**Project Title: Learning Management System**

Group Members: 曾麗紅(08)

劉綺琪(12)

李敏豐(14)

李俊烽(23)

**1. Requirement Document**

An effective Learning Management System is crucial for teachers to efficiently organize and manage student records, enabling them to provide a high-quality educational experience. The proposed system aims to streamline various administrative tasks and communication channels between teachers and students.

**1.1 Introduction**

1.1.1 Purpose

This document outlines the software requirements for a comprehensive Learning Management System that empowers teachers to effectively manage student records, track academic progress, and facilitate communication with students and their families.

1.1.2 Scope

The system will provide teachers with a centralized platform to manage enrolled students, attendance tracking, academic performance monitoring, and communication, ensuring a seamless and organized learning environment.

1.1.3 Audience

This document is intended for the development team, project stakeholders, and teachers who will be the primary users of the Learning Management System.

**1.2 Glossary**

1.2.1 Student Records: The management of student information, including personal details, enrolled courses, and academic history.

1.2.2 Attendance Tracking: The process of monitoring and recording student attendance, including absences and tardiness.

1.2.3 Academic Performance: The tracking and management of student academic progress, such as assignment marks, test grades, and course completion.

1.2.4 Communication: Facilitating efficient communication channels between teachers and students, as well as with parents or guardians.

1.2.5 User Management: The process of registering, verifying, and configuring user accounts (teachers, and students) and assigning appropriate permissions and access levels.

1.2.6 Reporting: The generation of reports and analytics to provide insights into student performance and facilitate data-driven decision-making.

1.2.7 Accessibility: Ensuring the system is user-friendly and accessible for teachers, students/ parents, with support for various devices and platforms.

1.2.8 Security: Implementing measures to safeguard user data, ensure confidentiality, and comply with data protection regulations.

1.2.9 Scalability: The ability to adapt to changes in the user base, data volume, and system demands, ensuring the long-term viability and performance of the system.

1.2.10 Reliability: Ensuring the system maintains high availability and minimal downtime, supported by mechanisms such as backup, recovery, and redundancy.

**1.3 User Requirements Definition**

1.3.1 Stakeholder Identification

| **Stakeholder** | **User Pain Points** |
| --- | --- |
| Students | 1. Difficulty in accessing resources and support needed for their learning 2. Limited opportunities for personalized attention and engagement in the classroom 3. Challenges in adapting to the pace and curriculum of the school |
| Teachers | 1. Heavy workloads and limited time for providing individualized support to students 2. Insufficient access to teaching materials and tools tailored to diverse student needs 3. Difficulty in addressing the varying learning styles and abilities within the classroom |
| Parents/Guardians | 1. Lack of visibility into their child's academic progress and behavioral trends 2. Challenges in communicating effectively with teachers and school administrators 3. Difficulty in understanding and navigating the school's policies and procedures |
| School Administrators | 1. Maintaining accurate and up-to-date student records and information 2. Ensuring efficient resource allocation and utilization across the school 3. Addressing the diverse needs of students and teachers while balancing budgets and regulations |

1.3.2 Derived Core Functionalities

| **Derived Core Functions** | **Descriptions** |
| --- | --- |
| Student Management | Provide a user-friendly interface for teachers and administrators to manage student profiles, including personal information, academic records, and attendance data. |
| Attendance Tracking | Streamline the process of monitoring and recording student attendance, including absences and tardiness, with alerts and reporting capabilities. |
| Academic Performance Monitoring | Enable teachers to track and analyze student academic progress, including grades, test scores, and course completion, and generate performance reports. |
| Communication and Collaboration | Facilitate effective communication and collaboration between teachers, students, and parents/guardians through messaging, announcements, and shared resources. |
| User Management | Implement a robust user management system to register, verify, and configure accounts for teachers, and students, with appropriate permission levels and access controls. |
| Reporting and Analytics | Generate comprehensive reports and analytics to provide insights into student performance, attendance patterns, and other key metrics, supporting data-driven decision-making. |
| Accessibility and Usability | Ensure the system is user-friendly and accessible for all stakeholders, with support for various devices and platforms, and consider features like mobile responsiveness and intuitive interfaces. |
| Security and Data Privacy | Implement robust security measures to safeguard user data, ensure confidentiality, and comply with relevant data protection regulations. |
| Scalability and Reliability | Design the system to be scalable, capable of handling increasing user and data demands, and highly reliable, with mechanisms for backup, recovery, and redundancy. |

**1.4 System Requirements Specification**

1.4.1 Functional Requirements

1.4.1.1 User Management

The system should support secure user registration and authentication mechanisms for teachers, students, and professionals. Each user should have a distinct profile that includes relevant information and assigned appropriate access rights. The system should facilitate user account management and permissions administration for administrators.

| **User role** | **Access right** |
| --- | --- |
| School Administrator | * View Login Account * Create/ View/ Delete Users (Teacher/ Students)   + User Registration (Name/ Email/ Password) * Create/ View/ Edit/ Delete Announcement * Create/ View/ Edit/ Delete Courses   + View Student List   + Create/ View/ Edit/ Delete Assignments     - Add/ View/ Edit Marks   + View Statistics     - Attendance Ratio     - Assignment Marks ratio * Add/ View/ Edit Attendance * View Calandar * View Dashboard   + All Course   + All Announcement |
| Teachers | * View Login Account * Create/ View/ Edit Courses   + View Student List   + Create/ View/ Edit/ Delete Assignments     - Add/ View/ Edit Marks   + View Statistics     - Attendance Ratio     - Assignment Marks ratio * Add/ View/ Edit Attendance * View Calandar   + Teached Course * View Dashboard   + Teached Course   + Teacher Announcement |
| Students | * View Login Acount * View Enrolled Courses   + Upload Assignments     - View Marks   + View Statistics     - Self-Attendance Ratio     - Self-Assignment Marks ratio * View Calandar   + Enrolled Course * View Dashboard   + Enrolled Course   + Student Announcement |

1.4.1.2 Personalized Learning

The system should employ a comprehensive approach to deliver Individualized Education Plan (IEP) tailor-made to the specific needs of students. It should offer an extensive range of educational resources, including text-based materials, audio and video content, interactive exercises, and adaptive learning tools. The system should track and record students' progress, provide timely feedback, and performance insights to other parties.

1.4.1.3 Communication and Collaboration

The system should enable seamless communication channels among students/ parents, and teachers, leveraging features like announcements. It should facilitate collaboration on assignments and projects, fostering peer interaction and group learning opportunities. Teachers should have the ability to provide feedback and support to students after evaluating student progress through various reports.

1.4.1.4 Accessibility

The system should adhere to established accessibility standards to ensure optimal usability for students. It should support features such as adjustable font sizes, and other visual settings to accommodate diverse needs. The system should be compatible with assistive technologies, including screen readers, magnifiers, alternative input devices, etc.

1.4.1.5 Reporting and Analytics

The system should generate comprehensive reports on student performance, progress, and engagement, facilitating data-driven decision making. It should provide sophisticated data analytics capabilities to identify areas for improvement, measure the effectiveness of interventions, and support evidence-based practices. Administrators should be able to generate customizable reports and export data for further analysis and reporting purposes.

1.4.2 Non-Functional Requirements

1.4.2.1 Usability

The system should feature an intuitive and user-friendly interface that promotes ease of navigation and efficient interaction. Clear instructions, guidance, and contextual help should be provided within the system to ensure a seamless user experience.

1.4.2.2 Security

The system should prioritize the confidentiality and integrity of user data, implementing robust security measures such as encryption, secure data transmission, and user authentication mechanisms. It should comply with industry best practices and regulatory requirements for data protection and privacy.

1.4.2.3 Scalability

The system should be designed to accommodate a changing number of users, varying data volumes, and number of concurrent user interactions without significant performance degradation. Scalability considerations should include hardware, software, and network infrastructure, ensuring the system can handle future potential growth.

1.4.2.4 Reliability

The system should exhibit a prominent level of availability, minimizing downtime and ensuring uninterrupted access for users. It should incorporate backup and recovery mechanisms to safeguard against data loss and ensure system continuity in the event of failures or disruptions.

**1.5 Constraints and Assumptions**

The Learning Management System will be developed using modern web-based technologies, adhering to industry best practices and standards. It will be compatible with popular web browsers across various devices, including desktops, laptops, tablets, and mobile phones. Internet connection is needed to access and interact with the system.

**1.6 System Models**

1.6.1 Description of the Learning Management System

In the system, students can access course materials, submit assignments, and track their own progress. Parents can obtain school announcements, monitor their children's progress, and access support resources through the student profile. Teachers can create and manage course content, track students' progress and view attendance reports. Administrators manage user accounts, manage announcements, generate reports based on the information on the platform, and allocate resources to develop and maintain the system under educational policies.

1.6.2 Description of the System Architecture

The system architecture for the Learning Management System shall consist of Presentation, Application, and Data Layers. The Presentation Layer creates a visually appealing, user-friendly web interface in responsive design. Application Layer handles core functionalities through microservices, including user management, course, assignment, announcement, and reporting. Data Layer utilizes databases for efficient storage and retrieval of structured and unstructured data. Thus, it ensures scalability, modularity, and responsiveness, so user experience can be enhanced overall.

**1.7 System Evolution**

The future enhancements and scalability considerations for the system should provide personalized learning paths tailored for individual students, enhancing collaboration features such as virtual classrooms and discussion forums, offering advanced analytics and insights for administrators and teachers, and continuously gathering user feedback for ongoing improvements.

