a complete web based Blended Learning Center (BLC) site. The development will be designed on core modules which will form the critical digital ecosystem of blended learning in the university.

The project will cover the following major areas:

User Management & Authentication: Students and Teachers are provided with a secure system of user registration, logging in, managing their profiles, and having role-based access control.

Course Management: Teacher course creation and management and publication. This involves setting course information, module design and the posting of learning materials.

Course Enrollment & Catalog: A student facing catalog of the courses offered with an easy enrollment procedure.

Learning Content Delivery: A dedicated interface for enrolled students to access and navigate through course modules and materials.

Assessment & Evaluation: Fundamental elements to develop quizzes, make assignments and mark student work.

Progress Tracking & Analytics: A dashboard for students to view their own progress and grades. In the case of teachers, there should be a dashboard in the central place with a particular search option of a student to retrieve the report on their performance by using the Student ID.

Achievement System: This is a simple system of offering and presenting badges or certificates after fulfilling course milestones.

1.2 Project Planning and Initiation

Feasibility Study:

Phase 1 Preliminary Analysis & Project Scope Definition: It will entail the establishment of the fundamental goals and scope of the interactive learning platform. The project scope will include creating a web-based LMS with the help of PHP Laravel and Blade templates and attention to improved interaction with the user and efficient management of the courses. The critical deliverables consist of user authentication, course management applications, interactive assessments tools, tracking of progress and the analytics dashboards. The initial examination supports the necessity of a more interactive alternative to currently used LMS solutions and the obvious exclusion of mobile applications development, integration of payment, and live video conferencing options to stay focused.

Phase 2 Market Feasibility Analysis (or Market Research): Market research shows that there is a high demand in the market to have interactive and user friendly learning management systems in the learning institutions. Study of the available solutions such as Moodle and Canvas shows a lack of engagement among the users and intuitive interfaces. The target market will consist of universities, colleges, and training centers

that are interested in enhancing their digital learning facilities. Competitive analysis reveals that there is a high potential in the platform with better usability and student-teacher interaction on the feature-rich platforms. The surveys carried out by the users prove that they are unhappy with complicated navigation in the existing LMSs.

Phase 3 Technical Feasibility Analysis: The project is technically feasible based on the PHP Laravel framework, as well as Blade templating. The features of Laravel (Eloquent ORM, authentication, middleware) have strong base and can be developed quickly. The MySQL database has the power to store and retrieve academic data. Its technology stack offers responsive design by integrating Bootstrap. Laravel has high security measures, good documentation and a community support that gives minimal risks in terms of technical risks.

Phase 4 Financial Feasibility Analysis: Open-source technologies are used to optimize the cost of development. The project budget includes the entire amount of manpower in the development, testing and deployment infrastructure. It can be paid off using fewer administrative overheads and increased educational outcomes. Cloud hosting solutions are cost effective in terms of operational costs. The project will show high cost /benefit ratio since the efficiency of the development of Laravel and the possibilities of its mass implementation in educational departments

1.3 Target User Profile and Tentative Elicitation Process

Target User

Primary Users:

Students: Students who study courses, have assignments, follow the progress.

Teachers/Instructors: Teachers creating courses, managing content, monitoring student performance

Administrators: System administrators dealing with user accounts and platform configuration.

1.3.1 User profile

The platform supports two main user types: Students and Teachers. Student profiles track learning progress, course enrollments, completion statistics, and provide access to a personalized dashboard with courses, calendar, and activities. Teacher profiles manage assigned courses, student performance metrics, lesson materials, and teaching schedules. Both profiles include personal information, role-based navigation, search functionality, and access controls tailored to their specific educational needs within the system.

Table 1.3.1.1: User Profile for Student

User Class	**Note on Characteristics**
Type of user	Student
Age range	18-26
Frequency of use	Regularly
Mandatory	Yes, for enrolled courses
Computer experience	Intermediate
Education	Undergraduate
goal	Enroll in courses and access course contents
Language skills	English(fluent)
Number of users	Many
Training	Minimal
Others system use	Google slides, Google drive etc
Way of working	Goal-oriented

Table 1.3.1.2: User Profile for Faculty members

User Class	**Note on Characteristics**
Type of user	Faculty Members
Age range	26-60
Frequency of use	Regularly
Mandatory	Yes, for course delivery and management
Computer experience	Intermediate or Advance
Education	Graduate or Postgraduate
goal	Provide course contents and guidance
Language skills	English (fluent)
Number of users	Many
Training	Basic onboarding
Others system use	Google slides, Google drive etc

Way of working	Task-Oriented

1.3.2 Elicitation Process

A multi-method strategy will be used to ensure that all the stakeholders needs are incorporated in the final product and this will be done to ensure that the ultimate product meets the needs of all the stakeholders.

Structured Interviews: It aims at learning strategic goals, institutional limits and technical integration intersections. Interviews will have a script according to which they will be conducted yet it will be flexible to delve into new issues.

Focus Groups: Students and instructors will be separated into separate focus group sessions. These facilitated discussions will cover user experience with the current systems, areas of pain, and features they would want in the new platform. This is the best approach to come up with a thought and to learn the group dynamics and general workflow.

Online Surveys: Wide scale, quantitative surveys shall be administered to more students and faculty. They will collect information about the particular feature preferences, the rates of usability of the existing systems, and demographics. Surveys are useful when it is necessary to validate the results of interviews and focus groups and describe a broader scope of opinions.

User Observation (Contextual Inquiry): It will be observed how students and instructors use the existing LMS platforms in their natural environment. This tool is useful in revealing needs, and practical usability challenges that may not be expected in an interview or survey.

Prototype Feedback: When the first wireframes or interactive mockup have been developed, these will be exposed to a few users. They will give their feedback on the navigation, layout and workflow to make changes to the design to make it better and better before the development process occurs.

1.4 Project Block Diagram

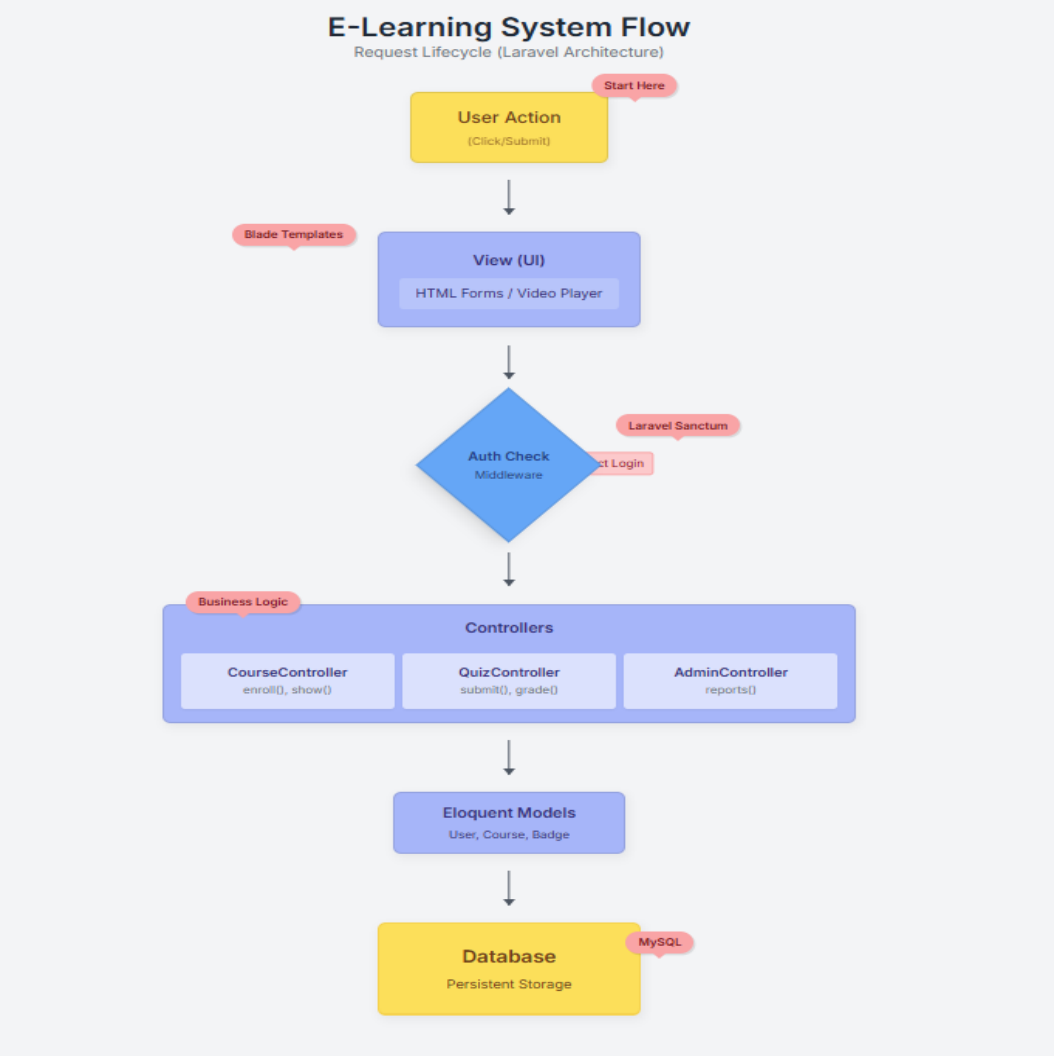


Figure 1.4: Block Diagram

1.5 System Requirements

1.5.1 Hardware Requirements

Development Machine:

- **Processor:** Intel Core i5 or similar, 2.0 GHz or more
- **RAM:** 8 GB minimum (16 GB suggested)
- **Storage:** 10 GB storage

Production Server:

- **Processor:** Multi-core 64-bit processor (2 or more cores)
- **RAM:** 4 GB minimum (8 GB suggested)
- **Storage:** 50 GB SSD storage (OS, application, and database)

1.5.2 Software Requirements

Development Environment:

- **Operating System:** Windows 10/11, macOS, or Linux (Ubuntu 20.04 or more)
- **Web Server:** Apache
- **PHP:** Version 8.1 or higher
- **Database:** MySQL 8.0 minimum
- **Framework:** Laravel 10.x
- **Composer:** For PHP dependency management
- **Code Editor:** VS Code, PHPStorm, or any other

1.5.3 Constraints and Dependencies

Constraints:

Budget: Development and initial deployment should be performed according to the set project budget.

Time: The timeframe that the project has to be completed and be prepared to pilot test is within 6 months

Technology: The system has limitations of application of the PHP Laravel framework and relational database (MySQL).

Human Resources: The project relies on the fact that there is one full-stack developer and a part-time UI/UX designer.

Dependencies:

External:

- Availability and stability of the chosen Laravel framework and its packages.
- Dependency on network infrastructure and internet connectivity of the university.

Internal:

- Availability of important stakeholders (students, faculty, admin) to elicit the requirement as well as feedback.
- The IT department of the university will provide availability of a testing server and final production server.

1.6 Project Scheduling

1.7 Summary

The chapter has provided the necessary foundation of the learning platform project. It started by setting up the project context and making it clear which gaps exist in digital learning tools and what particular problems our solution will be able to alleviate. We have done an extensive feasibility study that has ensured that the project has been viable through technical, operational, and financial aspects. The chapter described our target users, who are students, teachers and administrators and explained the methodological way of collecting their requirements. We defined the functional measures of the system, as well as the non-functional measures, which it should have, with regards to the hardware, software, and constraints that are required. Lastly, we gave a realistic 6-month project plan, and risk management plan. Taken together, this chapter gives a sound background and roadmap towards the design and implementation work ahead.

CHAPTER 2: DESIGN AND IMPLEMENTATION

2.1 Introduction

The chapter explains the systematic way of converting the conceptual structure of the interactive learning platform to a working, powerful, and user-friendly software system. It traces the whole process of architectural designs to a functioning application aiding in a detailed account of the design decisions, technologies underpinning the design, as well as development techniques used.

The chapter starts with the description of the high-level Architectural Design of the system, which defines the selected patterns and data flow which will guarantee the scalability and maintainability. Next, it goes into the Technology Stack, with reasons as to why PHP was chosen with the Laravel framework with Blade templating engine to make a streamlined and well-structured development. The essence of the design is captured in the Detailed System Design that contains Entity-Relationship Diagrams (ERD) to capture the database schema and graphical representation of major workflows of the user.

The next chapter concerns the Implementation stage, where the ways of how the designs are translated into code, how to organize the development process, and how the technical issues could be resolved. Lastly, it will be followed by the Testing Strategy which is used to test the functionality, performance, and security of the system so that the product sent to be deployed meets all the stated requirements and is outfitted to be used. The chapter is the ultimate documentation of the conceptualization, construction and testing of the system.

2.2 Functional Requirements

FR01	**Registration**
Description	Every user (Student/Teacher) must be registered to able to use the system
Stakeholder	Student, Teacher

FR02	**Login**
Description	Teacher and Student must login before using the Digital E-learning platform
Stakeholder	Student, Teacher, Admin

FR03	**Update Profile**
Description	Use will be able to update profile by adding or changing information including uploading profile photo.

Stakeholder	Teacher, Student

FR04	**View Course**
Description	Every user will be able to view all the available courses in the platform without login
Stakeholder	Student, Teacher

FR05	**Enroll Courses**
Description	Student will be able to enroll in different courses using enrollment key provided by assigned teacher for that course
Stakeholder	Student

FR06	**Access Modules and Contents**
Description	Every enrolled student will be able to access course modules and contentsof that course
Stakeholder	Student

FR07	**Take Quizzes**
Description	Student will be able to take or attempt quizzes for specific course set by the assigned teacher.
Stakeholder	Student

FR08	**Submit Assignments**
Description	User will be able to submit assignments set by course teacher in different file format(doc, pdf etc)
Stakeholder	Student

FR09	**Track Progress**

Description	Student will be able to see the progress of enrolled courses which will be measured based on different task completion
Stakeholder	Student

FR10	**View Leaderboard**
Description	Student will be able to see leaderboards in their profile where their performances, activities and everything will be counted and rank or badge will be provided and they will be able to see the rank of other student
Stakeholder	Student

FR11	**Manage Courses**
Description	User will be able to manage courses by adding and removing contents and files of assigned courses.
Stakeholder	Teacher

FR12	**Manage Achievements**
Description	User will be able to set badges or points for different tasks which will be claimed after the completion of the tasks
Stakeholder	Teacher, Admin

FR13	**View Student Results**
Description	Teacher will be able to see student's SGPA and CGPA by searching with their student ids
Stakeholder	Teacher

FR14	**Manage User**
Description	The system will allow admin to view, add, disable, enable, remove any users from the system
Stakeholder	Admin

FR15	Manage Authentication
Description	System will verify the users credentials during login.
Stakeholder	Admin

FR16	View Analytics
Description	The system will allow user to view analytical reports in dashboard(course specific matric)
Stakeholder	Admin, Teacher

2.3 Non-Functional Requirements

2.3.1 Performance

- Normally, the system must be able to load all web pages in 3 seconds.
- The platform must be able to accommodate 2000 parallel users at a time.
- 95 percent of requests performed against the database should take less than 200ms.
- Student records should take less than 2 seconds to respond to the search.