**2. Design Document**

**2.1 Introduction**

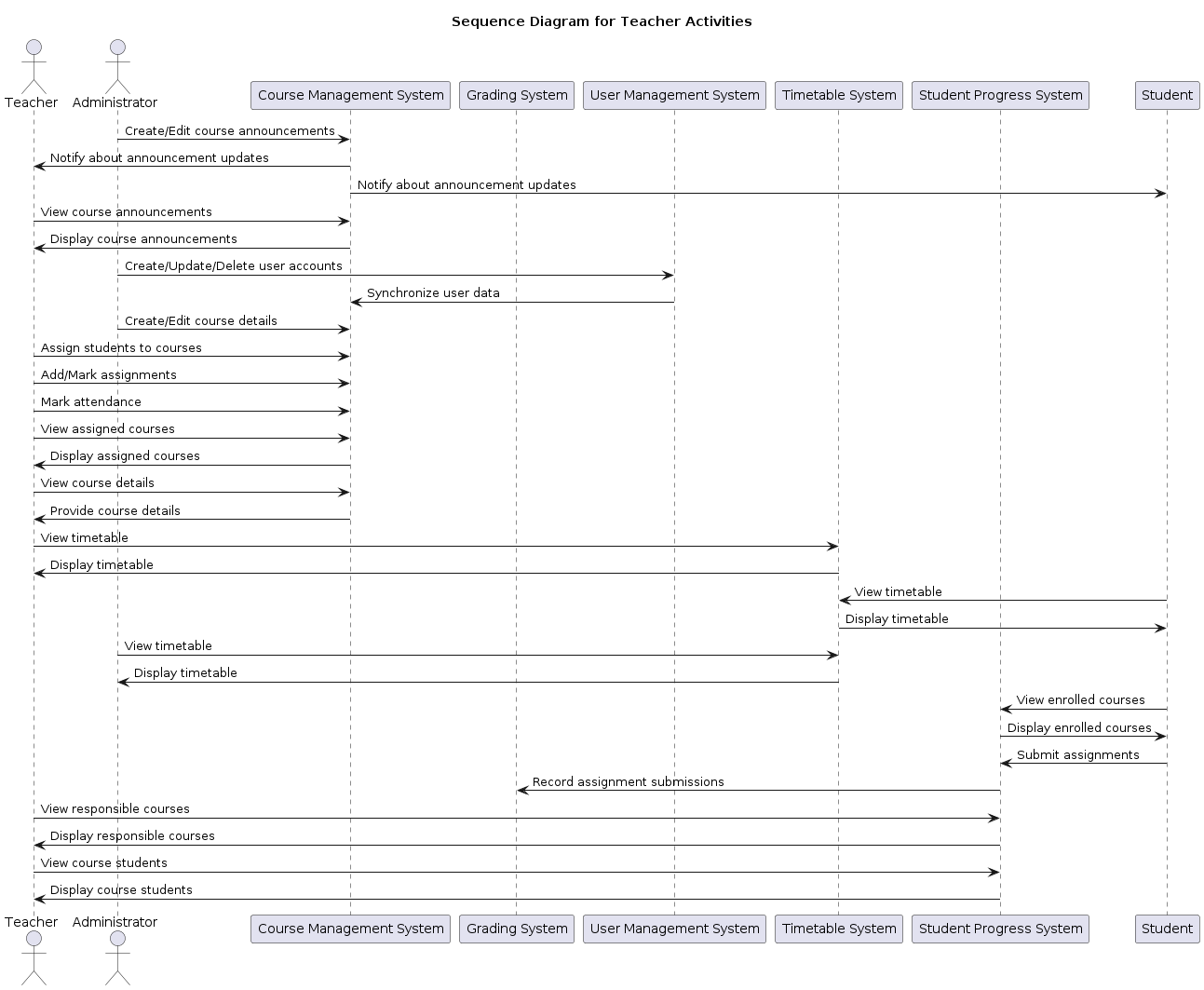
This comprehensive system fosters an inclusive environment, streamlines administrative tasks, and significantly enhances the overall educational experience, transforming the way stakeholders interact and collaborate within the educational ecosystem.

**2.2 System Overview**

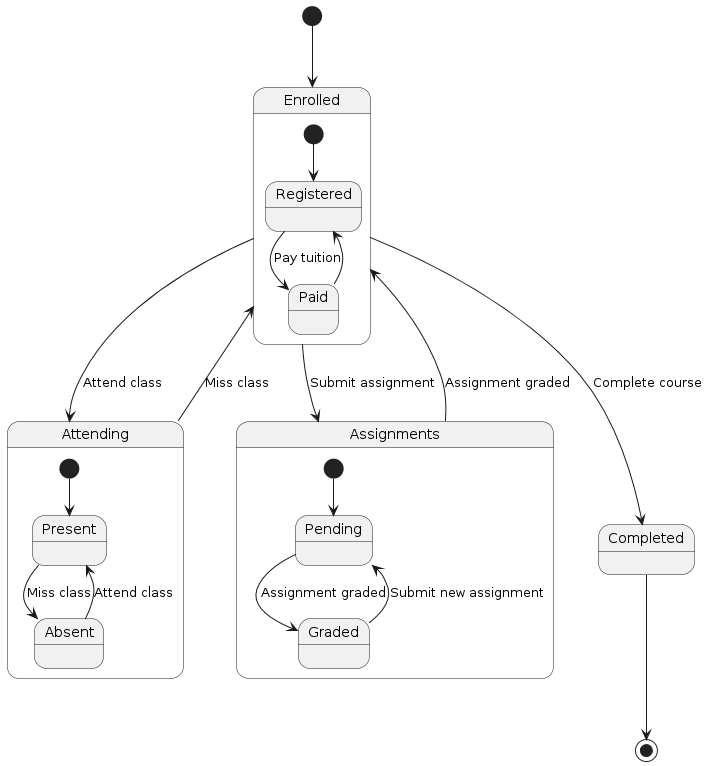
2.2.1 Use Case diagram



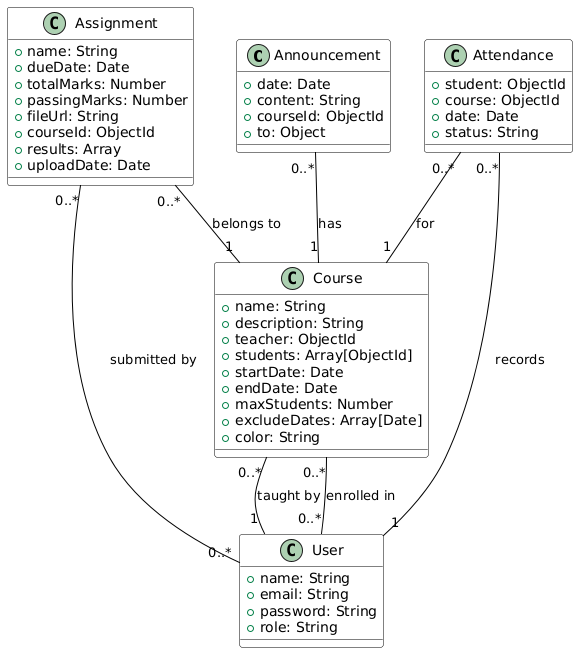
2.2.2 Sequence Diagram



2.2.3 State machine models



**2.3 Database Design**



**2.4 System Architecture**

| **Layer** | **Details** |
| --- | --- |
| Presentation Layer | 1. Follow the design for accessibility to provide optimal user experience across different devices and platforms.  2. User interface communicates with the application layer via APIs, facilitating decoupling and ease of maintenance. |
| Application Layer | 1. Comprising microservices, each responsible for a specific functional area.  2. Implemented using Node.js with Express, providing scalability and independent deployment.  3. Key microservices include:  a. User Management: Handles user authentication, authorization, and manages user profiles.  b. Course Management: Manages the creation, modification, and retrieval of course plans, including versioning and collaboration features.  c. Resource Management: Provides functionality to upload, store, and retrieve assignment resources (e.g. documents, images, videos, etc.)  d. Collaboration: Offers announcement features between students, teachers and administrators.  e. Reporting and Analytics: Collects and analyzes data to generate reports, insights, and performance metrics for administrators and teachers. |
| Data Layer | 1. Utilizes MongoDB, a NoSQL database, to store and retrieve data efficiently.  2. MongoDB is used to store unstructured or semi-structured data, such as user profiles, courses, resources, and collaboration data.  3. Data replication, sharding, or partitioning techniques may be employed to ensure high availability, fault tolerance, and scalability for future development..  4. Data access is abstracted through an Object Document Mapping (ODM) framework, like Mongoose, to provide a consistent and efficient way to interact with the MongoDB database. |

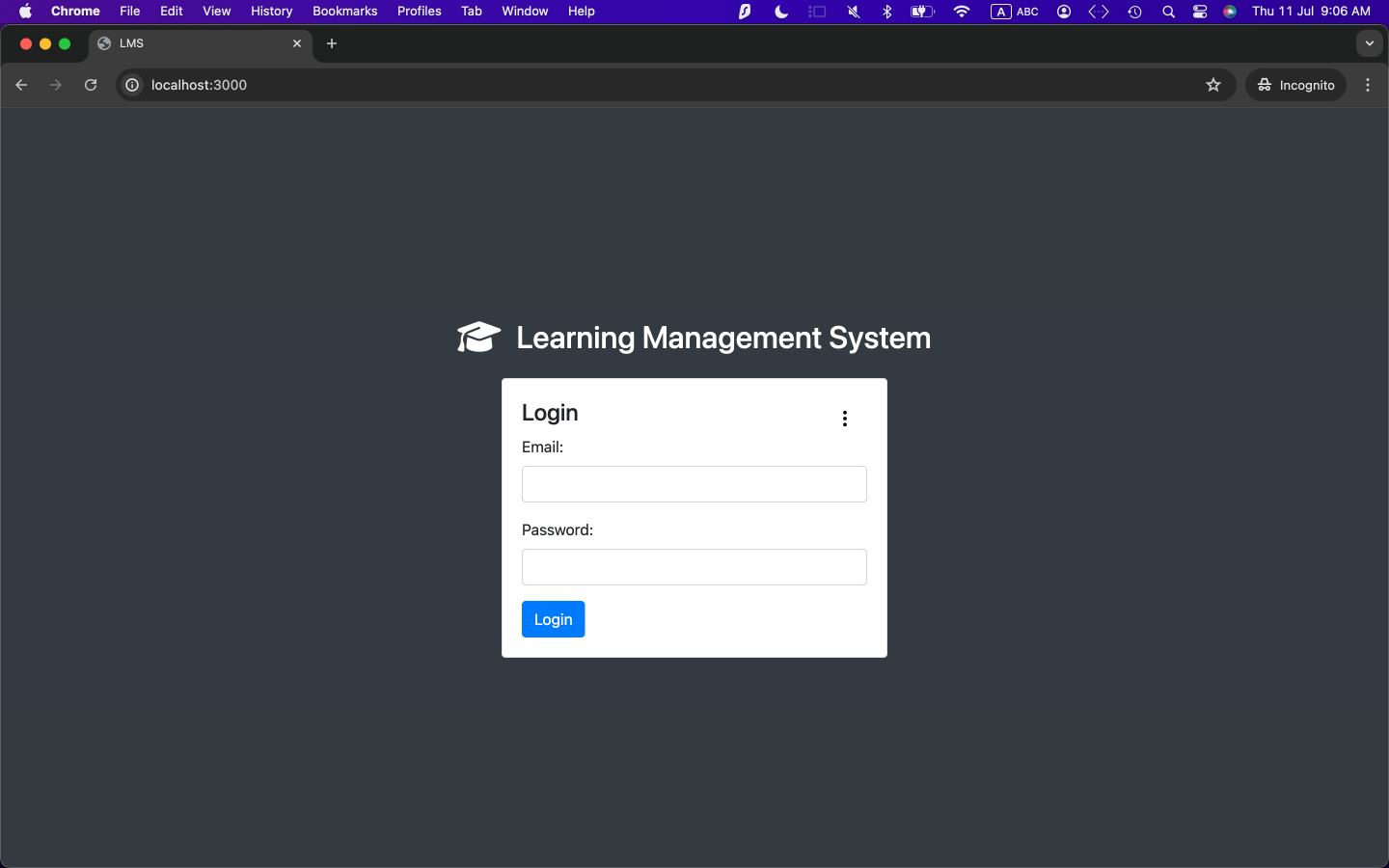
**2.5 User Interface Design**

Figure 1. Login Page with Token-Based Authentication and Hash Encryption

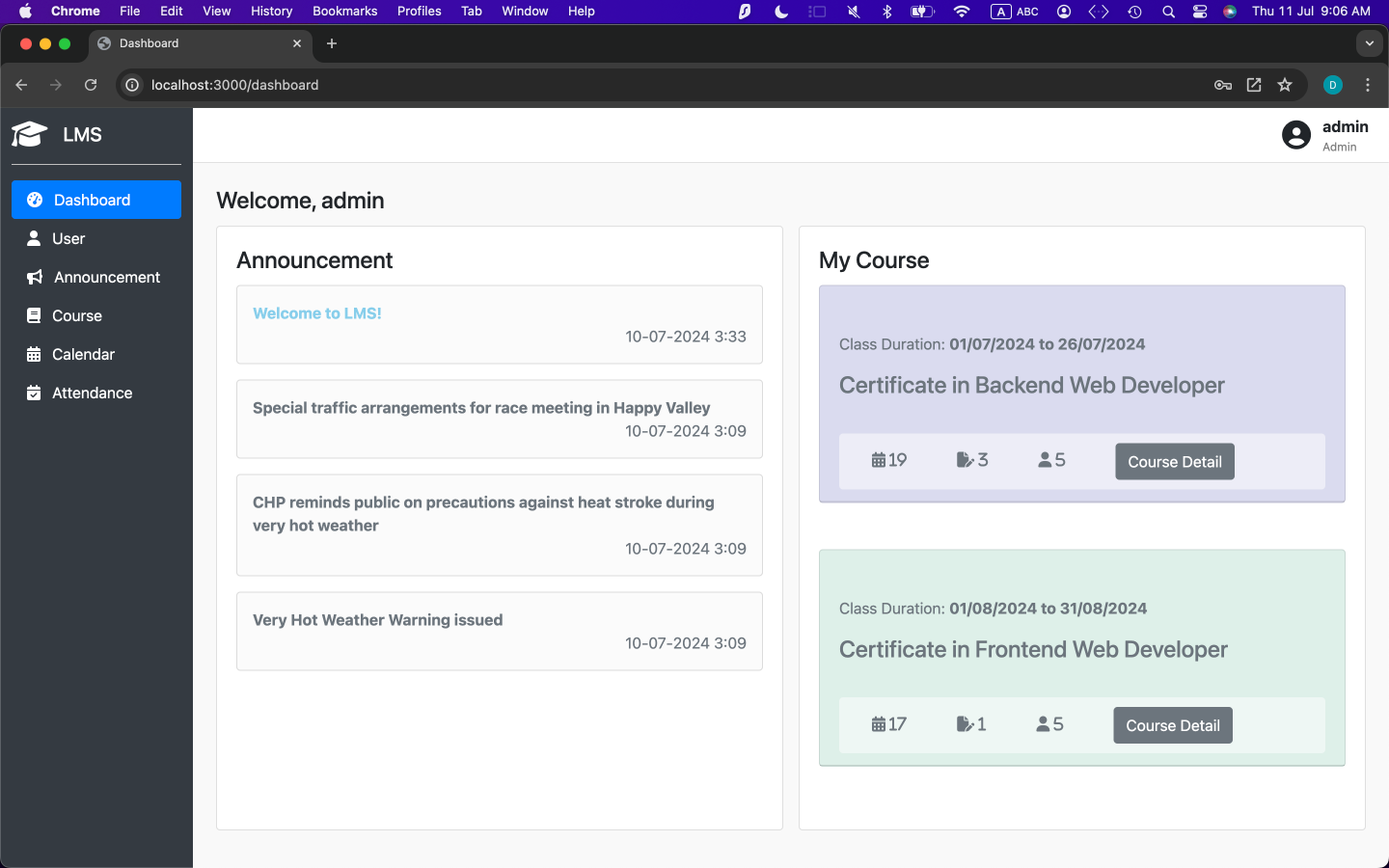


Figure 2. Dashboard with View Courses/ View Announcements

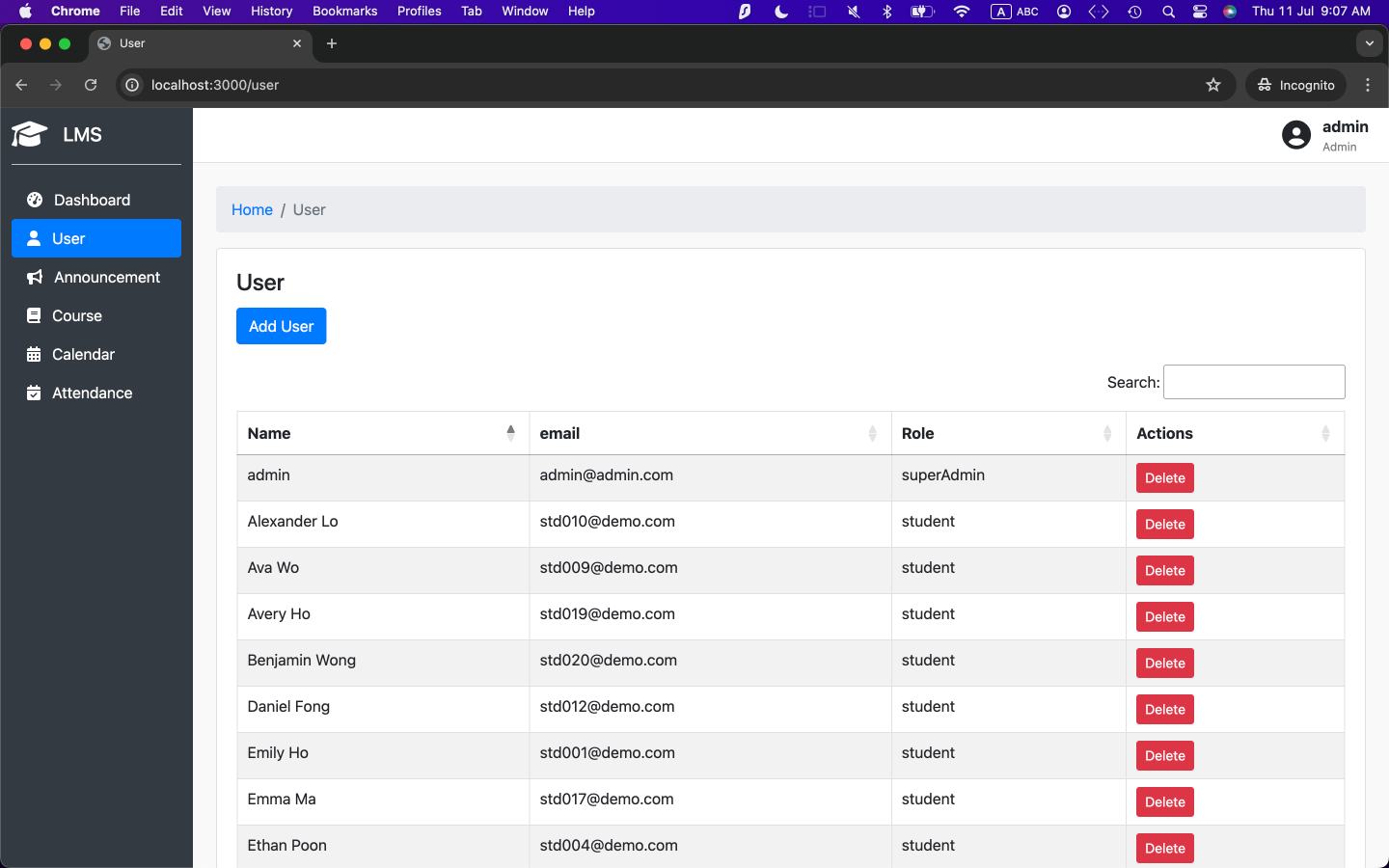


Figure 3. User Management

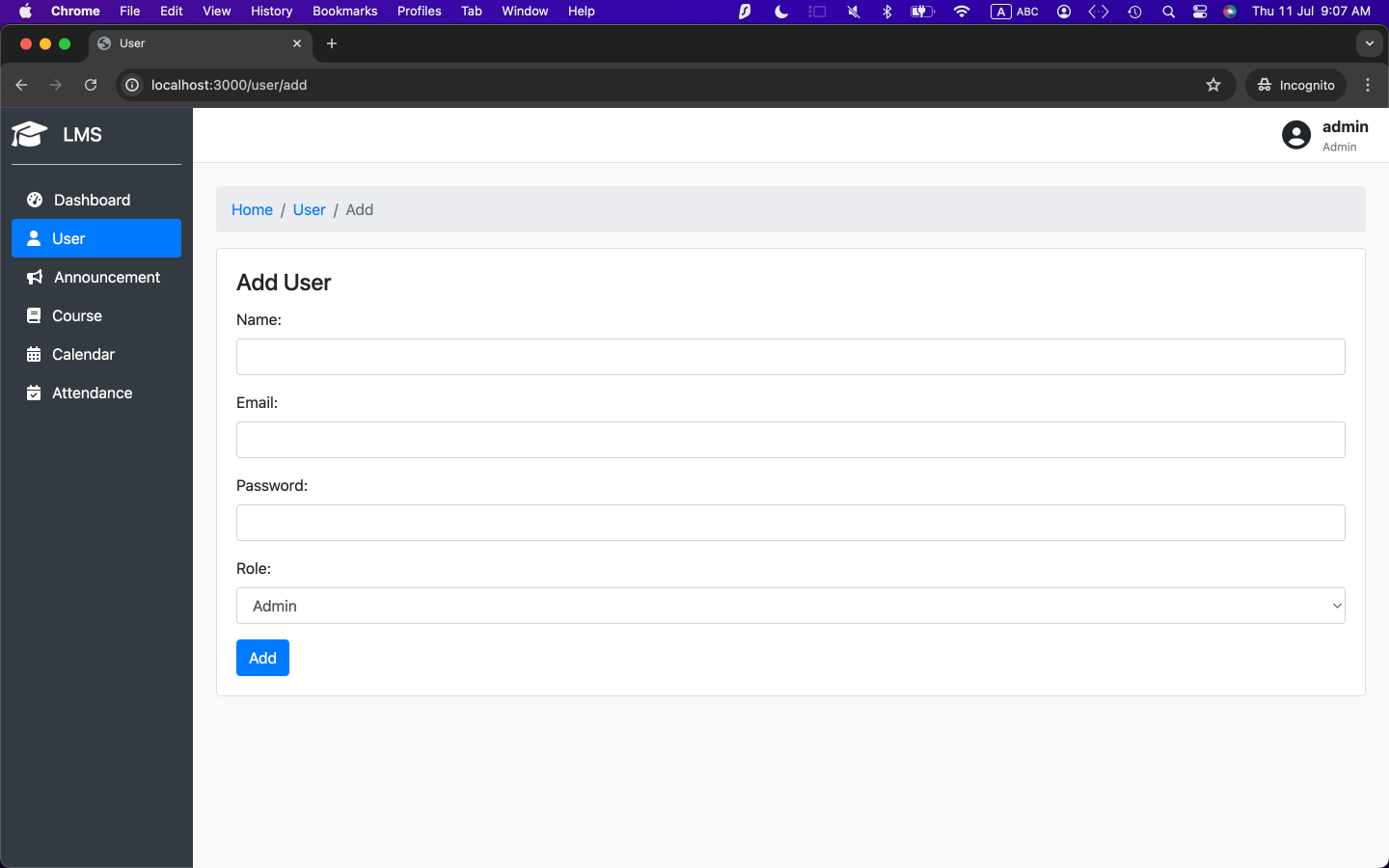


Figure 4. User Modification (Add)