2.3.2 Reliability& Availability

- The system will have a 99.5% uptime in academic calendar periods.
- Daily automated backups with the capability of point in time recovery.
- Data integrity should be ensured and transaction support on very important operations should be supported.

2.3.3 Security

- Safe authentication using Laravel's built-in encryption and hashing
- Student, teacher and administrator role-based access control.
- Protection against common web vulnerabilities (SQL injection, XSS, CSRF)
- Secure session management with configurable timeout periods

- Encryption of important information about users.

2.3.4 Usability

- Responsive design compatible with desktop, tablet, and mobile devices
- Easy intuitive navigation that needs minimum training to use.
- Coherent user interface in all the system modules.

2.3.5 Maintainability

- Clean, documented code following Laravel and PHP standards
- Modular architecture with easy updates and additions made.
- Full-fledged logging of debugging and monitoring.
- Database migrations for version-controlled schema changes

2.3.6 Scalability

- Horizontal scaling by load balancing.
- Effective caching of data that is highly demanded.
- Data optimization with growing volumes of data.

2.4 Object-oriented System design using UML

2.4.1 Use Case Diagram

Figure 2: Use case Diagram

2.4.2 Case Description

Table 2.4.2.1 Case Description: Registration

<table>
<tr><td>Use Case</td><td>Registration</td></tr>
<tr><td>Goal</td><td>Users can register to sign in to the system.</td></tr>
<tr><td>Precondition</td><td>User must use Edu email and fill all the fields of the registration form</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Successfully Registered!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Registration failed”</td></tr>
<tr><td>Primary Actors:</td><td>Student</td></tr>
<tr><td>Trigger</td><td>User will request a registration form to fill up</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “Sign Up” Button</td></tr>
<tr><td>2.</td><td>Provide registration form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Submit” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and notify
Successfully Registered!!!</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>1.1.a. Checked By the system & Notify by “Please! Fill Up the Box”</td></tr>
<tr><td>4.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>4.1.a.. Show Error Message</td></tr>
<tr><td>5.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>5.1.a.. Notification: “Credentials are not recorded”</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.2 Case Description: Login

<table>
<tr><td>Use Case</td><td>Login</td></tr>
<tr><td>Goal</td><td>Users can login to the system.</td></tr>
<tr><td>Precondition</td><td>Users must be registered in the system.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Successfully Logged in!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Login failed”</td></tr>
<tr><td>Primary Actors:</td><td>Student, teacher</td></tr>
<tr><td>Trigger</td><td>User will request a registration form to fill up</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td><table>
<tr><td>1.</td><td>Press “Login” Button</td></tr>
<tr><td>2.</td><td>Provide login form</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Submit” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system allows the user to access the dashboard and shows “Login Successful”</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>1.1.a. Checked By the system & Notify by “Please! Fill Up the Box”.</td></tr>
<tr><td>4.1</td><td>Wrong credentials</td></tr>
<tr><td></td><td>4.1.a. Show “incorrect email or password”</td></tr>
<tr><td>5.1</td><td>The user isn’t registered</td></tr>
<tr><td></td><td>5.1.a. Notification: “Sign up first”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.3 Case Description: Update Profile

<table>
<tr><td>Use Case</td><td>Update Profile</td></tr>
<tr><td>Goal</td><td>Users can update profile information</td></tr>
<tr><td>Precondition</td><td>Users must be logged into the Digital E-learning system.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Successfully updated!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Profile not updated”</td></tr>
<tr><td>Primary Actors:</td><td>Customer</td></tr>
<tr><td>Trigger</td><td>User will request a edit profile form to fill up</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td><table>
<tr><td>1.</td><td>Click “Profile” Button</td></tr>
<tr><td>2.</td><td>Go to edit profile</td></tr>
<tr><td>3.</td><td>Enter Information</td></tr>
<tr><td>4.</td><td>Press “Save” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!
Profile Updated!!</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Checked By the system & Notify by “Please! Fill Up the Box”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Profile not updated”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.4 Case Description: View Course

<table>
<tr><td>Use Case</td><td>View Course</td></tr>
<tr><td>Goal</td><td>Users can view available courses</td></tr>
<tr><td>Precondition</td><td>Users must visit the site using the proper link.</td></tr>
<tr><td>Success End Condition</td><td>!!!Show available courses!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “404 not found”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Student, Teacher</td></tr>
<tr><td>Trigger</td><td>User will click on browse courses</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td>
<table>
<tr><td>1.</td><td>Press “Browse course” Button</td></tr>
<tr><td>2.</td><td>Course list will appear</td></tr>
<tr><td>3.</td><td>Click “view details” for any course</td></tr>
<tr><td>4.</td><td>Single page view will be visible</td></tr>
<tr><td>5.</td><td>Course view successful</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>No course listed</td></tr>
<tr><td></td><td>4.1.a. Notify by “No course available”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.5 Case Description: Enroll Courses

<table>
<tr><td>Use Case</td><td>Enroll course</td></tr>
<tr><td>Goal</td><td>A student can enroll in an available course.</td></tr>
<tr><td>Precondition</td><td>User must be logged in as a student and the course must be active</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Successfully Enrolled!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Enrollment Failed”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Student</td></tr>
<tr><td>Trigger</td><td>User will click the “Enroll” button for specific course</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Select any course from course list</td></tr>
<tr><td>2.</td><td>Navigate to course details</td></tr>
<tr><td>3.</td><td>Enter enrollment key</td></tr>
<tr><td>4.</td><td>Press “Enroll” Button.</td></tr>
<tr><td>5.</td><td>User is granted access to the course materials</td></tr>
<tr><td>6.</td><td>The system shows “Enrollment Successful”</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user already enrolled</td></tr>
<tr><td></td><td>4.1.a. Notification “User already enrolled”</td></tr>
<tr><td>5.1</td><td>The enrollment key is wrong</td></tr>
<tr><td></td><td>5.1.a. Show “Incorrect key” Message.</td></tr>
<tr><td>6.1</td><td>The system not responding.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Try again later”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.6 Case Description: Access Modules and Contents

<table>
<tr><td>Use Case</td><td>Access Modules and Contents</td></tr>
<tr><td>Goal</td><td>Users can access course modules and contents</td></tr>
<tr><td>Precondition</td><td>User must login to the system</td></tr>
<tr><td>Success End Condition</td><td>!!!Modules and contents accessible!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Can’t access contents”</td></tr>
<tr><td>Primary Actors:</td><td>Student</td></tr>
<tr><td>Trigger</td><td>User will click on enrolled course</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td><table>
<tr><td>1.</td><td>Select any course</td></tr>
<tr><td>2.</td><td>Get enrolled using key</td></tr>
<tr><td>3.</td><td>Click “Enroll”</td></tr>
<tr><td>4.</td><td>Enrollment successful</td></tr>
<tr><td>5.</td><td>Course modules and contents are visible</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>The user is not enrolled!</td></tr>
<tr><td></td><td>1.1.a “Please! Enroll first”.</td></tr>
<tr><td>4.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>4.1.a. Show Error Message.</td></tr>
<tr><td>5.1</td><td>Course content not available.</td></tr>
<tr><td></td><td>5.1.a. Notification: “content not updated”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.7 Case Description: Take Quizzes

<table>
<tr><td>Use Case</td><td>Take Quizzes</td></tr>
<tr><td>Goal</td><td>Users can take or attempt quizzes.</td></tr>
<tr><td>Precondition</td><td>Users must enroll in that specific course</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Attempt Successful!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Attempt failed”</td></tr>
<tr><td>Primary Actors:</td><td>Student</td></tr>
<tr><td>Trigger</td><td>User will request for attempting a quiz</td></tr>
<tr><td>Description / Main
Success
Scenario</td><td>
<table>
<tr><td>1.</td><td>Select the course</td></tr>
<tr><td>2.</td><td>Go through contents</td></tr>
<tr><td>3.</td><td>Search for available quiz</td></tr>
<tr><td>4.</td><td>Select quiz</td></tr>
<tr><td>5.</td><td>Attempt quiz</td></tr>
<tr><td>6.</td><td>The system saves the details and shows them !!!
Successfully attempted!!!</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user did not attempt on time!</td></tr>
<tr><td></td><td>4.1.a. “Quiz unavailable”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Attempt not recorded”</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.8 Case Description: Submit Assignment

Use Case	Submit Assignment
Goal	Users can Submit an assignment
Precondition	Users must login and enrolled in specific course
Success End Condition	Notification: !!!Successfully Submitted!!!
Failed End Condition	Notification: “Submission failed”
Primary Actors: Secondary Actors:	Student
Trigger	User will request to submit assignment
Description / Main Success Scenario	1. Select a course 2. Search for available assignment 3. Select the assignment 4. Press “Attempt submission” Button. 5. Select file from device 6. Successfully Submitted!!! Notify
Alternative Flows	1.1 System Error 1.1.a. Try Again!! 4.1 The user used wrong format 4.1.a. Notify by “File format not supported”. 5.1 The system did not respond 5.1.a. Show Error Message. 6.1 Deadline expired 6.1.a. Notification: “Submission closed”
Quality Requirements	The user Will fill up all the details in 30 minutes.

Table 2.4.2.9 Case Description: Track Progress

<table>
<tr><td>Use Case</td><td>Track Progress</td></tr>
<tr><td>Goal</td><td>Users can track progress of enrolled course.</td></tr>
<tr><td>Precondition</td><td>Users must be enrolled in that course</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Progress visible!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Progress Not Found”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Student</td></tr>
<tr><td>Trigger</td><td>User will request to track progress</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Select a course</td></tr>
<tr><td>2.</td><td>Access contents</td></tr>
<tr><td>3.</td><td>Complete tasks</td></tr>
<tr><td>4.</td><td>Click on “Progress”</td></tr>
<tr><td>5.</td><td>Progress visible</td></tr>
<tr><td>6.</td><td>Course Progress tracking successful</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not complete any task!</td></tr>
<tr><td></td><td>4.1.a. Notify by “No progress yet”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the details.</td></tr>
<tr><td></td><td>6.1.a. Notification: “Progress not recorded”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.10 Case Description: Manage Course

<table>
<tr><td>Use Case</td><td>Manage Course</td></tr>
<tr><td>Goal</td><td>Users can manage courses by adding or removing contents</td></tr>
<tr><td>Precondition</td><td>Users must be registered and assigned to any course.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Course Management Accessible!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Course Management not accessible”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Teacher,

Admin</td></tr>
<tr><td>Trigger</td><td>User will request to modify course content and modules</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Select assigned course</td></tr>
<tr><td>2.</td><td>Check modules</td></tr>
<tr><td>3.</td><td>Add new contents</td></tr>
<tr><td>4.</td><td>Remove unnecessary contents</td></tr>
<tr><td>5.</td><td>Content updated</td></tr>
<tr><td>6.</td><td>Successfully Updated!!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not use right format!</td></tr>
<tr><td></td><td>4.1.a. Notify by “File format not supported”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t save the update.</td></tr>
<tr><td></td><td>6.1.a. Notification: “update failed”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

Table 2.4.2.11 Case Description: View Student Result

<table>
<tr><td>Use Case</td><td>View Student Result</td></tr>
<tr><td>Goal</td><td>Users can search for student result.</td></tr>
<tr><td>Precondition</td><td>Users must logged in to the system.</td></tr>
<tr><td>Success End Condition</td><td>Notification: !!!Result visible!!!</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Unavailable”</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Teacher</td></tr>
<tr><td>Trigger</td><td>User will search for result using student id</td></tr>
<tr><td>Description / Main

Success

Scenario</td><td><table>
<tr><td>1.</td><td>Go to student result</td></tr>
<tr><td>2.</td><td>Click on search box</td></tr>
<tr><td>3.</td><td>Enter student Id</td></tr>
<tr><td>4.</td><td>Click “Search” Button.</td></tr>
<tr><td>5.</td><td>Result shown</td></tr>
<tr><td>6.</td><td>Search Successful!! Notify</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user entered wrong id!</td></tr>
<tr><td></td><td>4.1.a. Notify by “No result found”.</td></tr>
<tr><td>5.1</td><td>The system did not respond</td></tr>
<tr><td></td><td>5.1.a. Show Error Message.</td></tr>
<tr><td>6.1</td><td>The system Doesn’t have the record.</td></tr>
<tr><td></td><td>6.1.a. Notification: “No result found”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>The user Will fill up all the details in 30 minutes.</td></tr>
</table>