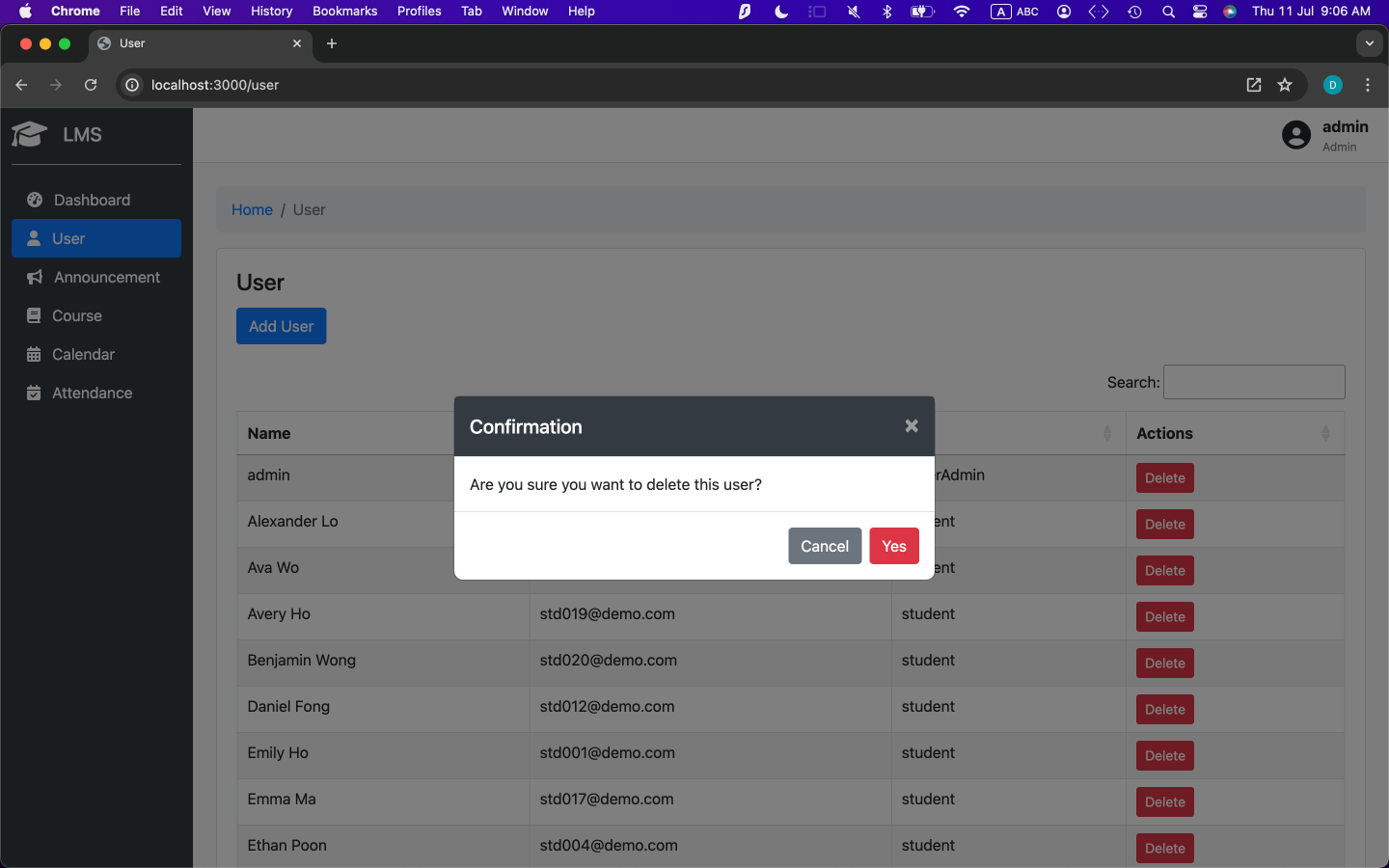
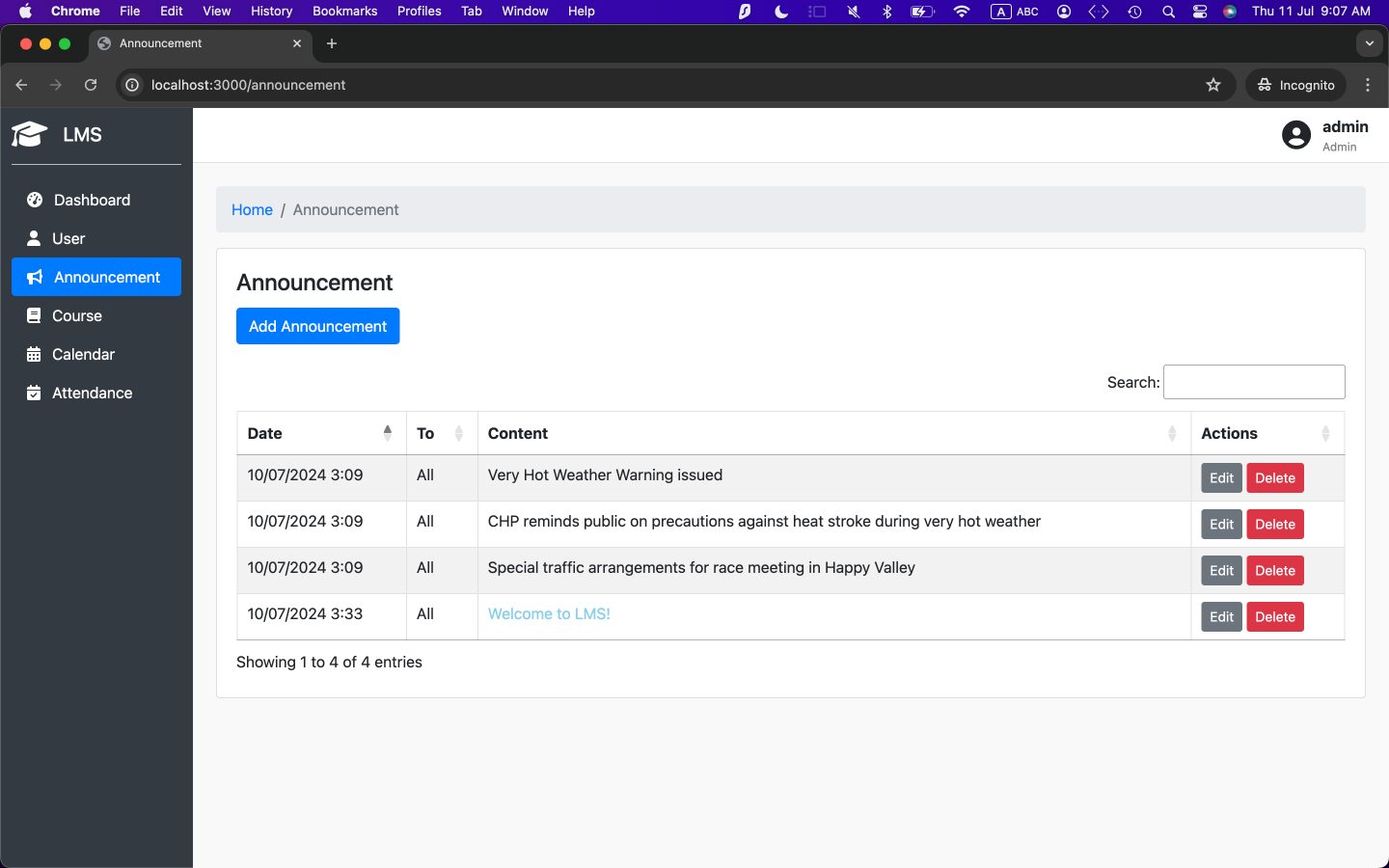
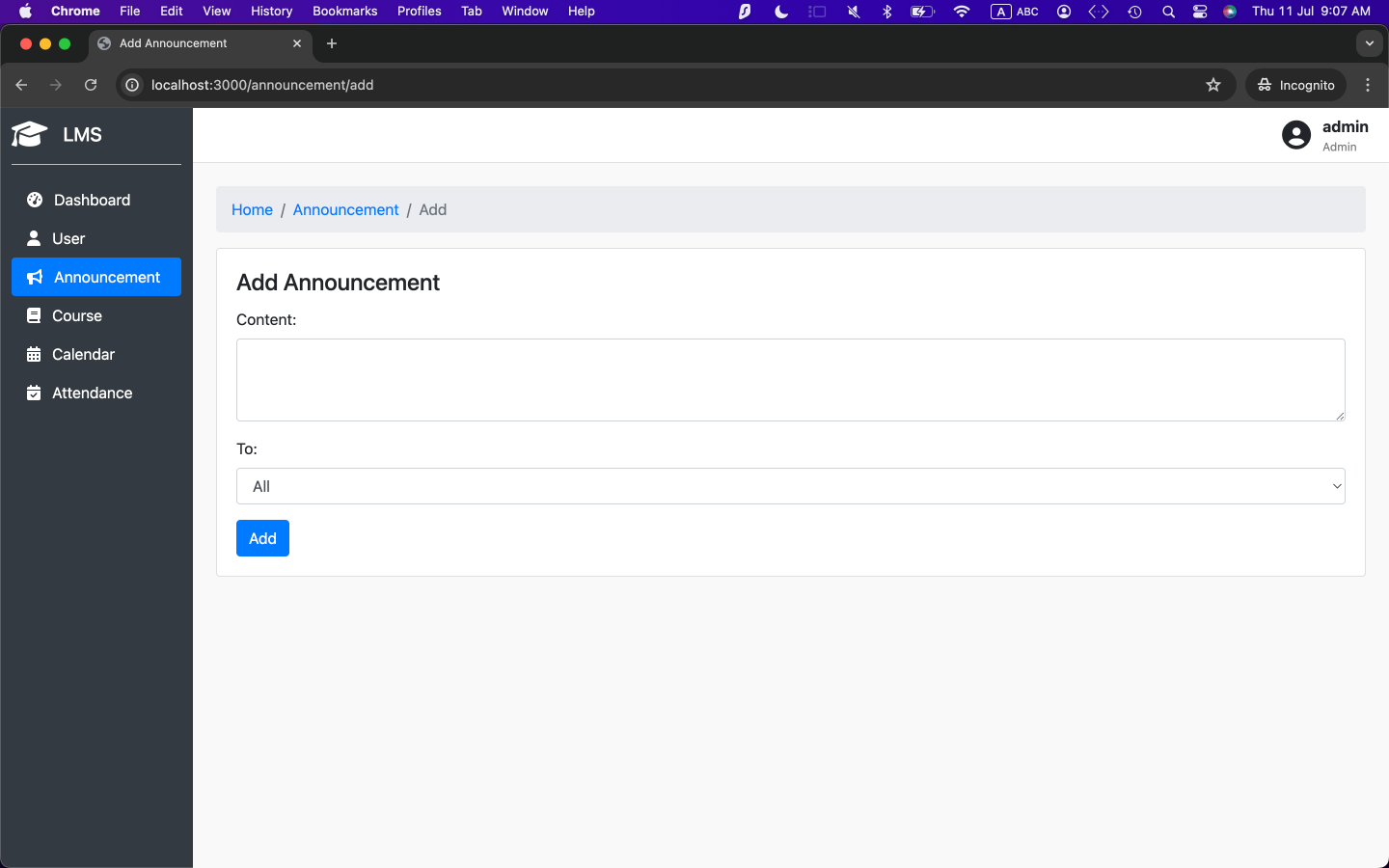
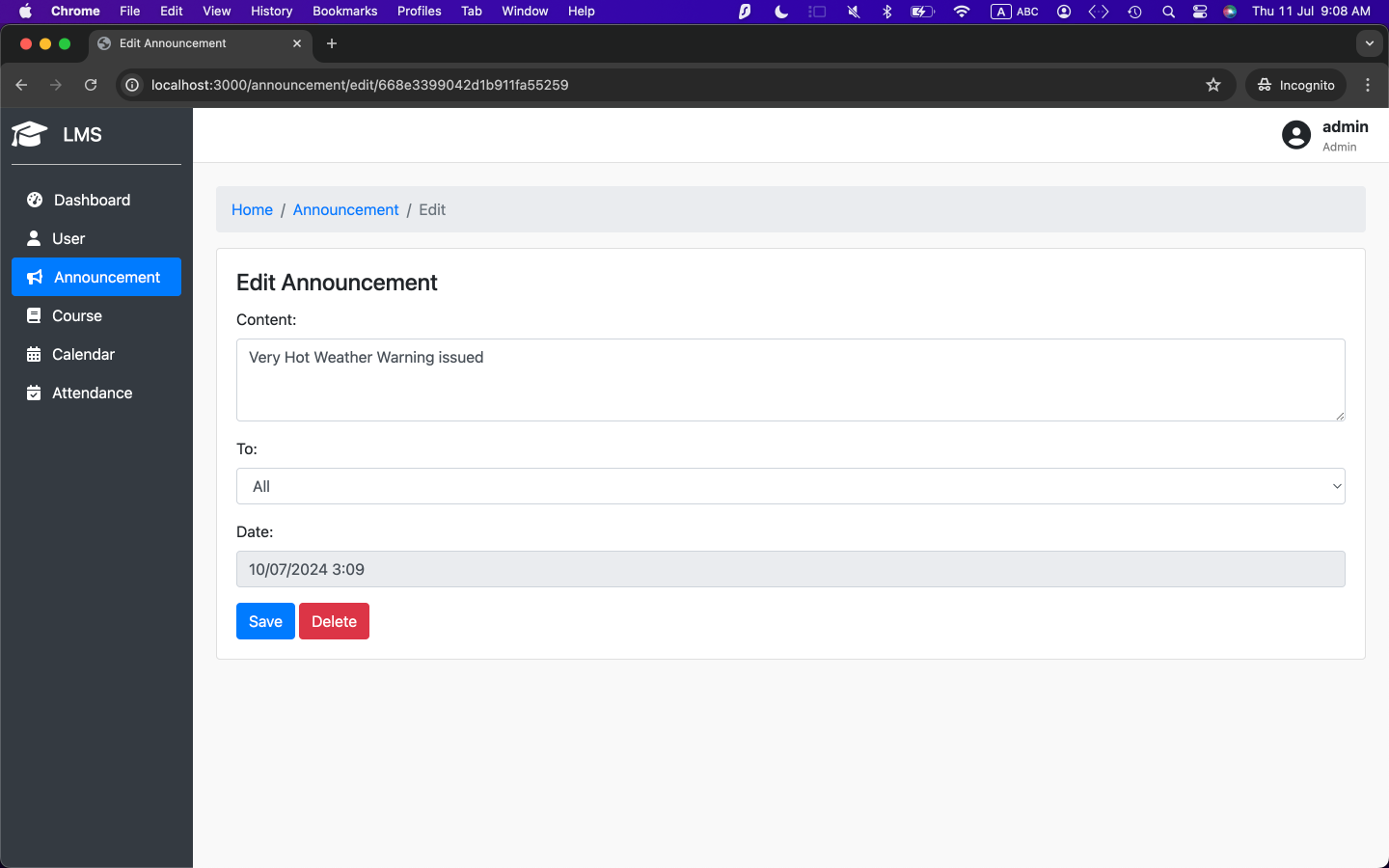
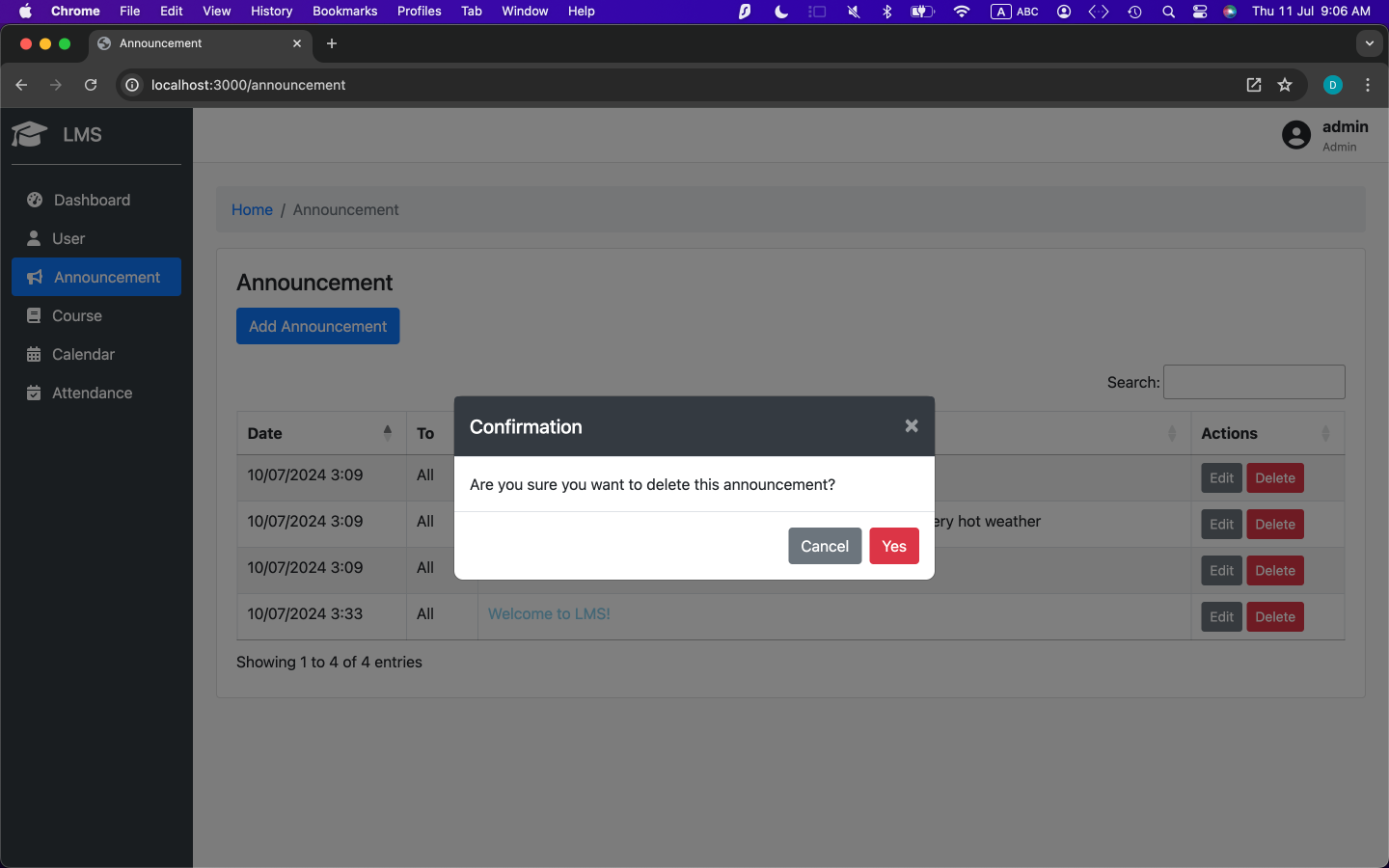