2.4.3 Activity Diagram

2.4.3.1 Activity Diagram: Update Profile

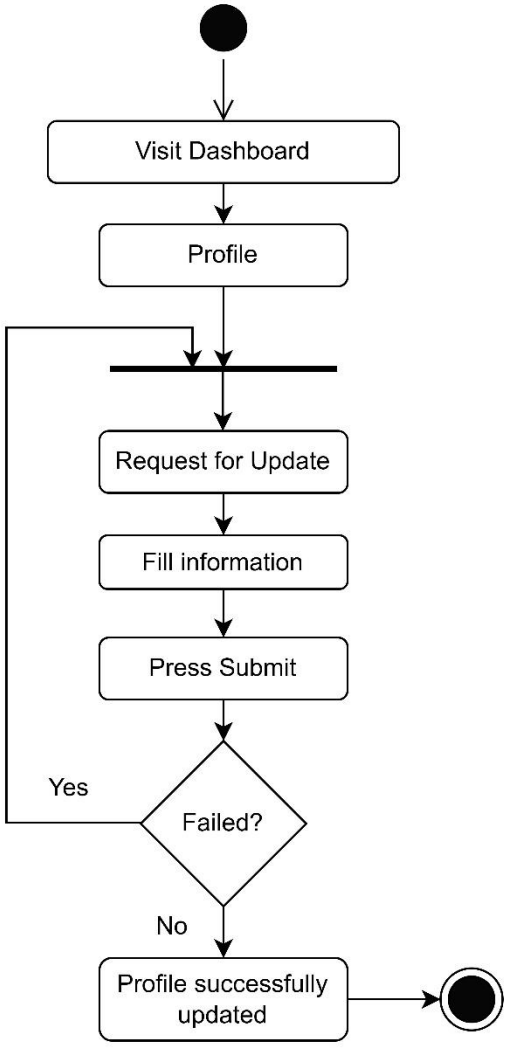


Figure 2.4.3.1: Update Profile

2.4.3.2 Activity Diagram: Course Enrolment

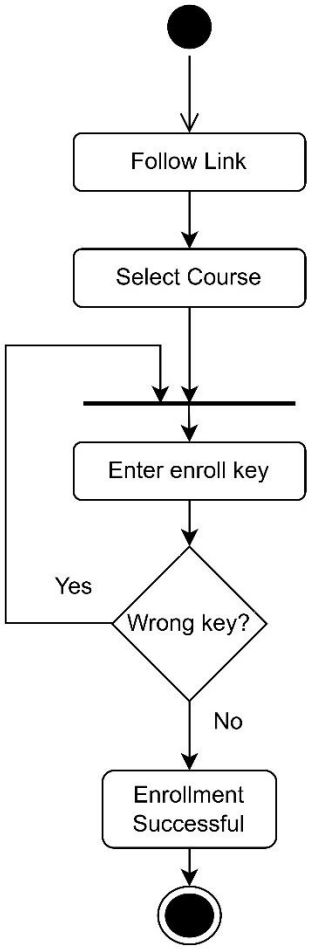


Figure 2.4.3.2: Enroll Courses

2.4.3.3 Activity Diagram: Access Module and Contents

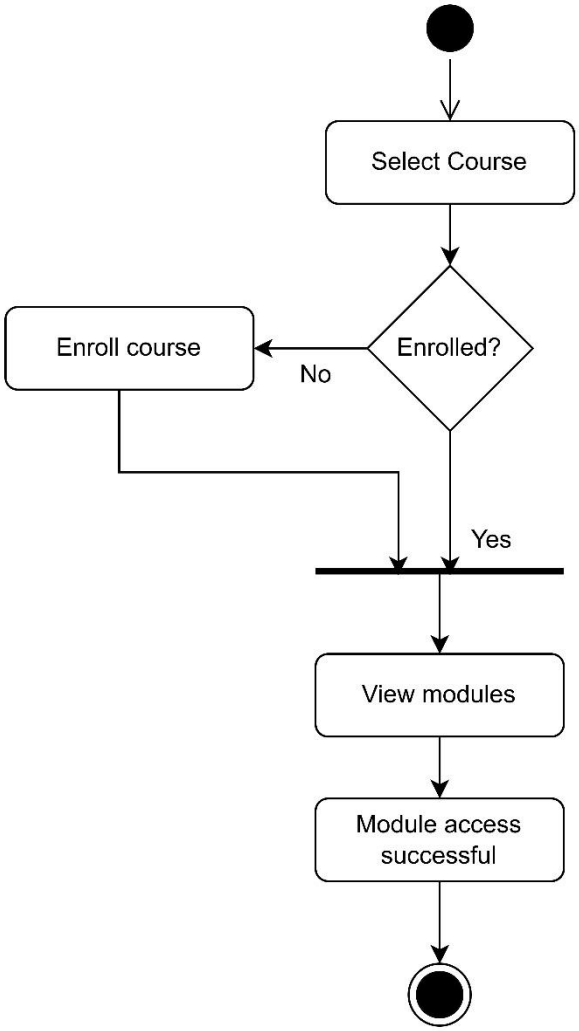


Figure 2.4.3.3: Access Modules and Contents

2.4.3.4 Activity Diagram: Take Quizzes

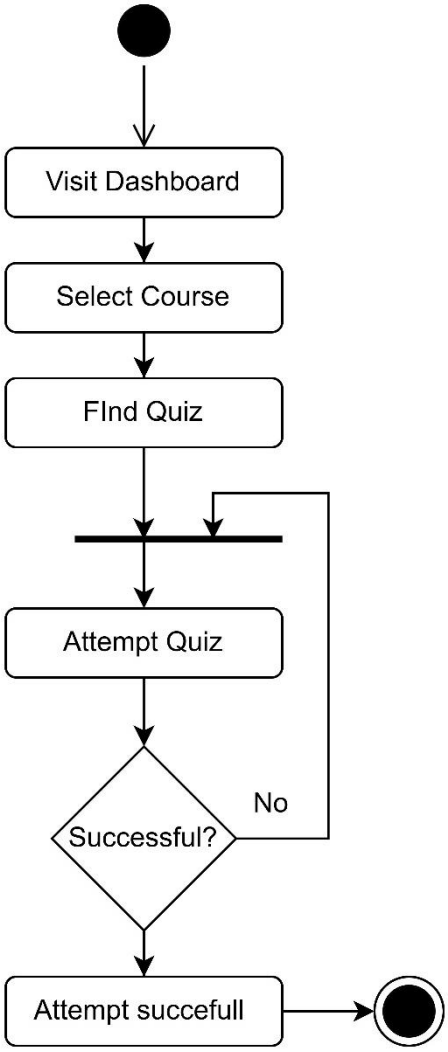


Figure 2.4.3.4: Take Quizzes

2.4.3.5 Activity Diagram: Track Progress

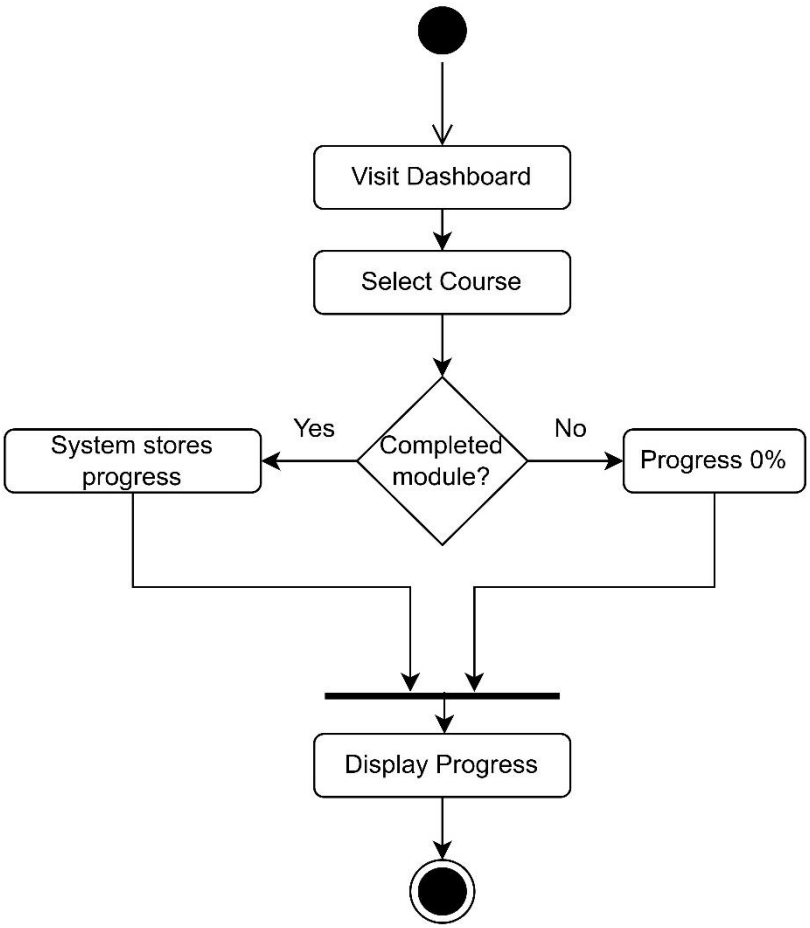


Figure 2.4.3.5: Track Progress

2.4.3.6 Activity Diagram: View Leader board

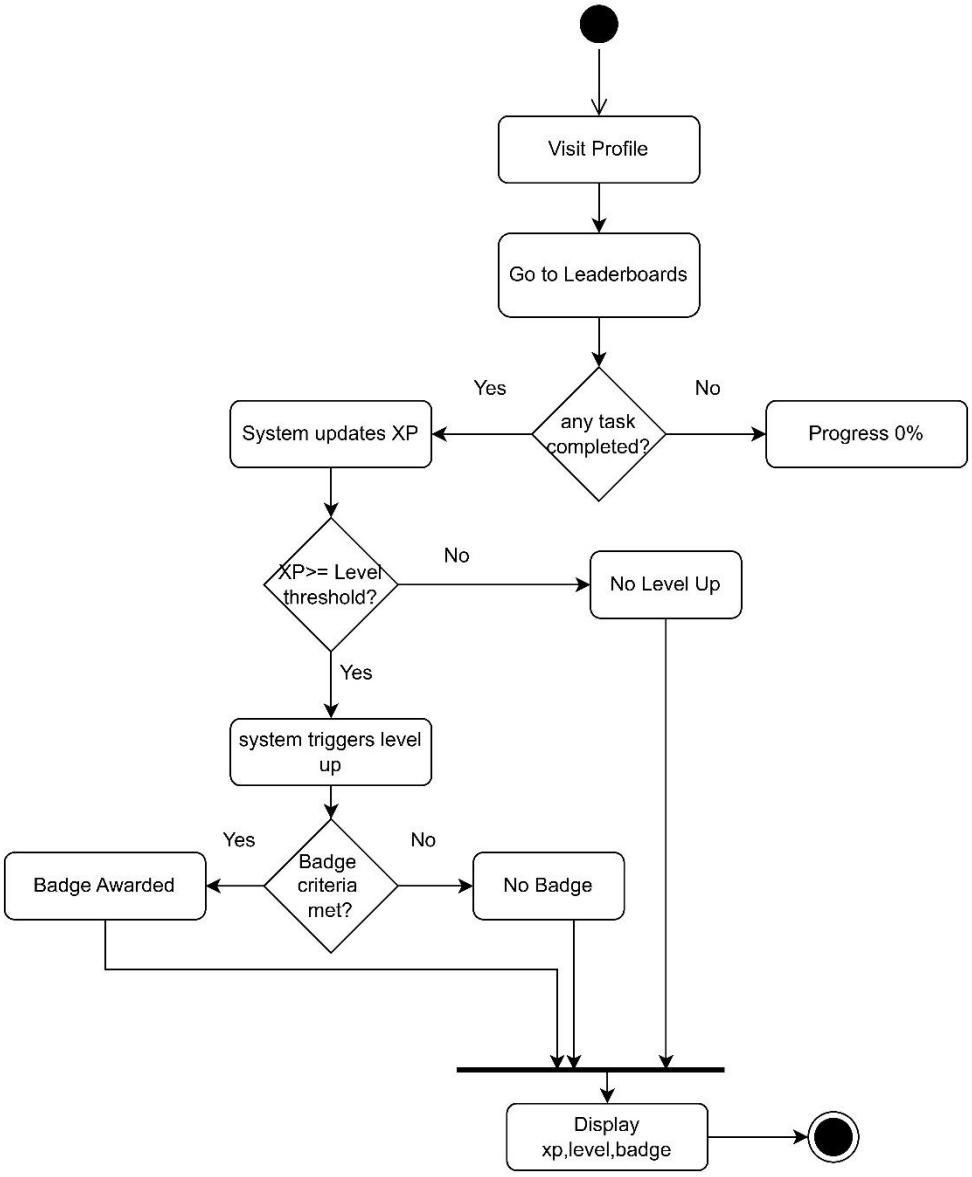


Figure 2.4.3.6: View Leaderboard