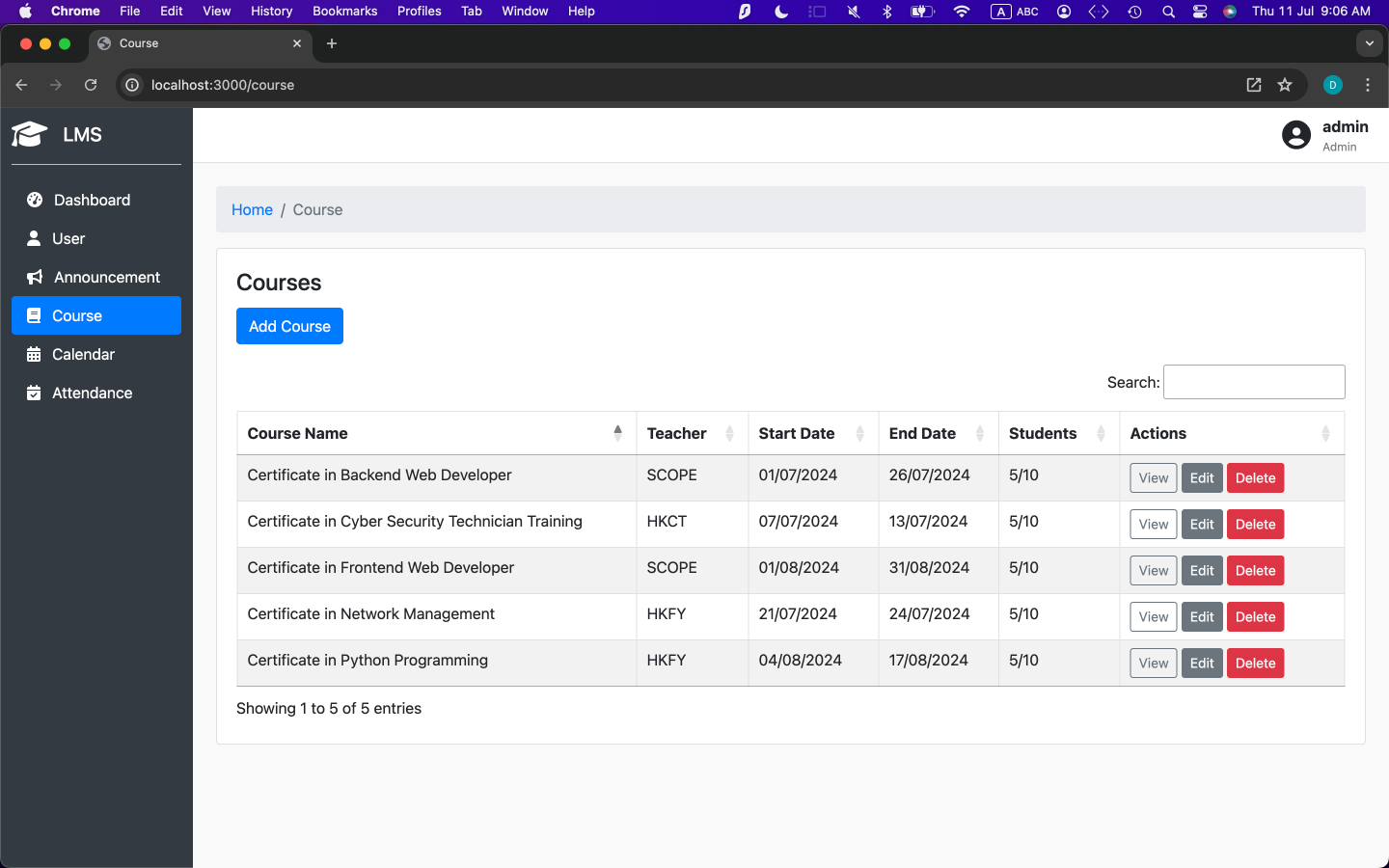
Figure 5. User Modification (Delete)