2.4.3.7 Activity Diagram: Assignment Submission

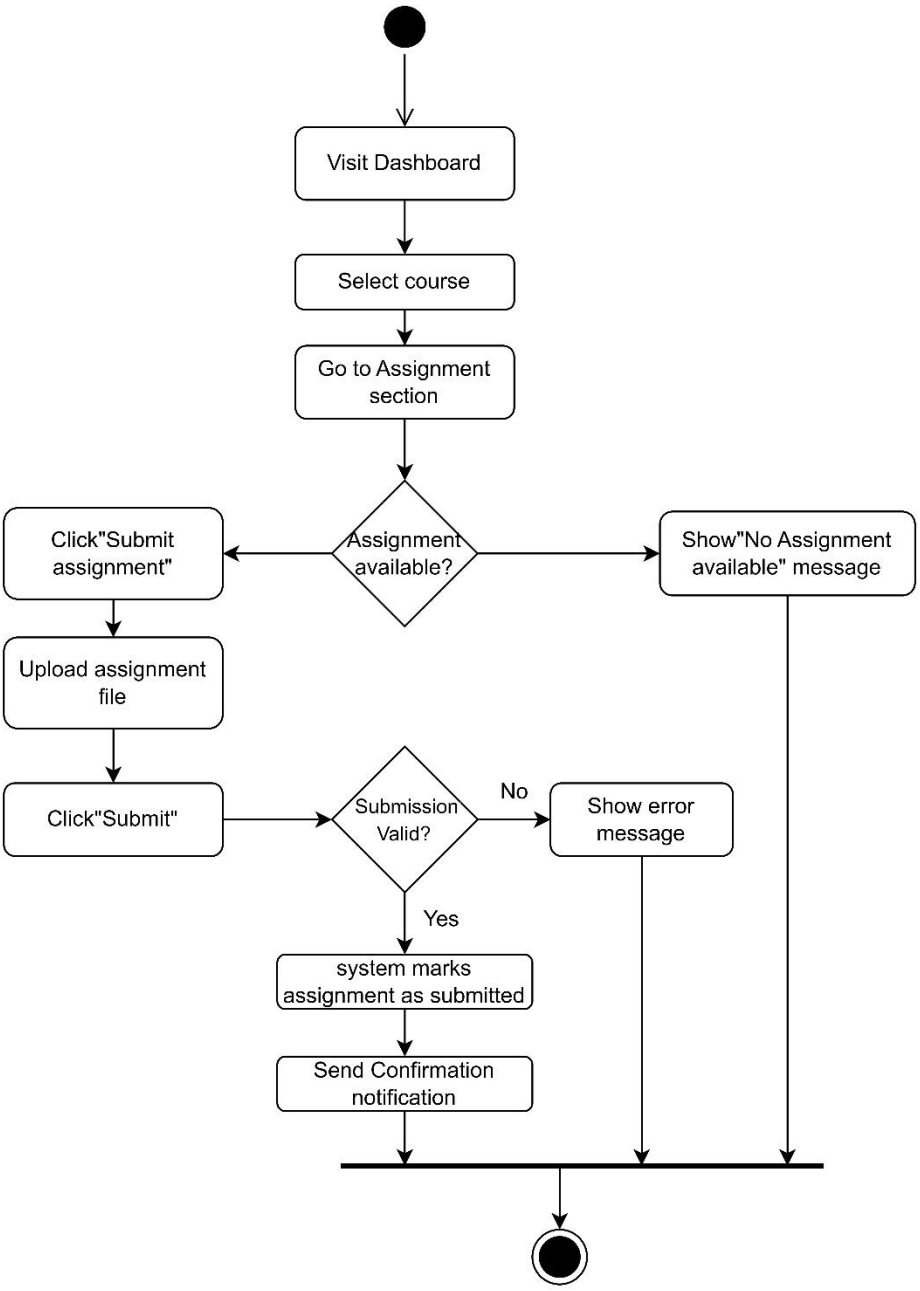


Figure 2.4.3.7: Submit Assignment

2.4.4 Sequence Diagram

2.4.4.1 Sequence Diagram: Update Profile

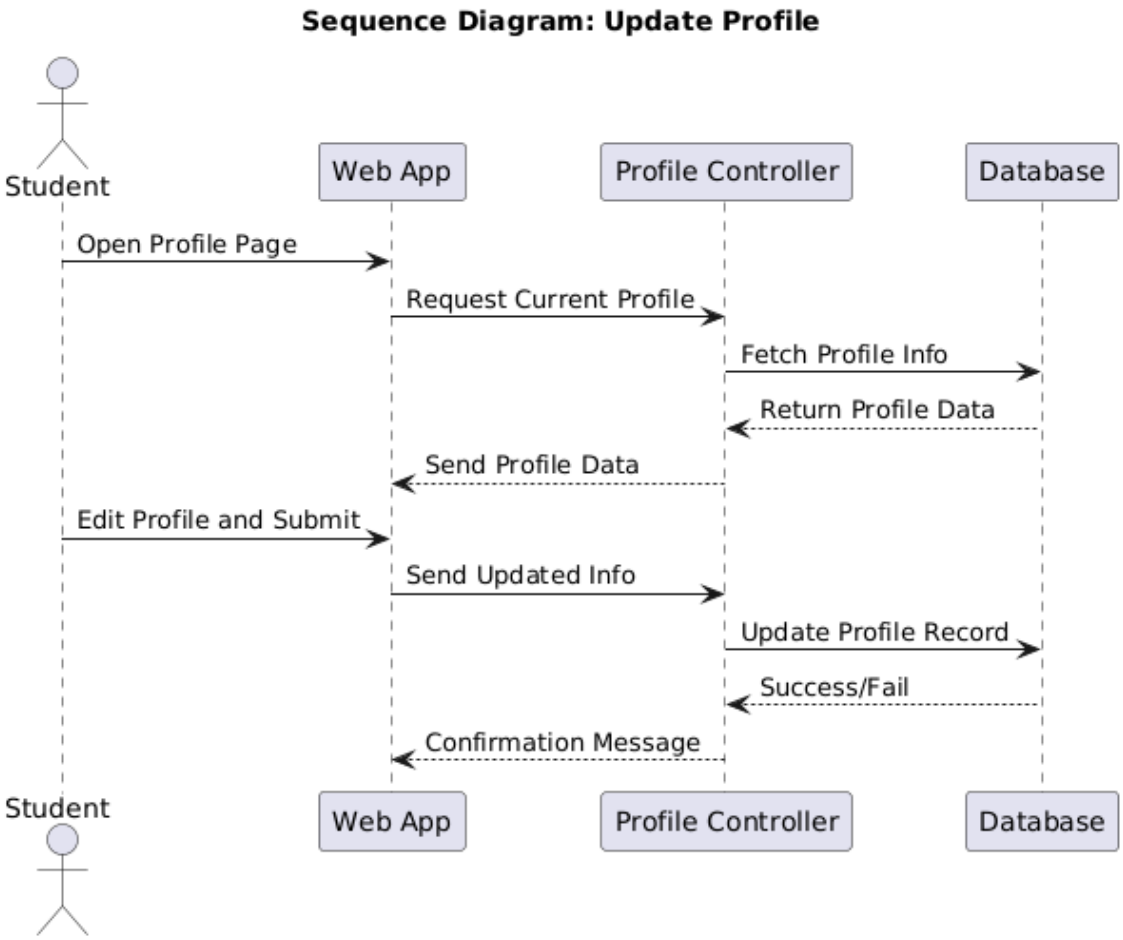


Figure 2.4.4.1: Update Profile

2.4.4.2 Sequence Diagram: Enrol in course

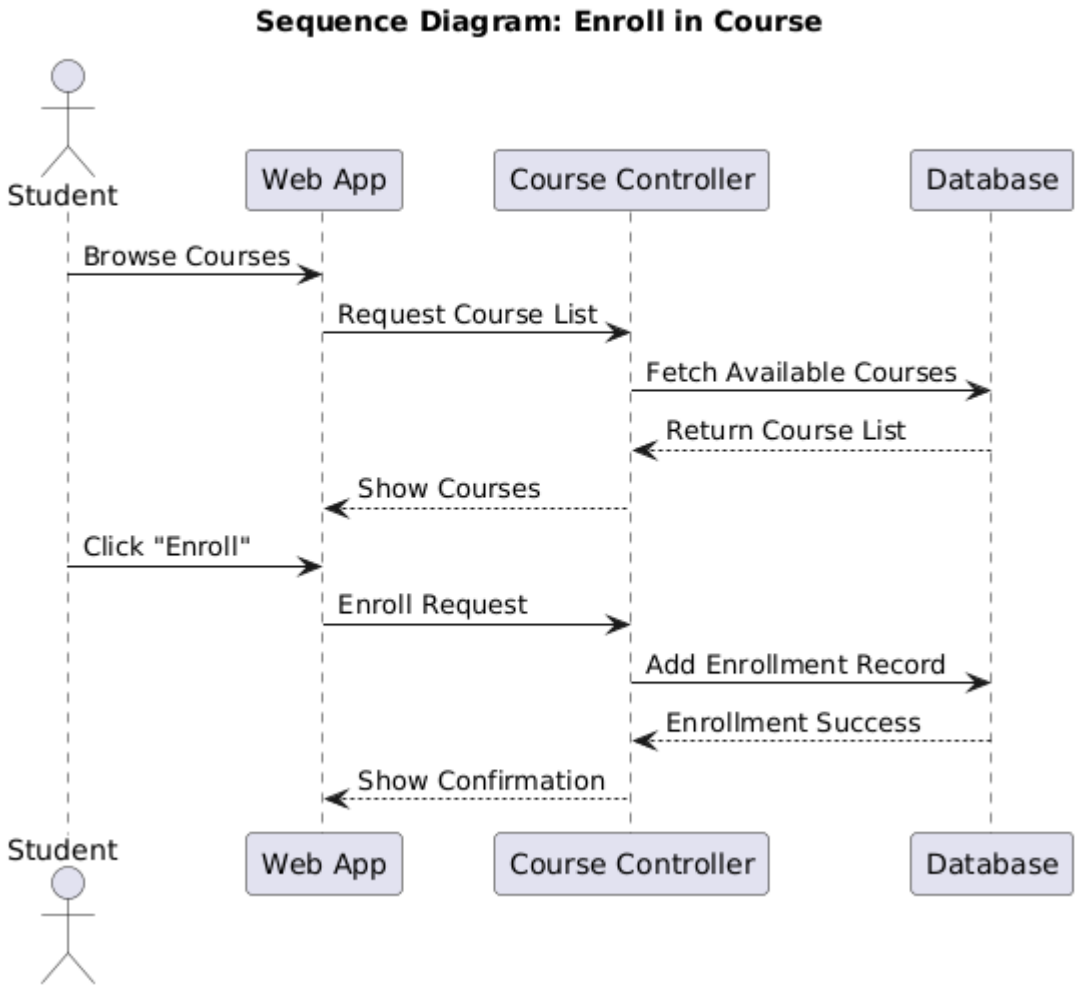


Figure 2.4.4.2: Enroll in Course

2.4.4.3 Sequence Diagram: Track Progress

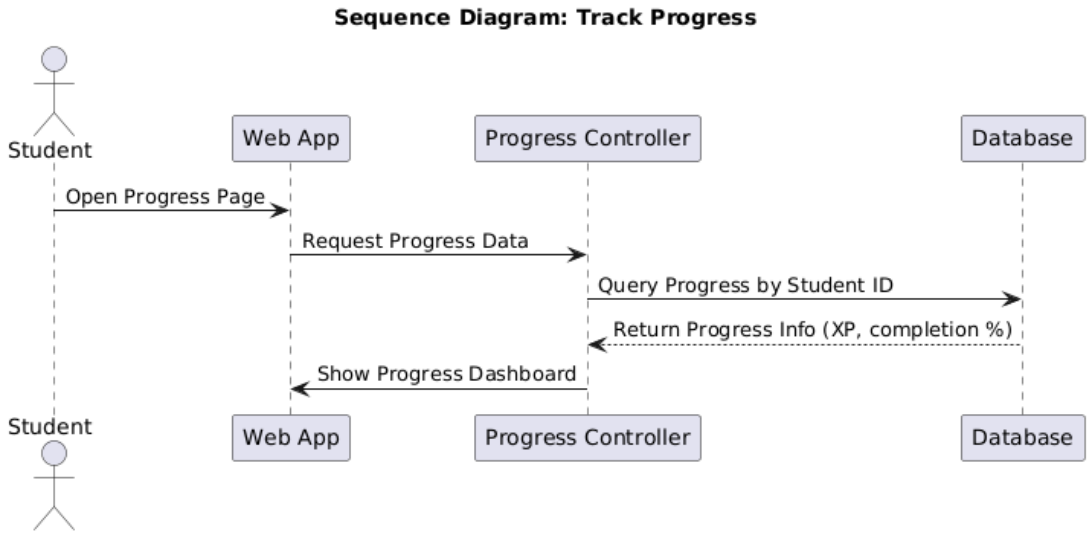


Figure 2.4.4.3: Track Progress

2.4.4.4 Sequence Diagram: Track Progress

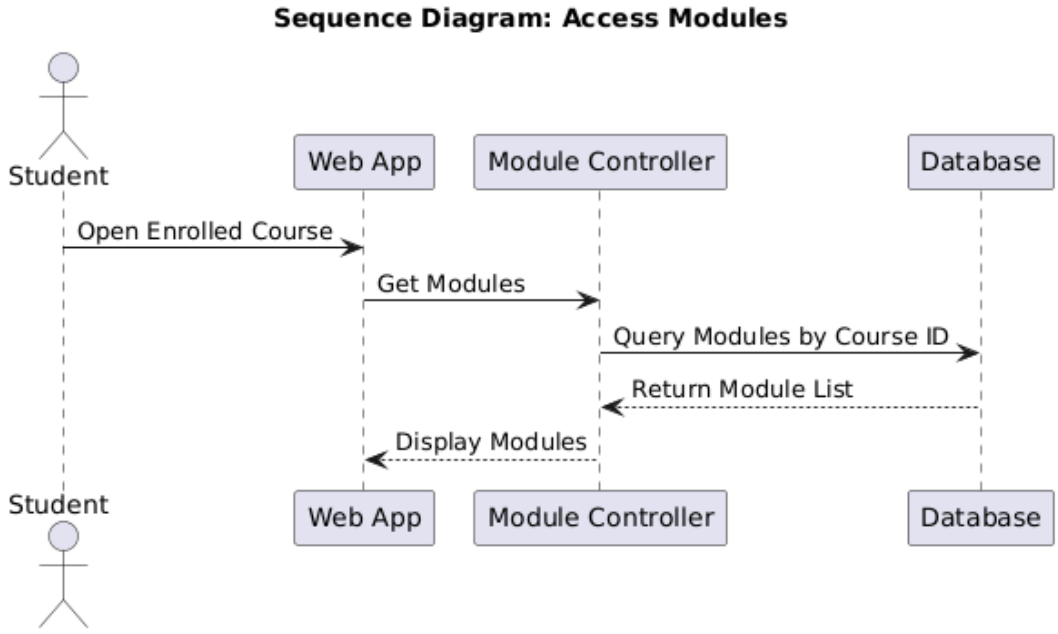


Figure 2.4.4.4: Access Modules

2.4.4.5 Sequence Diagram: Track Student Performance

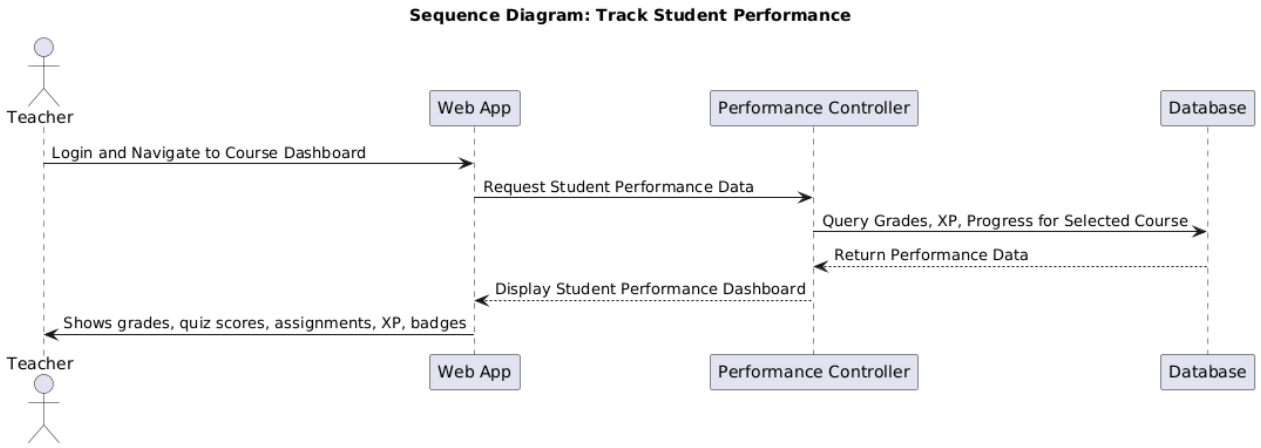


Figure 2.4.4.5: Track Student Result

2.4.4.5 Sequence Diagram: Track Student Performance

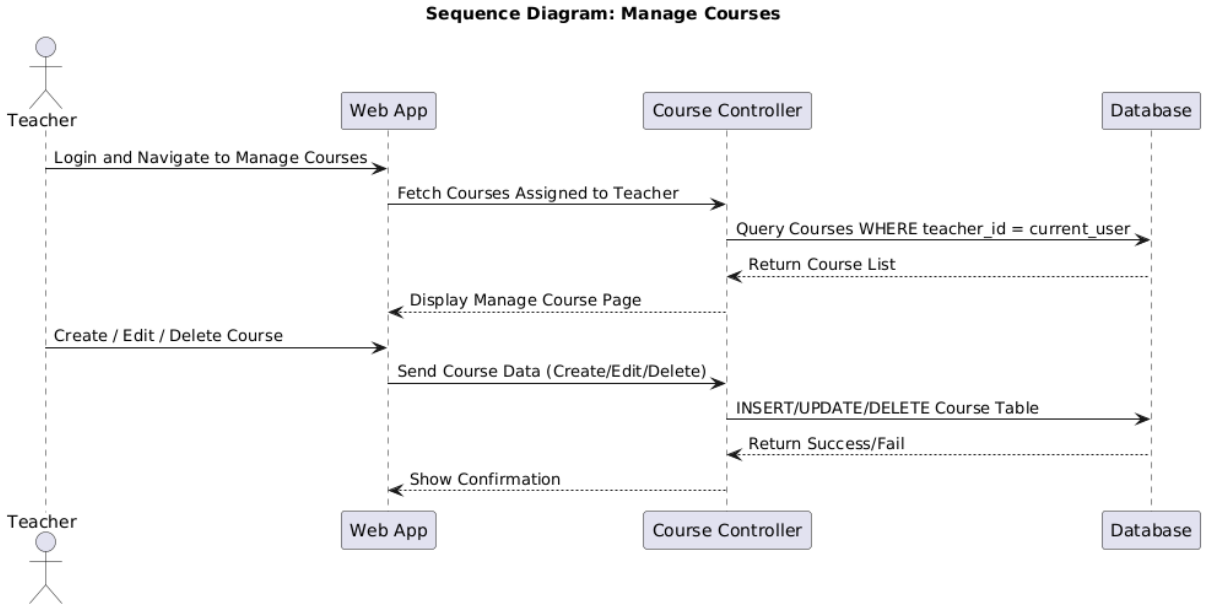


Figure 2.4.4.6: Manage Course

2.4.4.7 Sequence Diagram: Take Quizzes

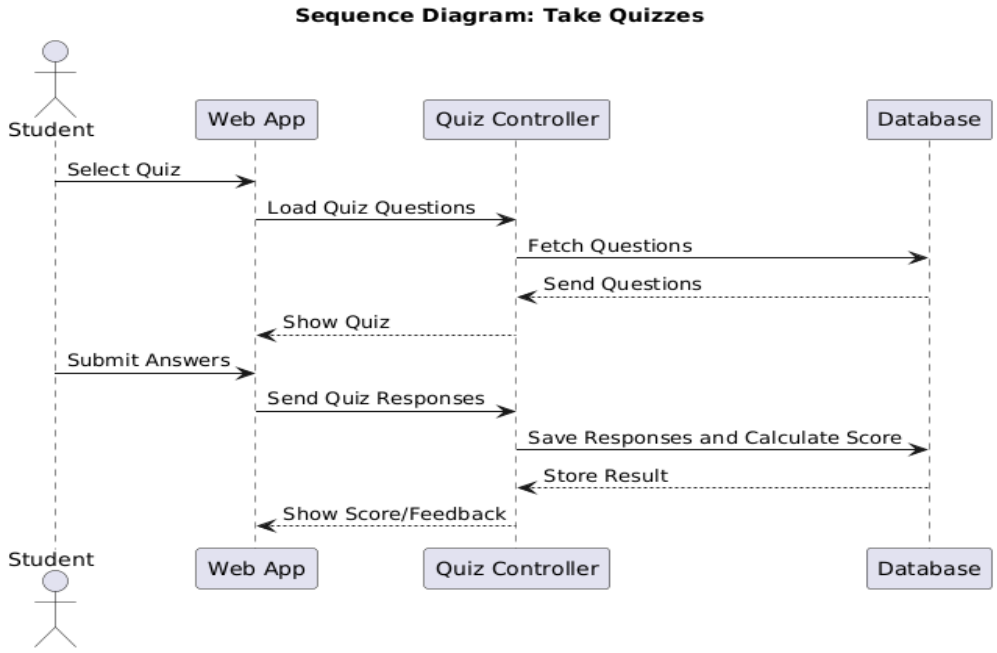


Figure 2.4.4.7: Take Quizzes

2.4.4.8 Sequence Diagram: View Leader board

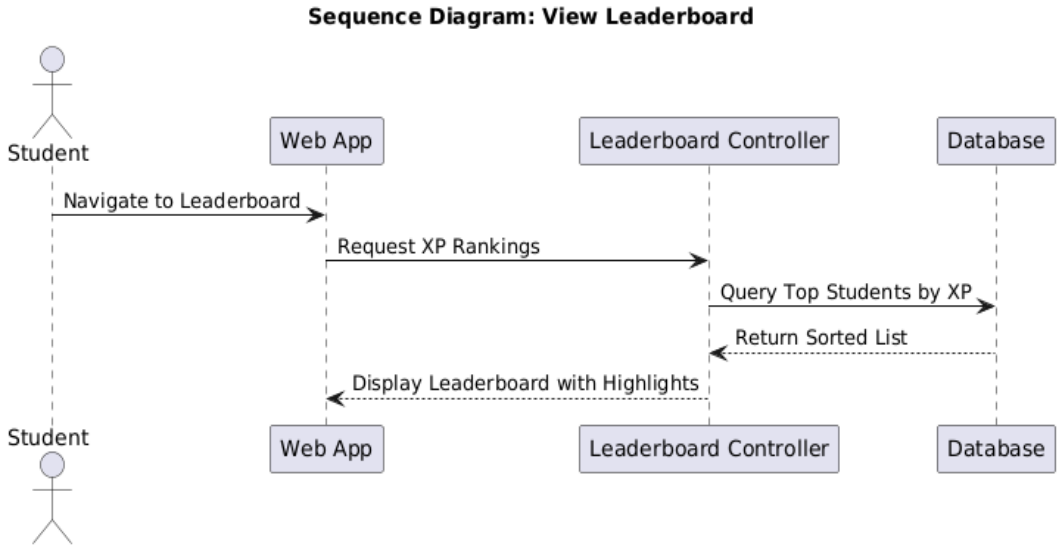


Figure 2.4.4.8: View Leaderboard

2.4.4.9 Sequence Diagram: Submit Assignment

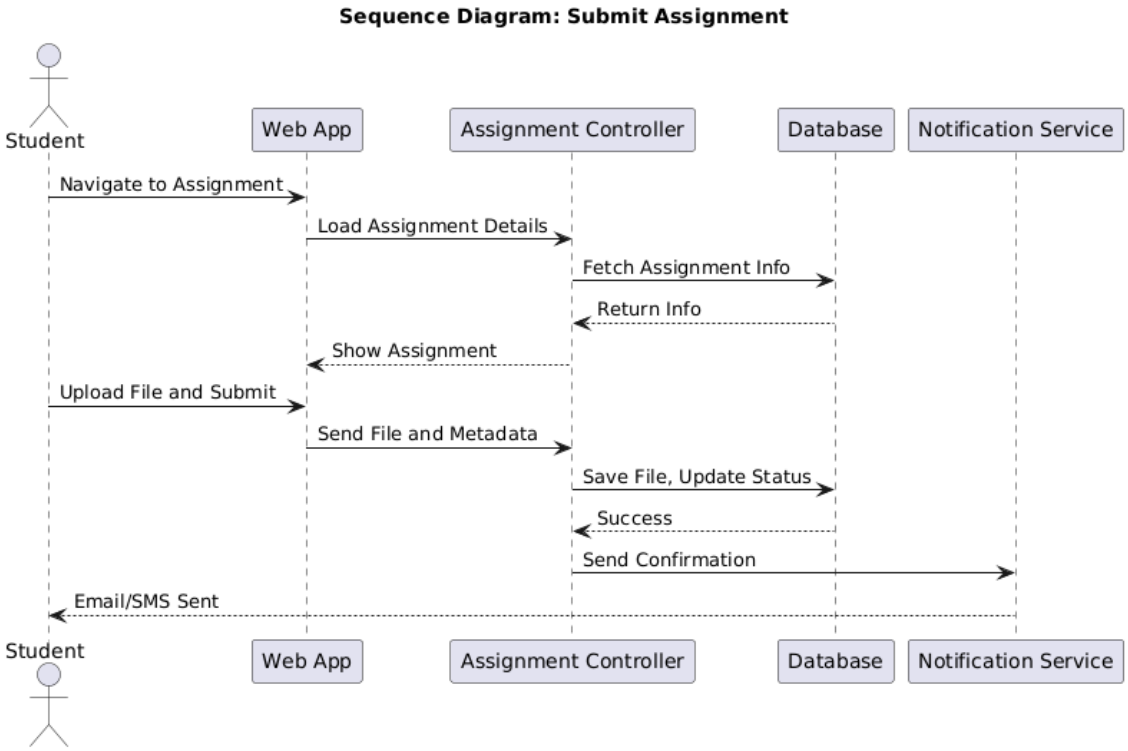


Figure 2.4.4.9: Submit Assignment

2.4.5 Class Diagram

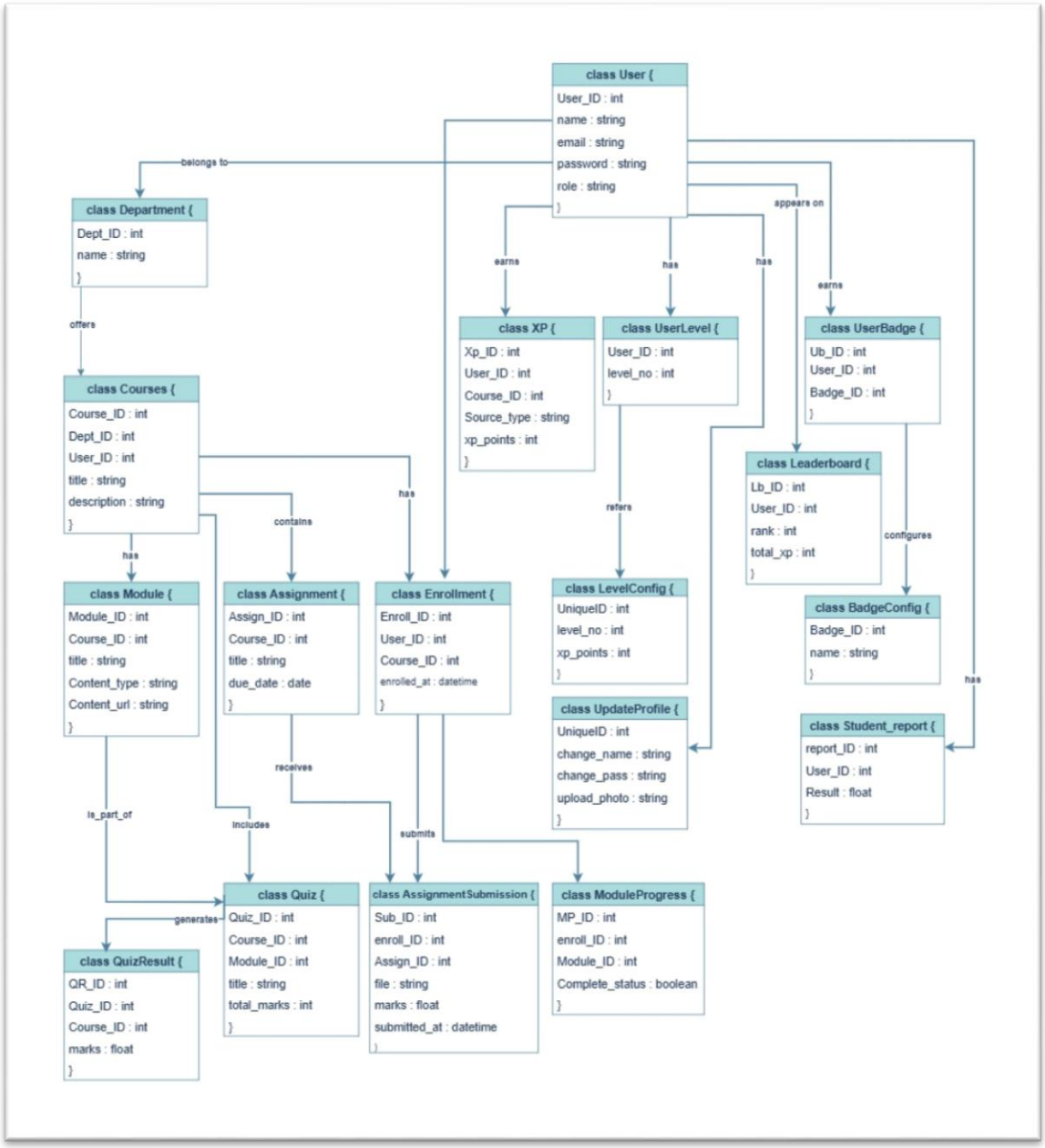


Figure 2.4.5: Class Diagram

2.4.6 Entity Relation Diagram

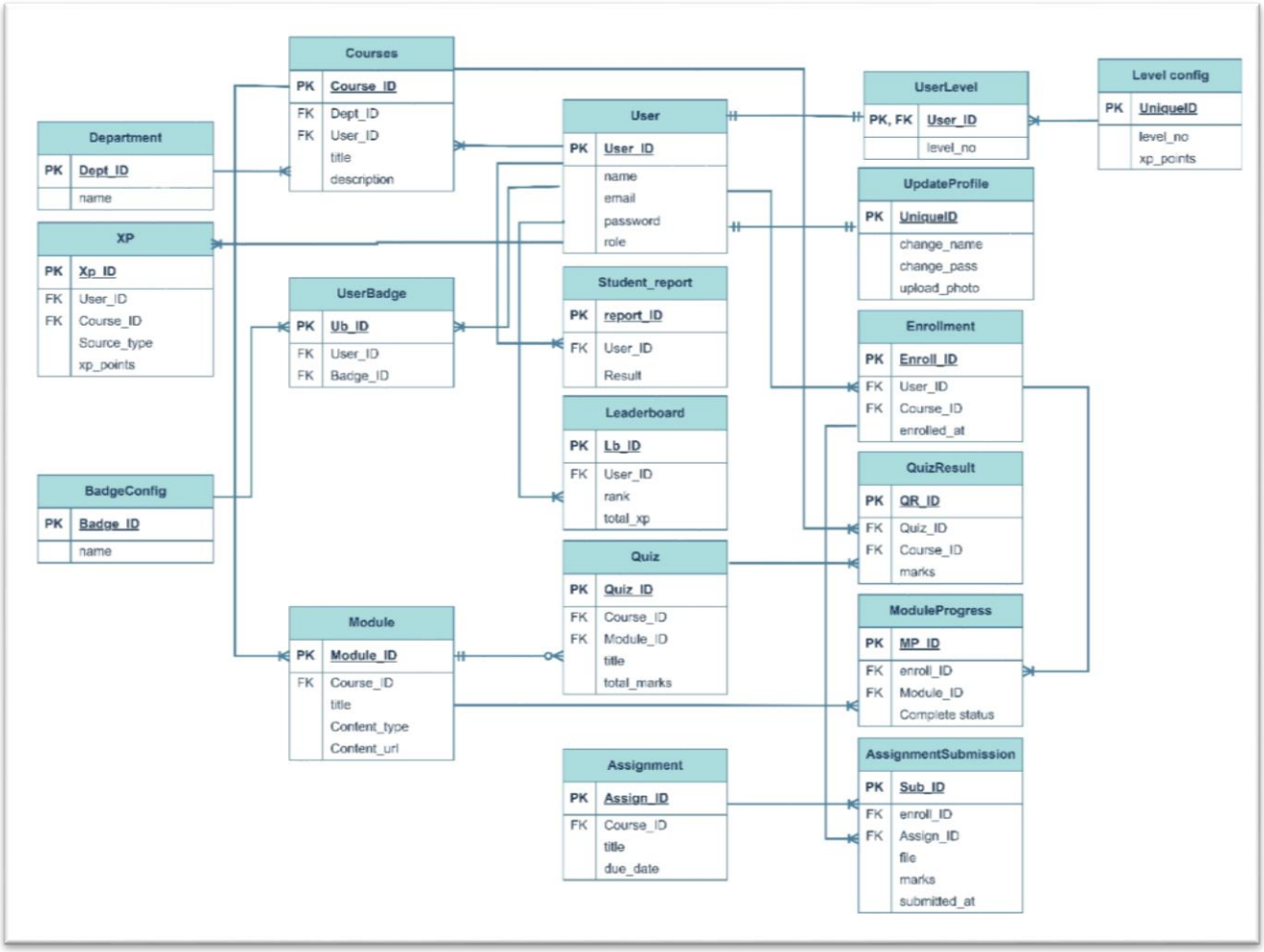
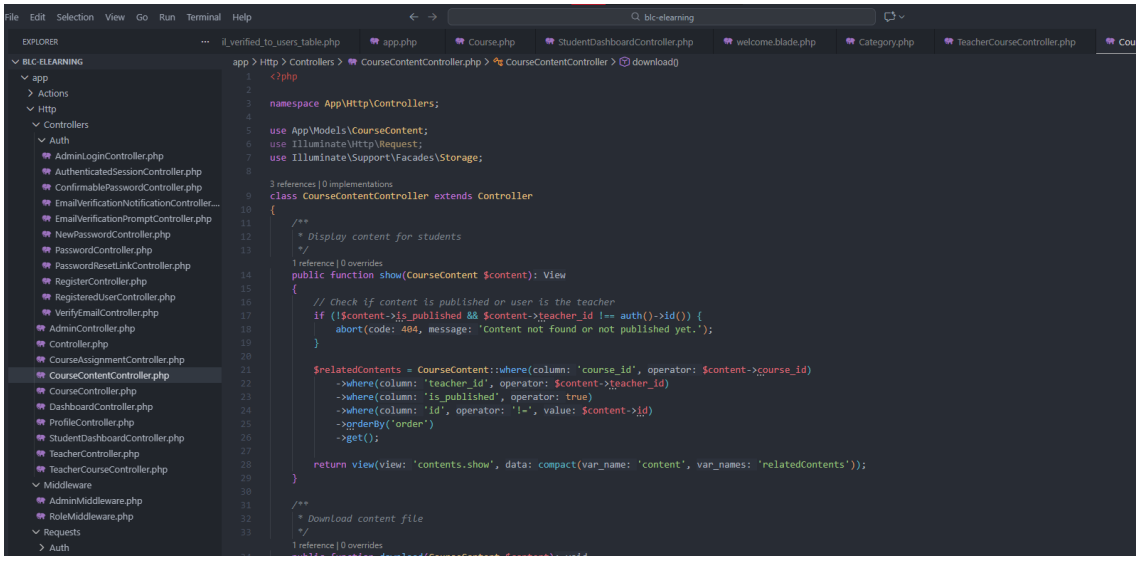
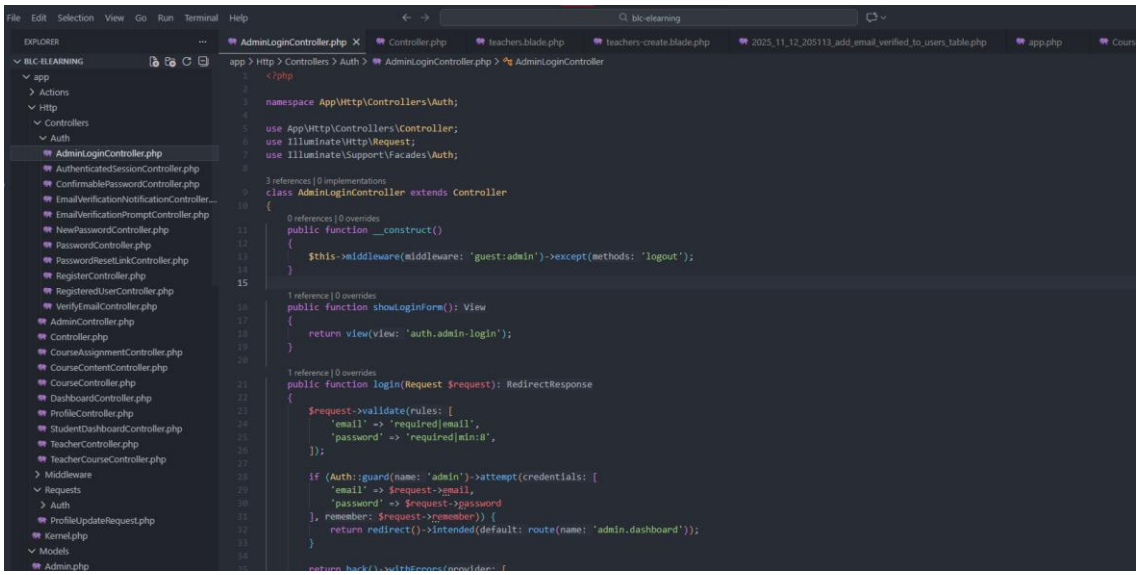