Figure 6. Announcement List

Figure 7. Announcement Modification (Add)

Figure 8. Announcement Management (Edit)

Figure 9. Announcement Management (Delete)

Figure 10. Course List

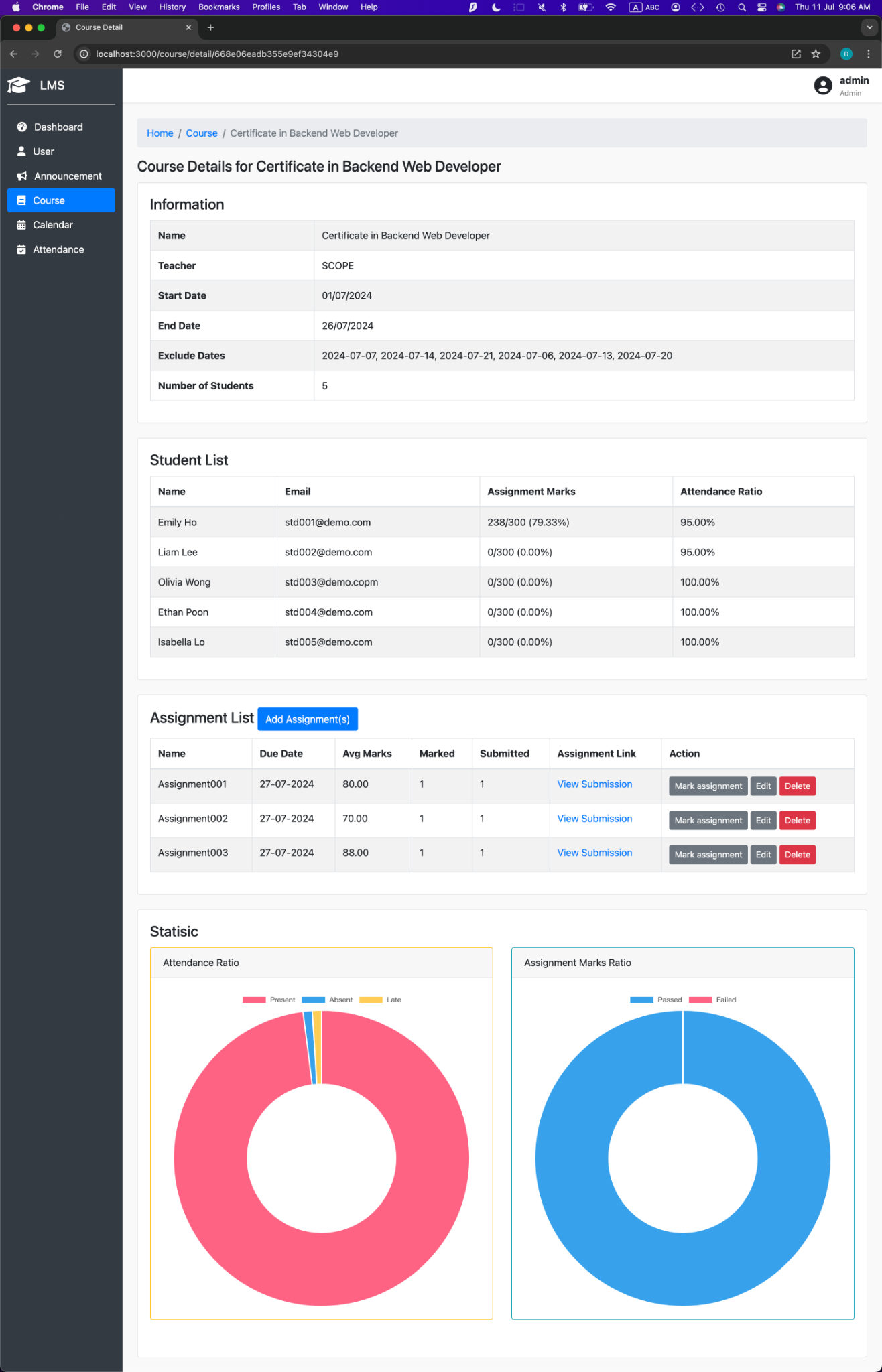
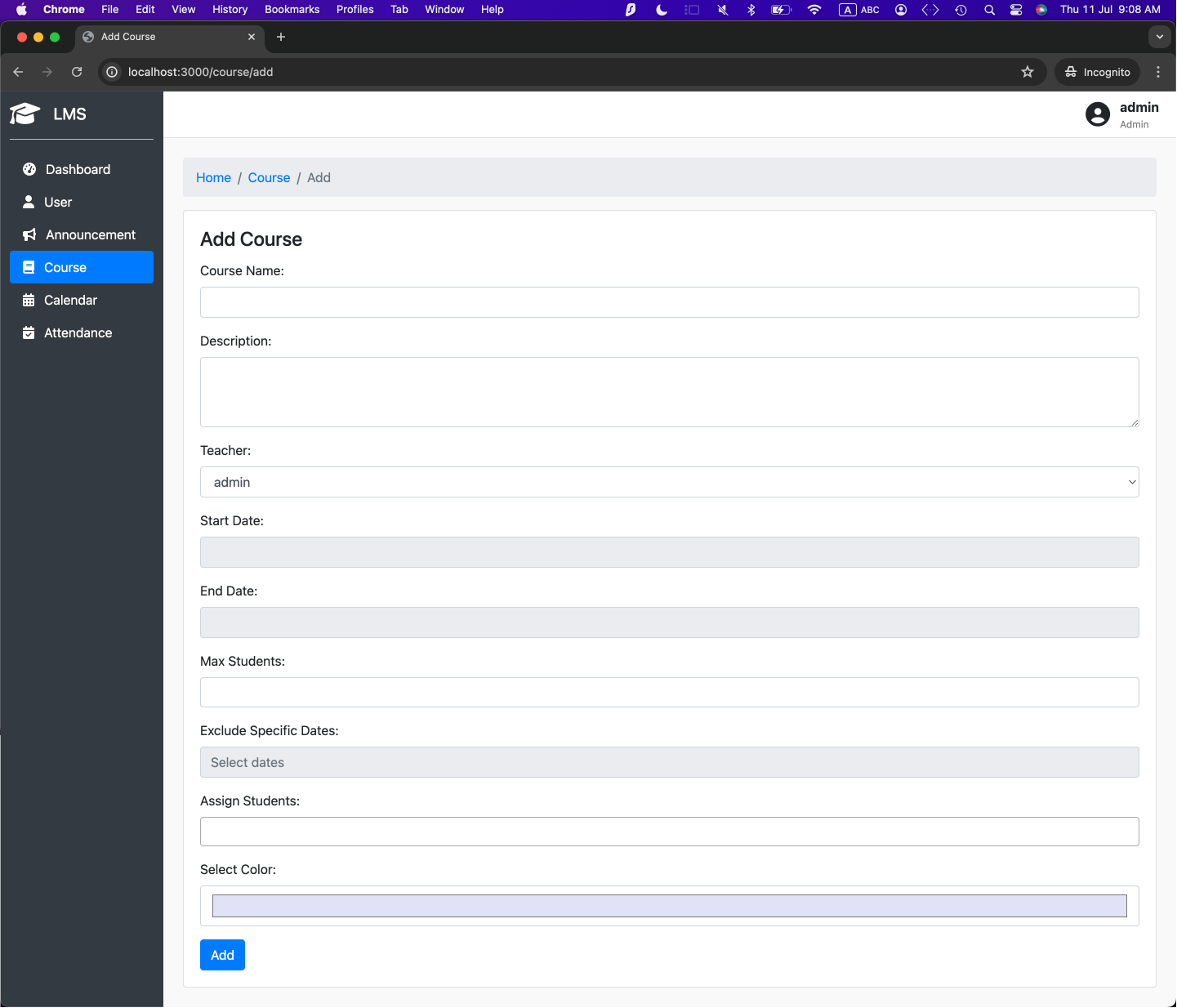


Figure 11. Course Details

Figure 12. Course Modification (Add)