Figure 2.4.6: ER Diagram

2.5 Coding: Appendix A



```php
<?php

namespace App\Http\Controllers;

use App\Models\CourseContent;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class CourseContentController extends Controller
{
    /**
     * Display content for students
     */
    public function show(CourseContent $content): View
    {
        // Check if content is published or user is the teacher
        if (!$content->is_published && $content->teacher_id !== auth()->id()) {
            abort(code: 404, message: 'Content not found or not published yet.');
        }

        $relatedContents = CourseContent::where(column: 'course_id', operator: $content->course_id)
            ->where(column: 'teacher_id', operator: $content->teacher_id)
            ->where(column: 'is_published', operator: true)
            ->where(column: 'id', operator: '!=', value: $content->id)
            ->orderBy('order')
            ->get();

        return view(view: 'contents.show', data: compact(var_name: 'content', var_names: 'relatedContents'));
    }

    /**
     * Download content file
     */
```



```php
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AdminLoginController extends Controller
{
    public function __construct()
    {
        $this->middleware(middleware: 'guest:admin')->except(methods: 'logout');
    }

    public function showLoginForm(): View
    {
        return view(view: 'auth.admin-login');
    }

    public function login(Request $request): RedirectResponse
    {
        $request->validate(rules: [
            'email' => 'required|email',
            'password' => 'required|min:8',
        ]);

        if (Auth::guard(name: 'admin')->attempt(credentials: [
            'email' => $request->email,
            'password' => $request->password
        ], remember: $request->remember)) {
            return redirect()->intended(default: route(name: 'admin.dashboard'));
        }
```

2.6 Summary

The system is a monolithic MVC application developed in PHP Laravel and Blade templates, which provide a well-separated use of concerns and a maintainable code. The database architecture revolves around the fundamental objects such as Users, Courses and Enrollments and strong relationships to facilitate complicated educational processes. The major players are secure authentication module and role based access, responsive course management system and user friendly analytics dashboard.

Chapter 3: Software Testing

3.1 Introduction

The chapter explains the all-encompassing testing plan that is used to make the Learning Management System reliable, secure and usable. Software testing is an important stage in our lifecycle of development and it is meant to detect and solve the defects, before the system is made available to the end-users. We check through the use of systematic validation and verification that all functional requirements are satisfied and that the platform will be able to operate effectively with the anticipated loads. Our testing strategy involves a number of testing techniques which include unit testing whereby we test an individual component and integration testing whereby we test the interaction between modules and the final testing is the user acceptance testing which involves testing whether the system is up to the expected expectations of the user. This intensive process ensures that it delivers quality, robust and user friendly application.3.2 Testing Features

3.2.1 Feature to Be Tested

The core functionalities will be run through rigorous testing to enable it to perform as expected:

- User Registration – User-ID should be authenticated by creating accounts with appropriate data integrity.
- User Login – This is to provide secure authentication of students and teachers.
- Admin Login – Verifying restricted access to administrative functions.
- Admin Dashboard – Testing management has capabilities of user and course management.

3.3 Testing Strategies

3.3.1 Test Approach

Our testing strategy follows a multi-layered approach to ensure comprehensive coverage:

Unit Testing: Individual components (e.g., controllers, models in Laravel) are tested in isolation using PHPUnit to verify correct logic and output.

Integration Testing: Interactions between modules (e.g., user authentication and course enrollment) are tested to ensure seamless data flow and functionality.

System Testing: The complete integrated application is tested according to both the functional and non-functional requirements such as the load performance and security vulnerability tests.

User Acceptance Testing (UAT): The real-life situations are implemented with the chosen students and teachers to ensure that the system can satisfy their needs and is capable of going online.

3.3.2 Pass/Fail Criteria

Pass Criteria: A test case is considered as Pass when the actual test result is the same as in the test scenario, and all the functional and performance standard-tests were met without failures.

Fail Criteria: A test case passes to the Failed state when the observed result is different than the desired result, the program crashes or the performance (e.g. response time) has been exceeded. Any failure is recorded, prioritized and rectified before the next testing cycle or deploying.

3.4 System Testing

Digital E-Learning platform is subjected to thorough testing such as functional testing of key functionality (user authentication, course enrollment, progress tracking), testing the user interface and user experience on various devices between the student and teacher dashboard, role-based access control test, and performance testing when the site has maximum users. Integration testing makes sure that the flow of data between course modules, enrollment systems and calendar functions flows smoothly, and regression testing makes sure the system is stable throughout updates.

3.4.1 Test Case: 01 Registration

Test Case: TC01	**Test Case Name:** Registration
System: Digital E-learning Platform	**Subsystem:** User Authentication
Designed by: Irfan Kafil Areyan	**Designed date:** 10-10-2025
Executed by: Irfan Kafil Areyan	**Executed date:** 20-10-2025
Description: The user registration for the Digital E-learning platform to access the contents	
Pre-condition: The user accesses the registration page	

Step	Name	Email	Pass	Confirm Pass	Response	Pass/Fail	Comment
1	Irfan Kafil	Irfan.edu@gmail.com	123456	123456	Successful	Pass	Registration successful with valid info
2		Irfan.edu@gmail.com	123456	123456	Name field empty	Fail	User must input name
3	Irfan Kafil		123456	123456	Email field empty	Fail	User must input email

Post-condition: The user is successfully registered and redirected to login page

3.4.2 Test Case: 02 Login

Test Case: TC02	**Test Case Name:** Login
System: Digital E-learning Platform	**Subsystem:** User Authentication
Designed by: Irfan Kafil Areyan	**Designed date:** 10-10-2025
Executed by: Irfan Kafil Areyan	**Executed date:** 20-10-2025
Description: Verify that the registered users can successfully login	
Pre-condition: The user accesses the login page	

Step	Email	Pass	Response	Pass/Fail	Comment
1	Irfan.edu@gmail.com	123456	Successful	Pass	Valid credentials accepted
2	Irfan.edu@gmail.com	123465	Invalid pass	Name field empty	System rejects wrong password
3	Irfan123@gmail.com	123456	User not found	Email field empty	Email not registered
4		123456	Email required	Password not matched	System detects empty email fields

Post-condition: The user is redirected to dashboard after successful login

3.4.3 Test Case: 03 Admin login

Test Case: TC03			**Test Case Name:**Admin login		
System: Digital E-learning Platform			**Subsystem:** User Authentication		
Designed by: Irfan Kafil Areyan			**Designed date:** 10-10-2025		
Executed by: Irfan Kafil Areyan			**Executed date**: 20-10-2025		
Description: Verify that admin can successfully login to the admin panel					
Pre-condition: The user accesses the admin login page					
Step	Name	Pass	Response	Pass/Fail	Comment
1	admin	admin123	Successful	Pass	Valid admin credentials accepted
2	admin	123456	Invalid Pass	Fail	System rejects wrong admin password
3	Sabbir	admin123	Access denied	Fail	System rejects invalid admin name
4		admin123	Username required	Fail	System validates empty name fields
Post-condition: Admin is redirected to admin dashboard upon successful login					

3.4.4 Test Case: 04 Admin dashboard

Test Case: TC04			Test Case Name: Admin dashboard		
System: Digital E-learning Platform			Subsystem: User Authentication		
Designed by: Irfan Kafil Areyan			Designed date: 10-10-2025		
Executed by: Irfan Kafil Areyan			Executed date: 20-10-2025		
Description: The user registration for the Digital E-learning platform to access the contents					
Pre-condition: The user accesses the registration page					
Step	Action	Expected result	Actual result	Pass/Fail	Comment
1	Access dashboard	Dashboard loads with statistics	Dashboard loaded successfully	Pass	All widgets displayed correctly
2	Click ”Manama users”	User management page opens	User list displayed	Pass	User management accessible
3	Search for “Irfan”	User appears in search result	User found and displayed	Pass	Search functionality works
4	Click “course management”	Course management page opens	Course list displayed	Pass	Course management accessible
Post-condition: Admin can perform all dashboard operations successfully					

3.5 Summary

The enables the Learning Management System to be reliable, secure, and perform well, which was achieved through a comprehensive testing strategy. A multi-layered method was used in the testing phase, and this consisted of unit testing, integration testing, system testing, and user acceptance testing (UAT).

Key Testing Outcomes:

- All the main features such as user registration, authentication, course management, and assessment systems were successfully tested.
- Security testing E. Security testing established a strong level of protection against popular vulnerabilities such as SQL injection and XSS attack.
- Performance test confirmed the ability to support 1,000 + simultaneous users and 3 seconds response times.
- Both the student and teacher groups gave positive feedback to user acceptance testing.

Chapter 4: Deployment and Maintenance

4.1 Introduction

This chapter outlines the key stage of transferring the developed Learning Management System to a testing environment to a live production environment, and the plan of how it will be maintained going forward. It discusses the deployment architecture, Release process step-by-step release process after the Software Release Life Cycle (SRLC) and the post-launch support plan. The emphasis is placed on providing a seamless, safe, and stable implementation of the end-users (students and teachers) and afterwards, the controlled monitoring, changes, and troubleshooting to ensure the long-term sustainability and functionality of the platform.

4.2 Try to follow the SRLC (software release life cycle)

The implementation will be based on a systematic SRLC in order to reduce risk and make the launch successful.

Phase 1: Planning & Preparation for Pre-release

- Goal: Complete all the pre-requisites to a stable deployment.
- Environment Setup: Configure the production server (LAMP/LEMP stack: Linux, Apache/Nginx, MySQL, PHP) with optimized settings for security and performance.
- Data Migration: Request and certify user and course data (where applicable) migration scripts to transfer any existing data out of any legacy system.
- Backup Plan: Develop a complete backup and roll back process to be able to recover the system in event of critical failure.
- Documentation: Complete documentation of students and teachers, and administrative documentation of IT staff.

Phase 2: Release Building & Testing

- Objective: Assemble and validate the final release candidate.
- Build Generation: Build the final deploy roll out package of the version controlled codebase.
- Smoke Testing: Execute a quick set of tests on the production-like environment to ensure basic functionality.
- User Acceptance Testing (UAT): A limited number of real students and teachers are used to test the platform in a staging environment to prove or refute the real-world workflows.

Phase 3: Deployment & Launch

- Goal: bring the system to life and make it available to all users.
- Phased Rollout (Suggested): Initially roll out to an small, controlled group of users (e.g., to a single department) and monitor the performance and get early feedback before rolling out to a large-scale user base. At this point, the database will be live, and the data will have been migrated to it using the data migration scripts.
- DNS & Server set-up: DNS Set the domain/subdomain of the university to the new production server.
- Go-Live: Open access to all target users.