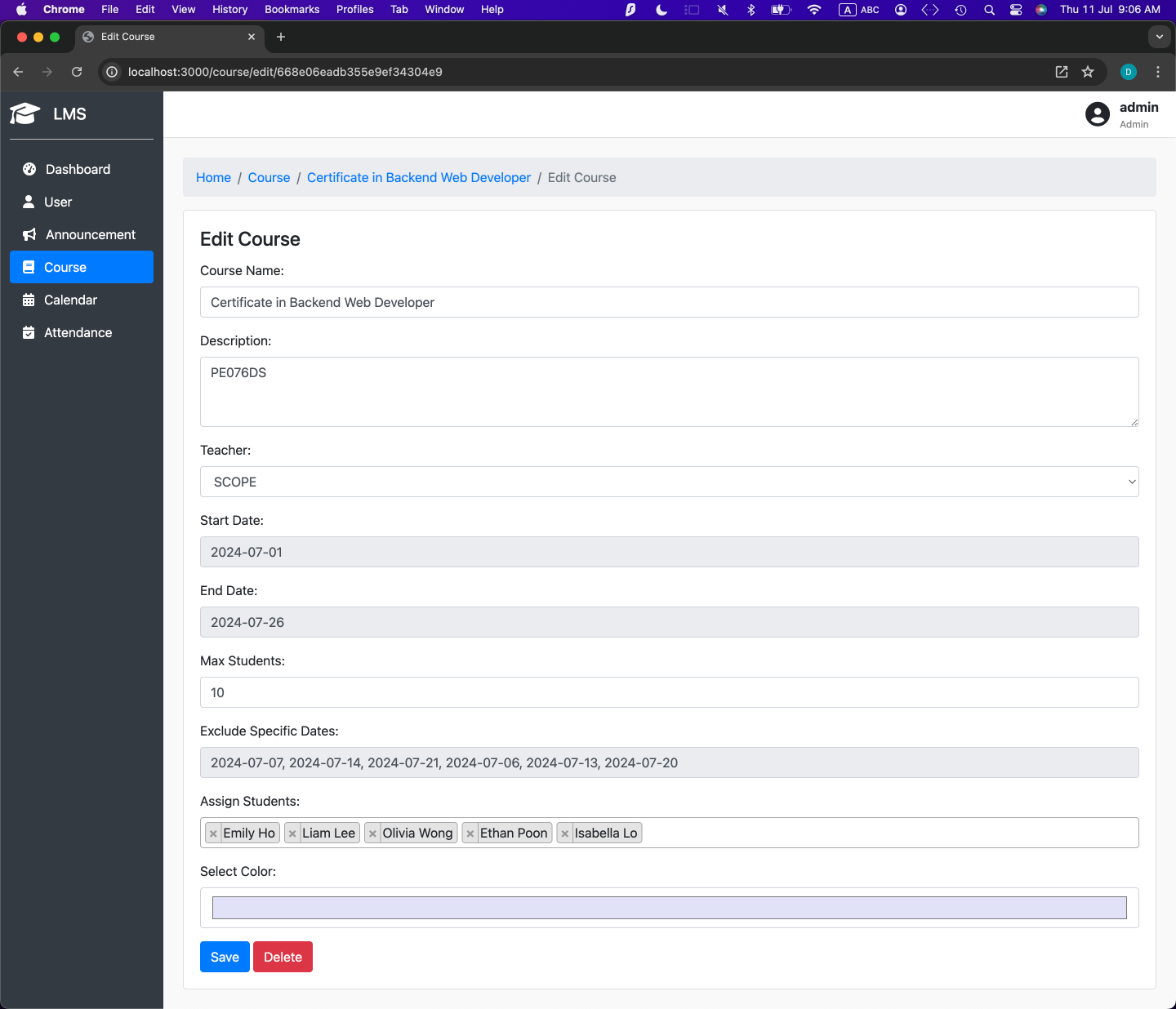


Figure 13. Course Modification (Edit)

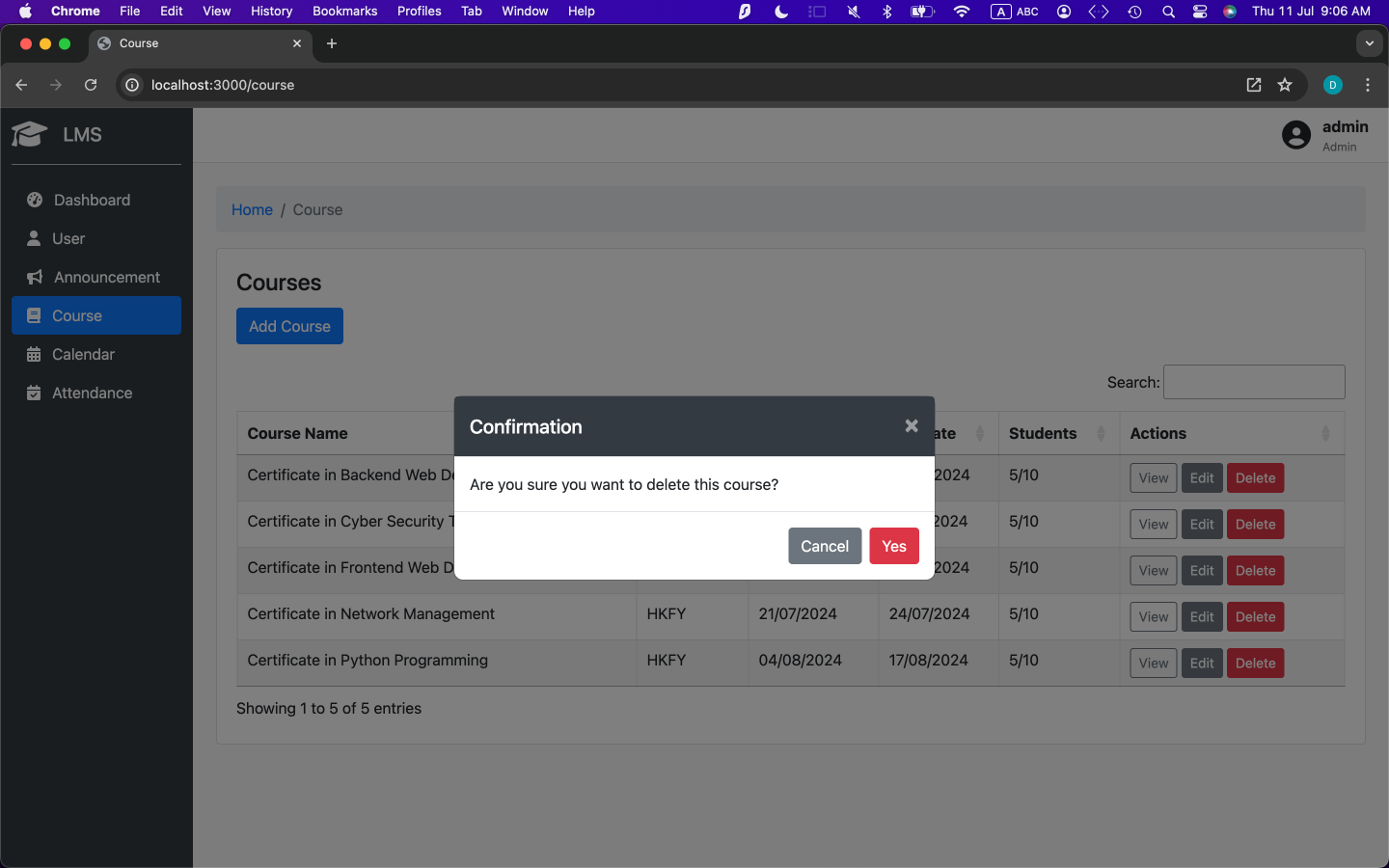
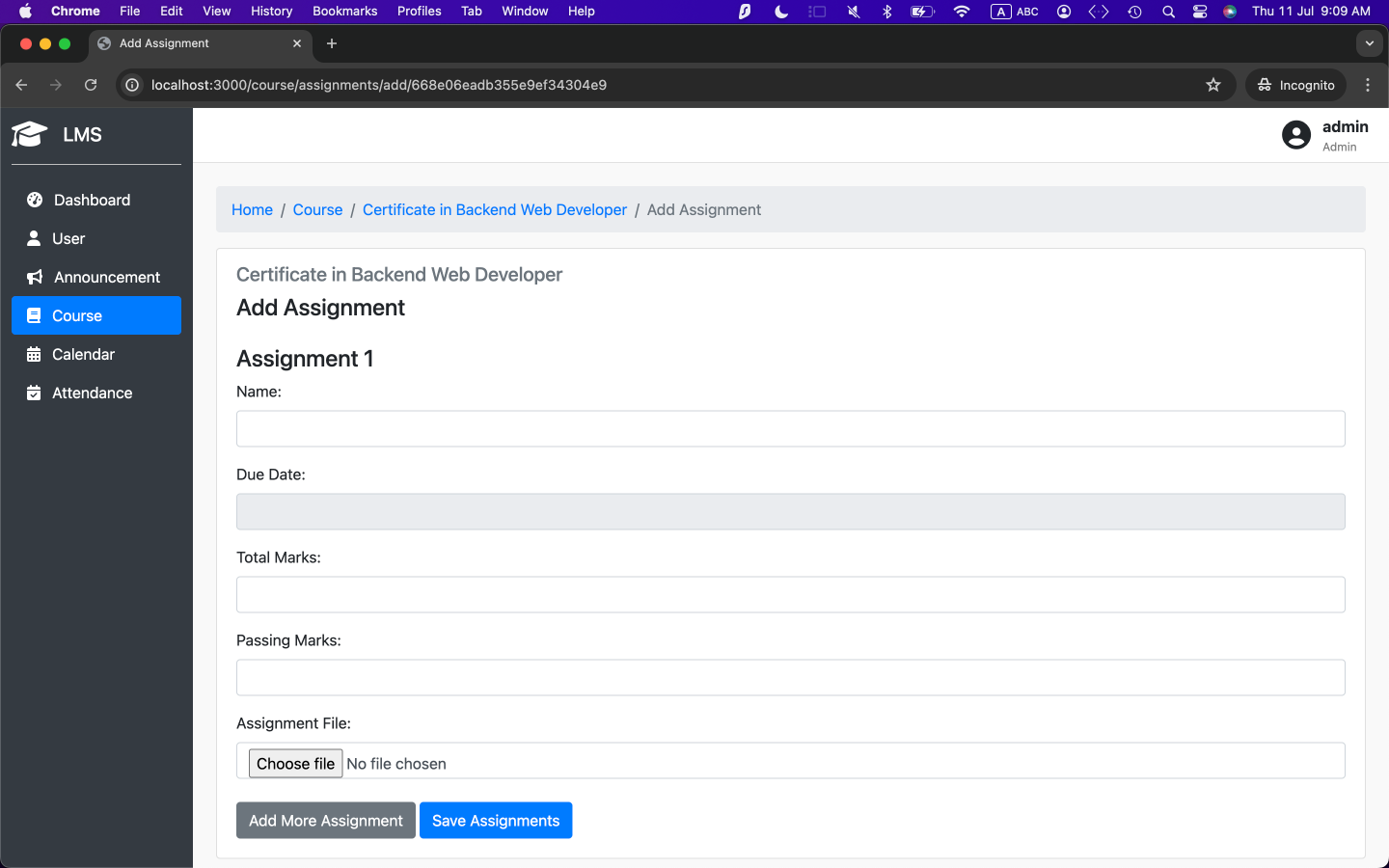


Figure 14. Course Modification (Delete)

Figure 15. Course Assignment Modification (Add)