Phase 4: Post-release Maintenance & Support

- Objective: Maintain the health of the system and improvement
- Monitoring: Keep an eye on the health of server, application logs, and the rate of errors.
- Patch Management: Implement security patches on a regular basis on the OS, web server, and Laravel framework.
- Bug Fixing: Correct any bugs raised by the users via a specific ticketing system. Critical fixes are deployed through hotfixes where minor ones are aggregated to be updated later.
- Feature Updates: Design and implement new features, which are published in the following releases in the same fashion as in SRLC.
- Technical Support: Have a special avenue (e.g. email, portal) where users can communicate to and receive support.
-

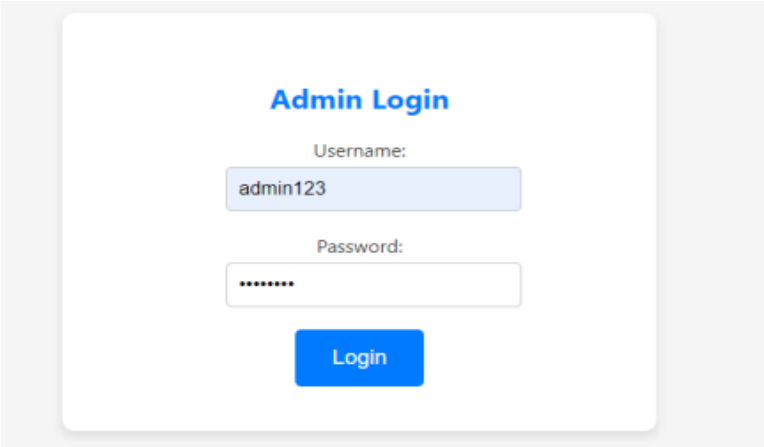
Chapter 5: User Manual

5.1 Introduction

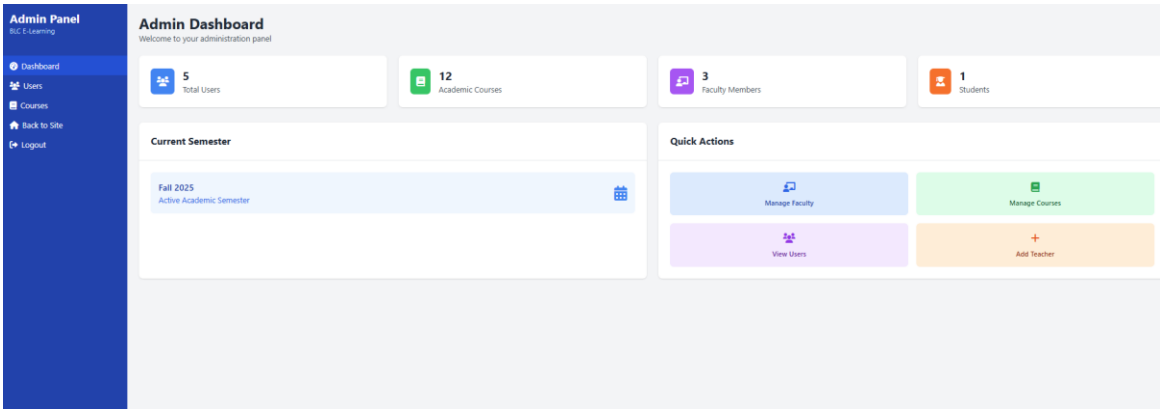
This user manual is the ultimate guide that will help you to navigate and get the full out of our new Learning Management System. This guide is meant to guide you throughout your process, be it as a student who wishes to get into the courses, track how you are doing or the teacher who is handling the classes and wants to keep track of the performance of the students.

5.2 Project Functionalities

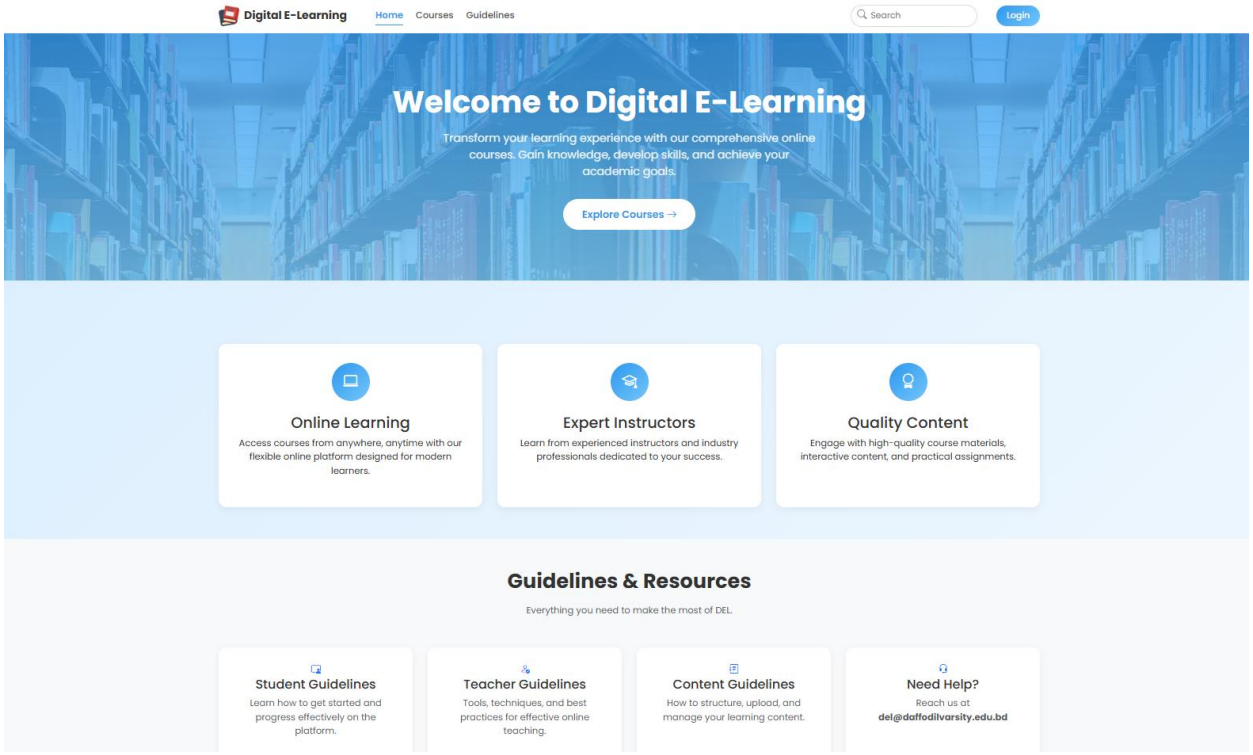
Admin Login:



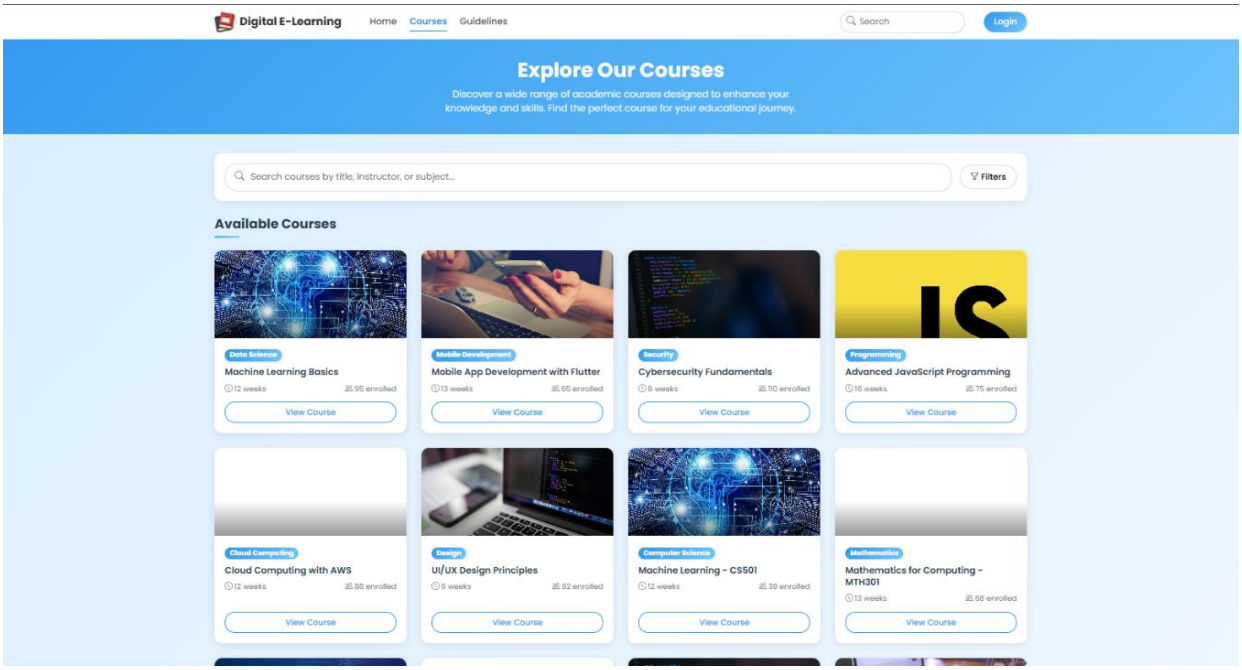
Admin Dashboard:



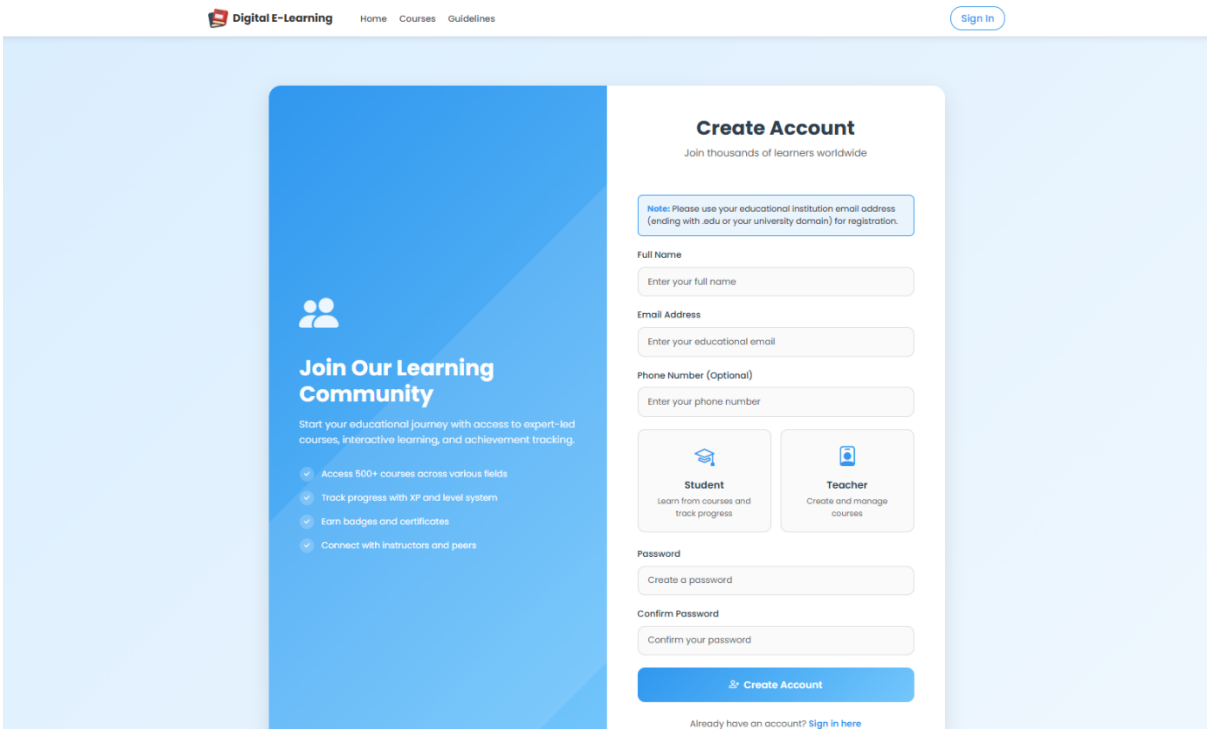
Home page



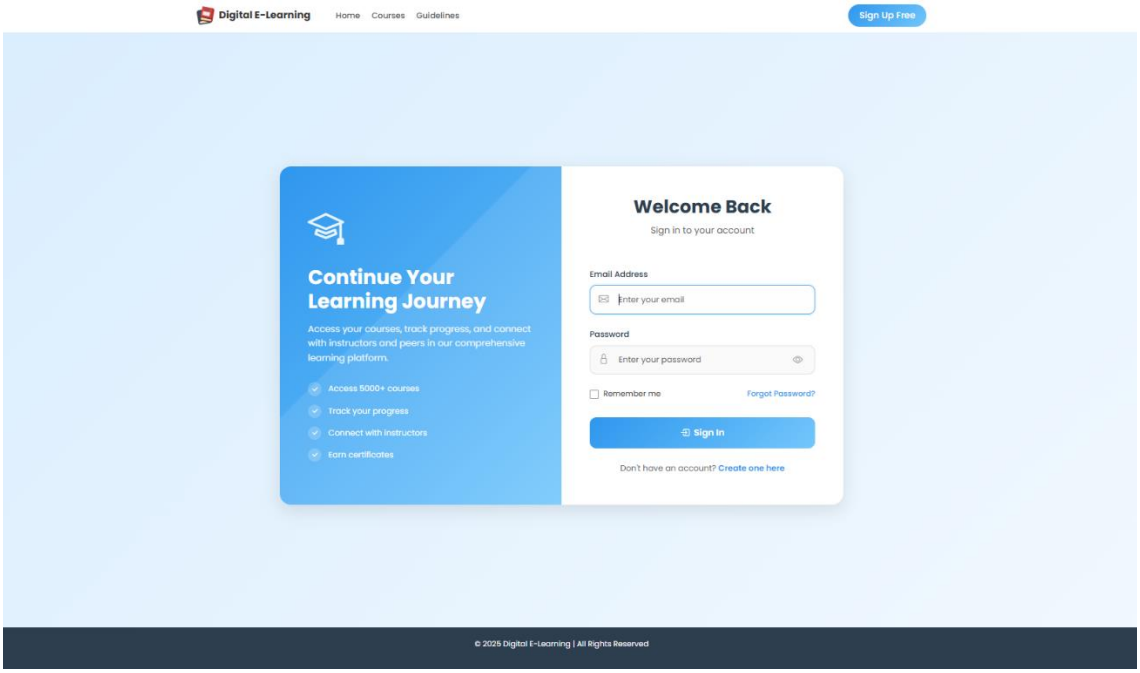
Course List



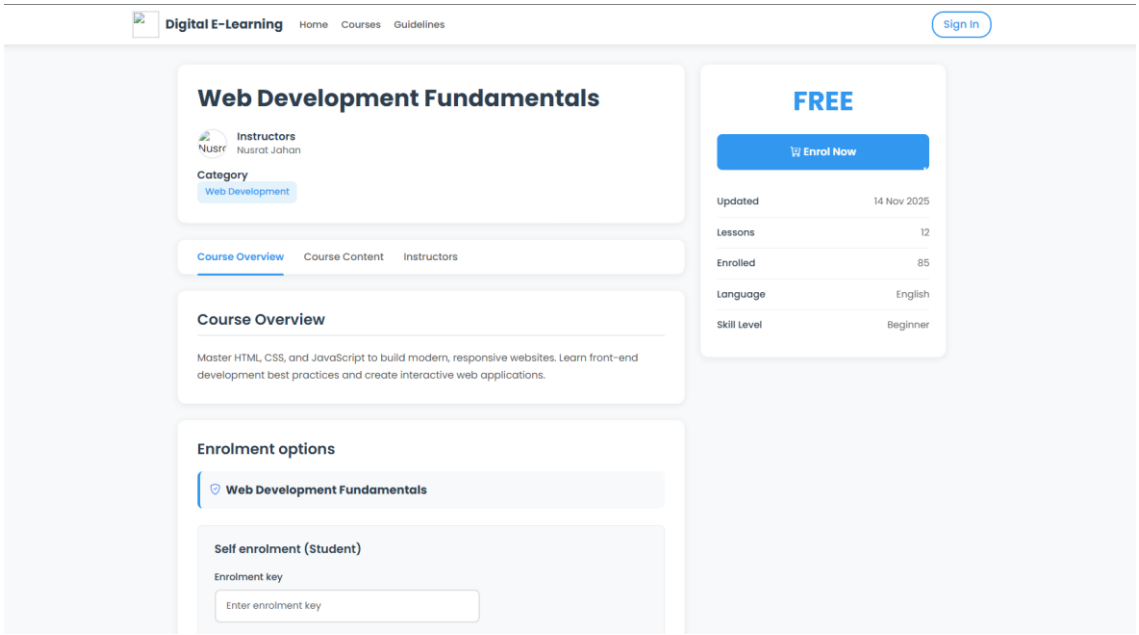
Registration



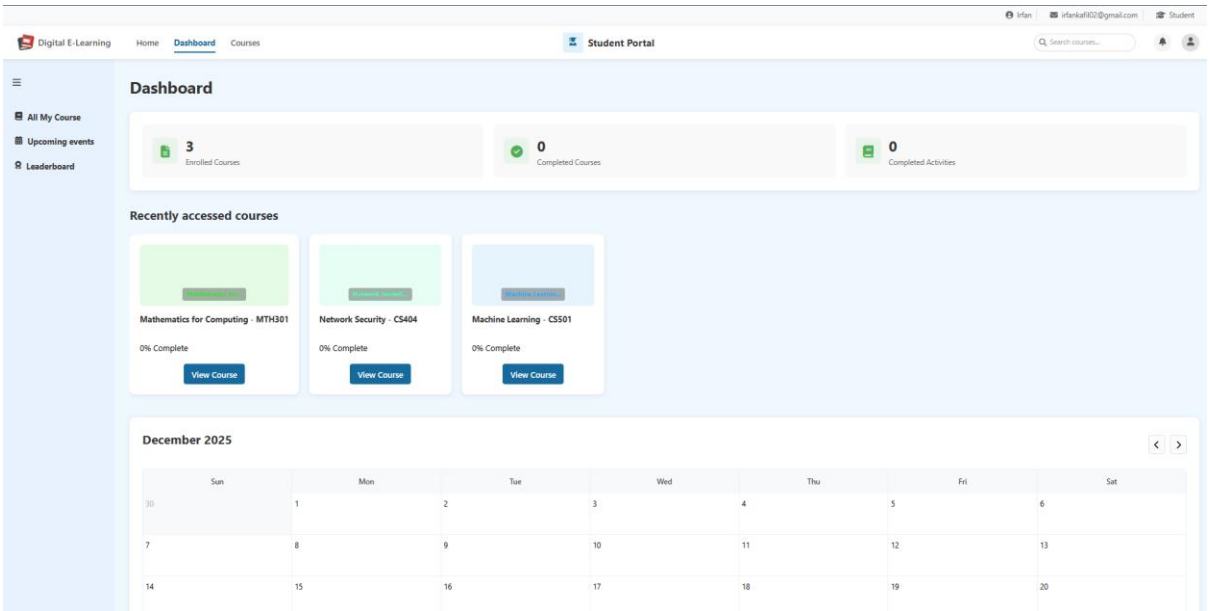
Login



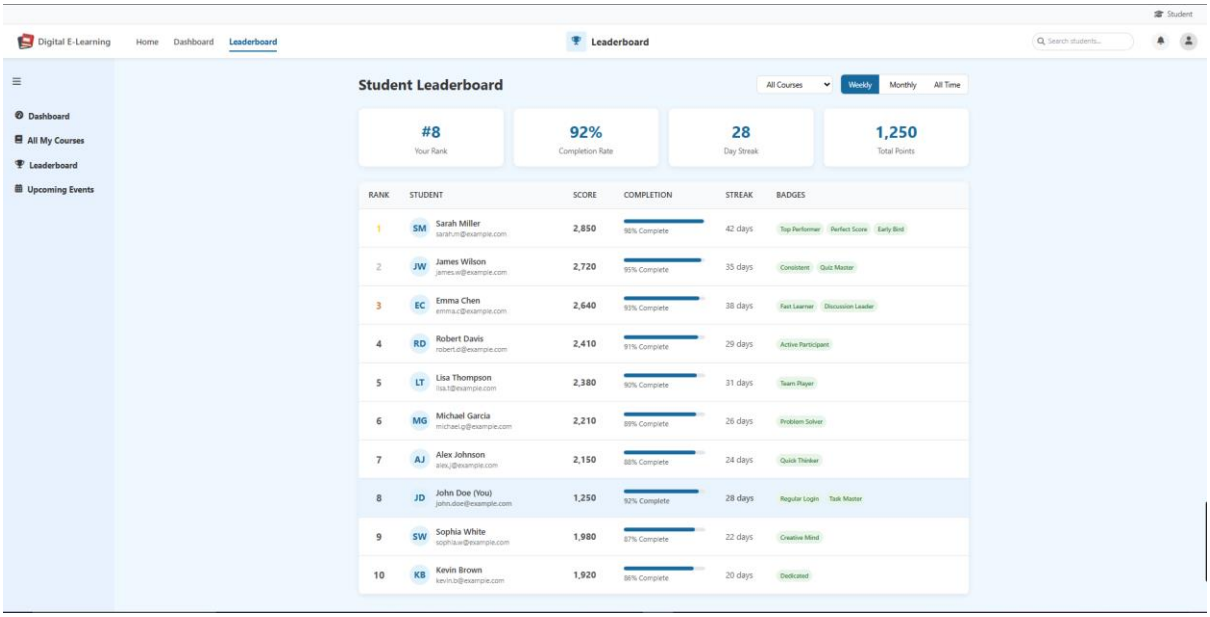
Single Course View



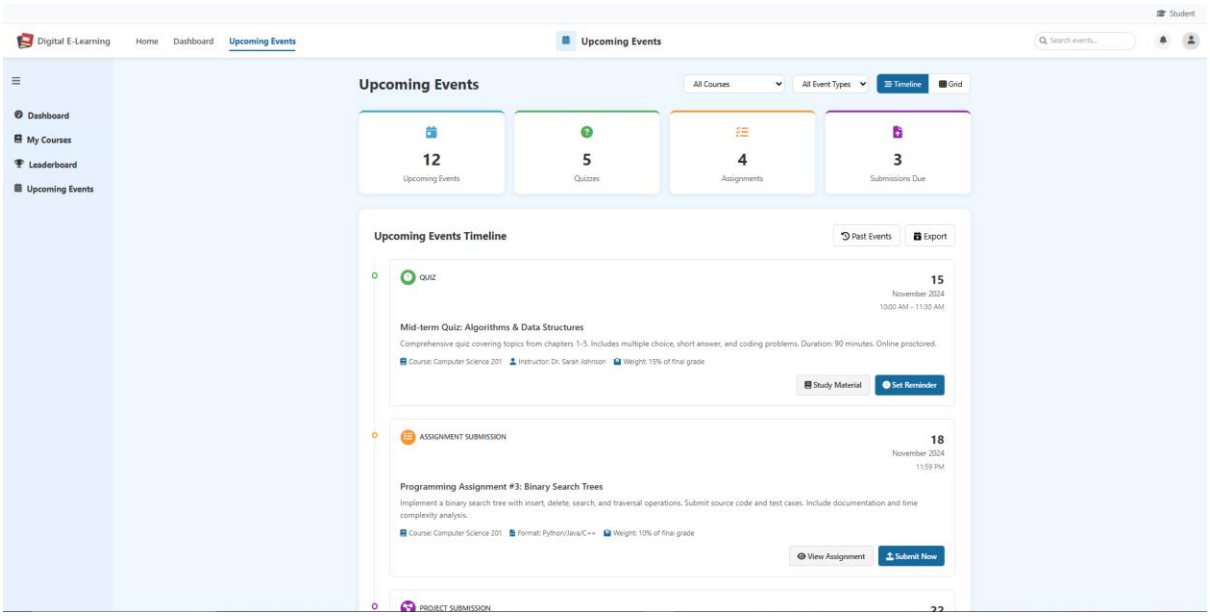
Student Dashboard



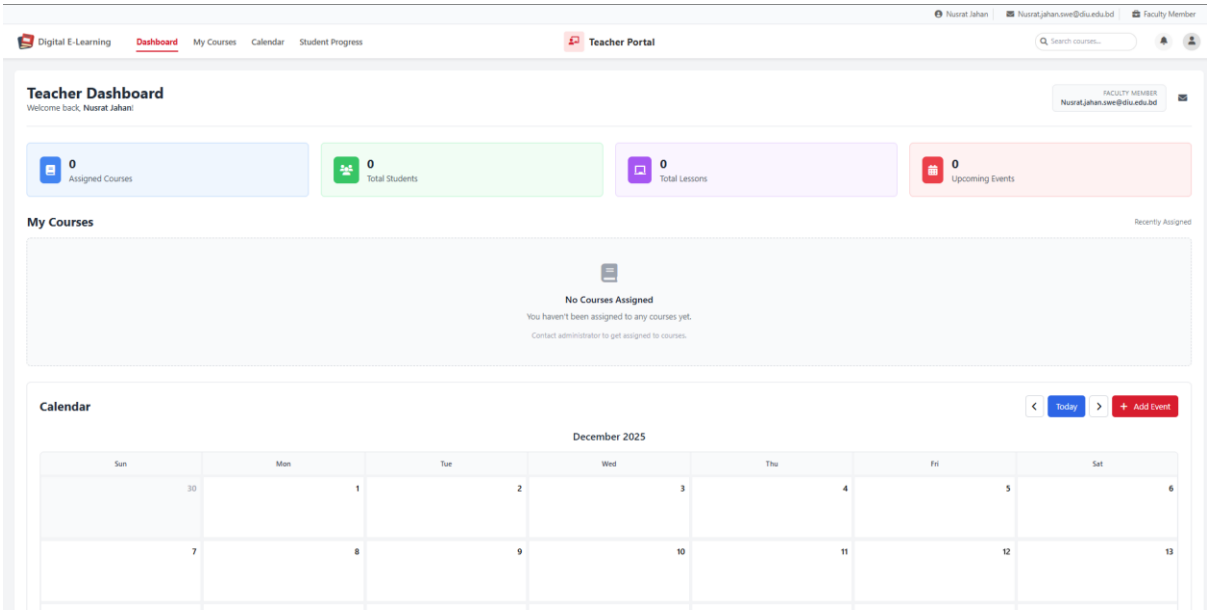
Student Leaderboard



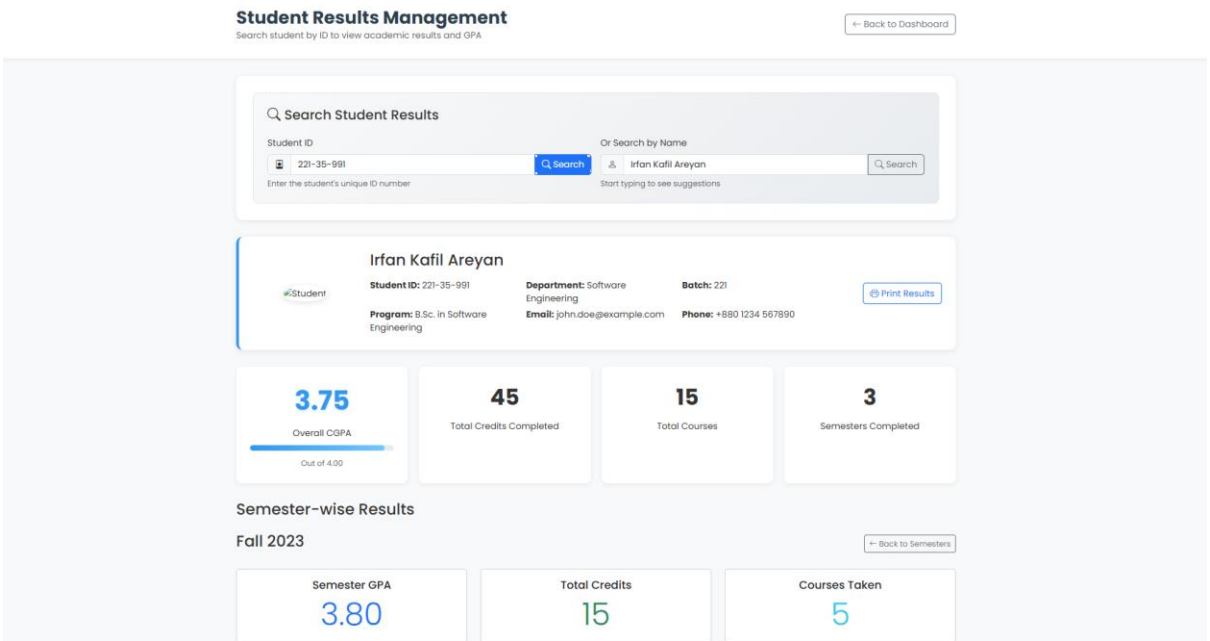
Upcoming Events:



Teacher Dashboard:



Student Result Track:



5.3 Summary

Getting Started Guide:

- Students: Fast enrolment, enrolment to courses, and dashboard information.
- Teachers: account creation, course building tools and management of students.
- Significant Characteristics: user-friendly interface and access to all learning resources based on roles.

Key Features: Intuitive interface with role-based access to all learning materials

Main Functions:

- Students: browsing of the courses, viewing of modules, assigning assignments, tracking progress, viewing grades.
- Teacher: student monitoring and performance, grading, student analytics, course management.
- Admins: content management, system analytics, user management.

Chapter 6: Project Summary

6.1 Introduction

This chapter offers an in-depth description of the Learning Management System project, its success, weaknesses, and future development prospects. It retrospectively looks into how the project was conceived and born, critical deliverables of the project, and the value the project has offered to the users.

6.2 Project Limitation

There were a number of restrictions that affected the scope and implementation of the project:

Time Limitations: Since the development time was 6 months, it did not allow incorporating more sophisticated features like AI-based analytics or a mobile app.

Budget Constraints: Limitations on resources meant that there was not much integration of the best third-party tools or plenty of custom graphics.

Technical Boundaries: The framework of Laravel and the use of relational database came with some structural constraints that did not permit the use of other emerging technologies.

Unmet Requirements: The features such as real time chat, high-quality video conferencing and integration with payment gateway were left to be addressed later on.

6.3 Scope

The project covers the following core modules and functionalities:

Included Features:

- Role based access control and user authentication.
- Course management, enrolment and delivering the content.
- Submission of assignments, management of quizzes, grading.
- Progress tracking and analytics for teachers and students.

Excluded Features:

- Mobile application development.
- Integration with payment gateway.
- High-tech AI-driven suggestions.

6.4 Future Work

To make the platform even more advanced, it is proposed to improve it in the following ways:

- Additional Features: Added live collaboration tools and gamification and a specialized mobile application.
- Technical Upgrades: AI adoption in individual learning journey and predictive analytics.
- Scalability: Use of micro services architecture to enable work with more users and other functions.
- User Experience: Improved UI/UX and features of the course and dash card customization.

6.5 Conclusion

The project has managed to provide a strong yet easy to use Learning Management System that achieves its fundamental goals. Some of the significant accomplishments have been a smooth interface between students and teachers, secure and scalable architecture and thorough testing that guarantees reliability. Some of the lessons learnt during the project like the significance of testing and feedback by the stakeholders have helped to offer invaluable insights in future activities. The platform provides a solid base of digital education and it has the prospect of expanding and changing with the new demands in the educational sector.

References

References

DIU Blended Learning Center (https://elearn.daffodilvarsity.edu.bd)

Laravel Authentication (https://www.cloudways.com/blog/setup-laravel-login-authentication)

Laravel CRUD operation (https://www.cloudways.com/blog/laravel-crud)