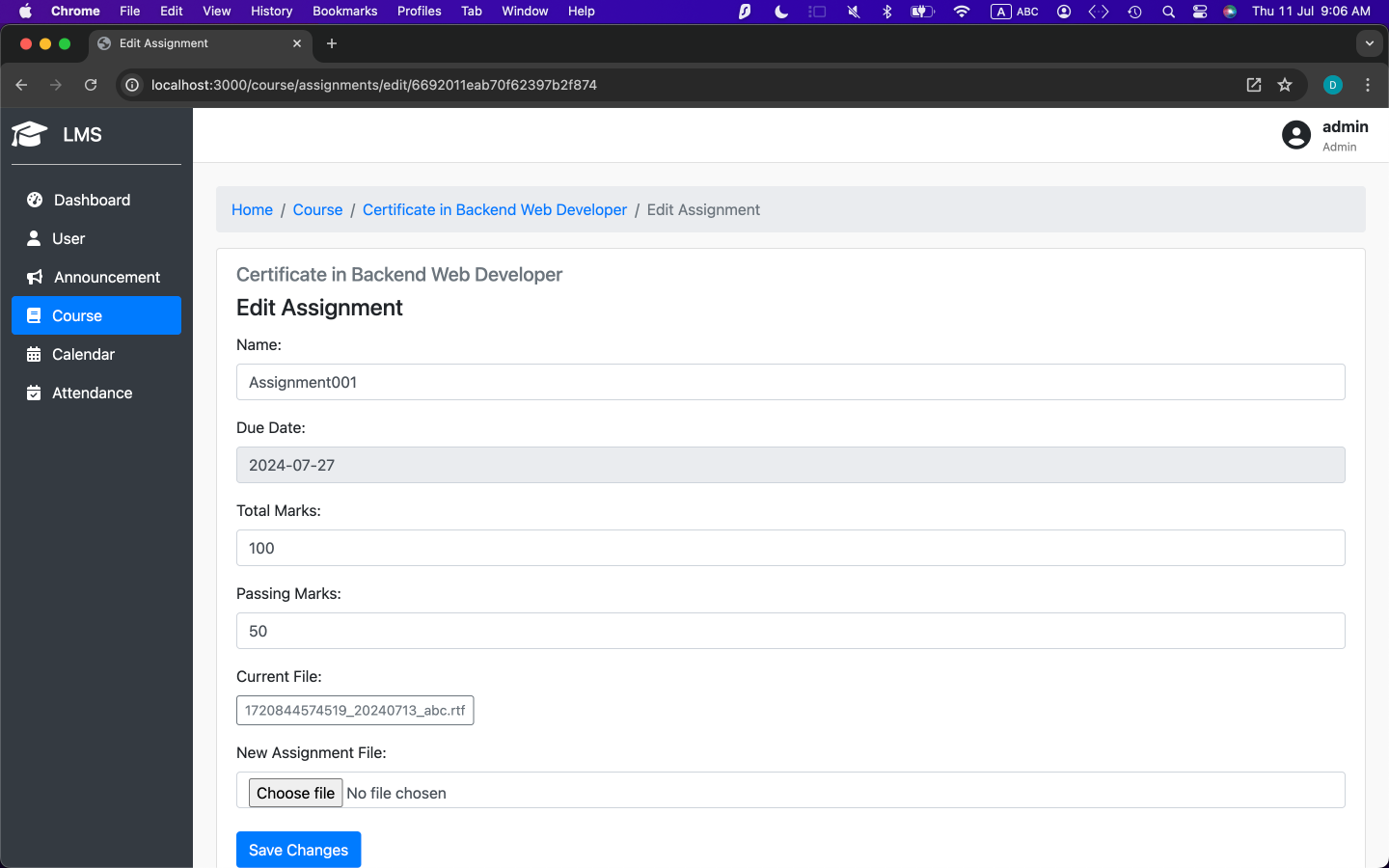
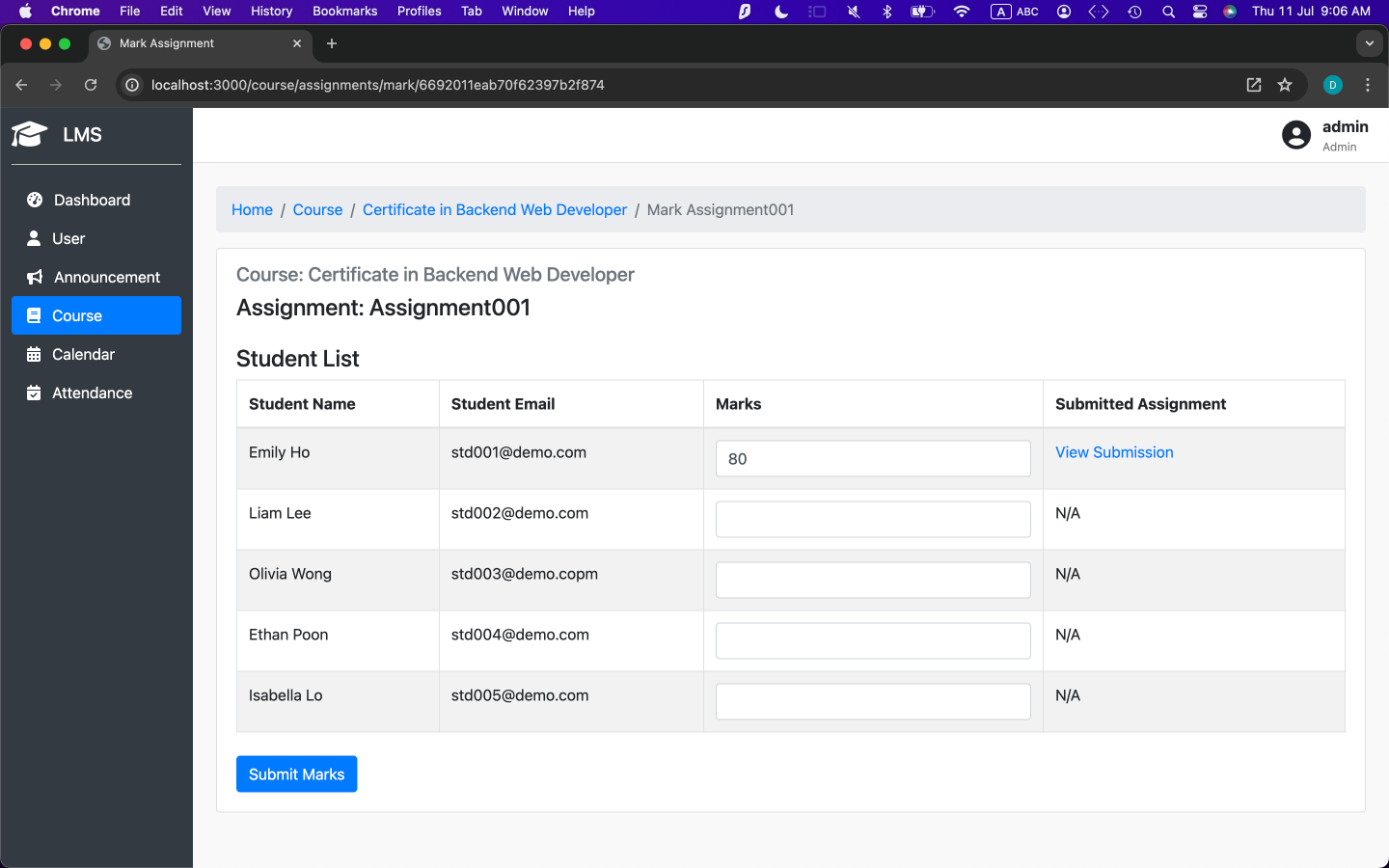
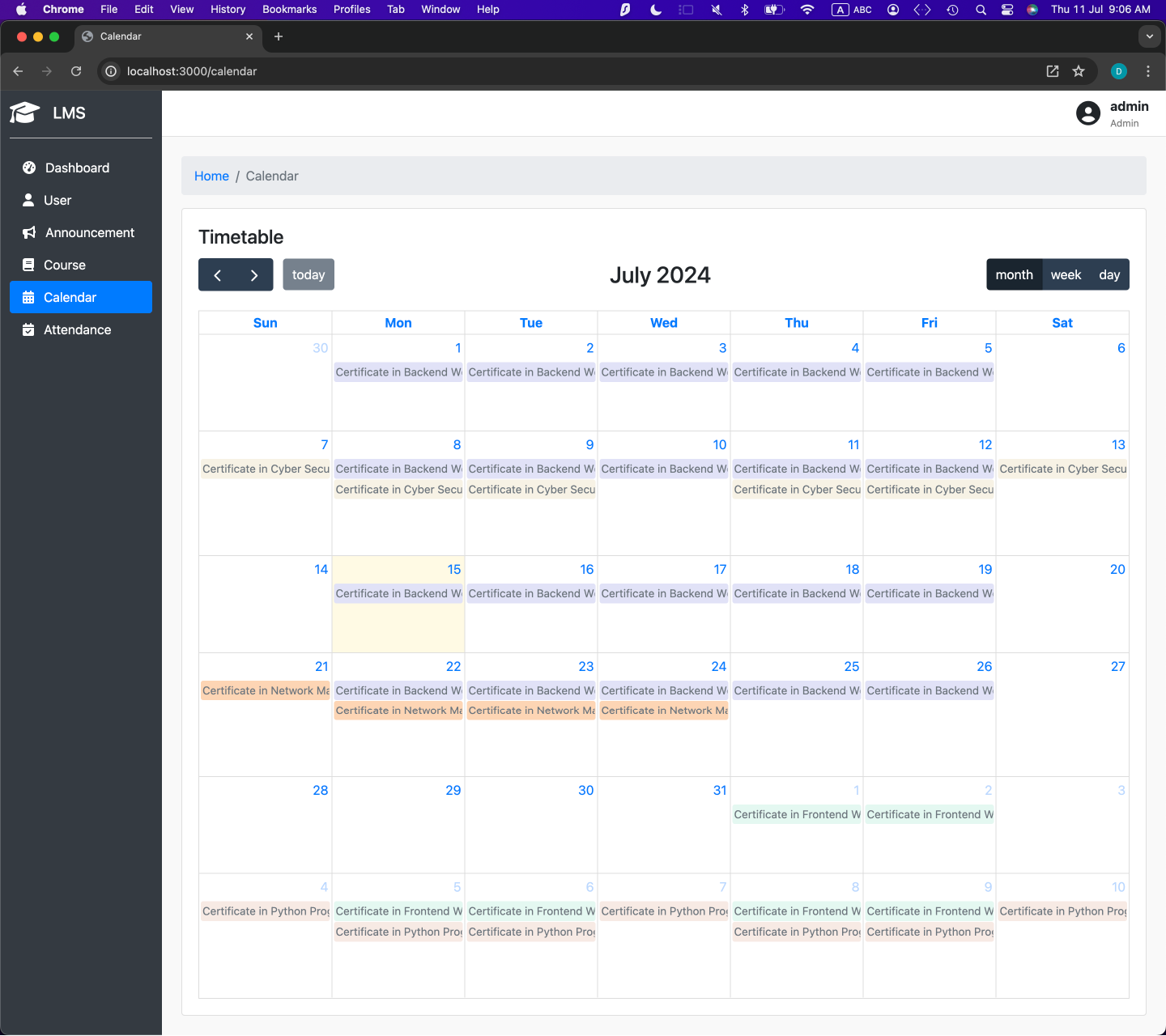


Figure 16. Course Assignment Modification (Edit)

Figure 17. Course Assignment Submission List (View/ Mark assignment)

  
Figure 18. Course Calandar

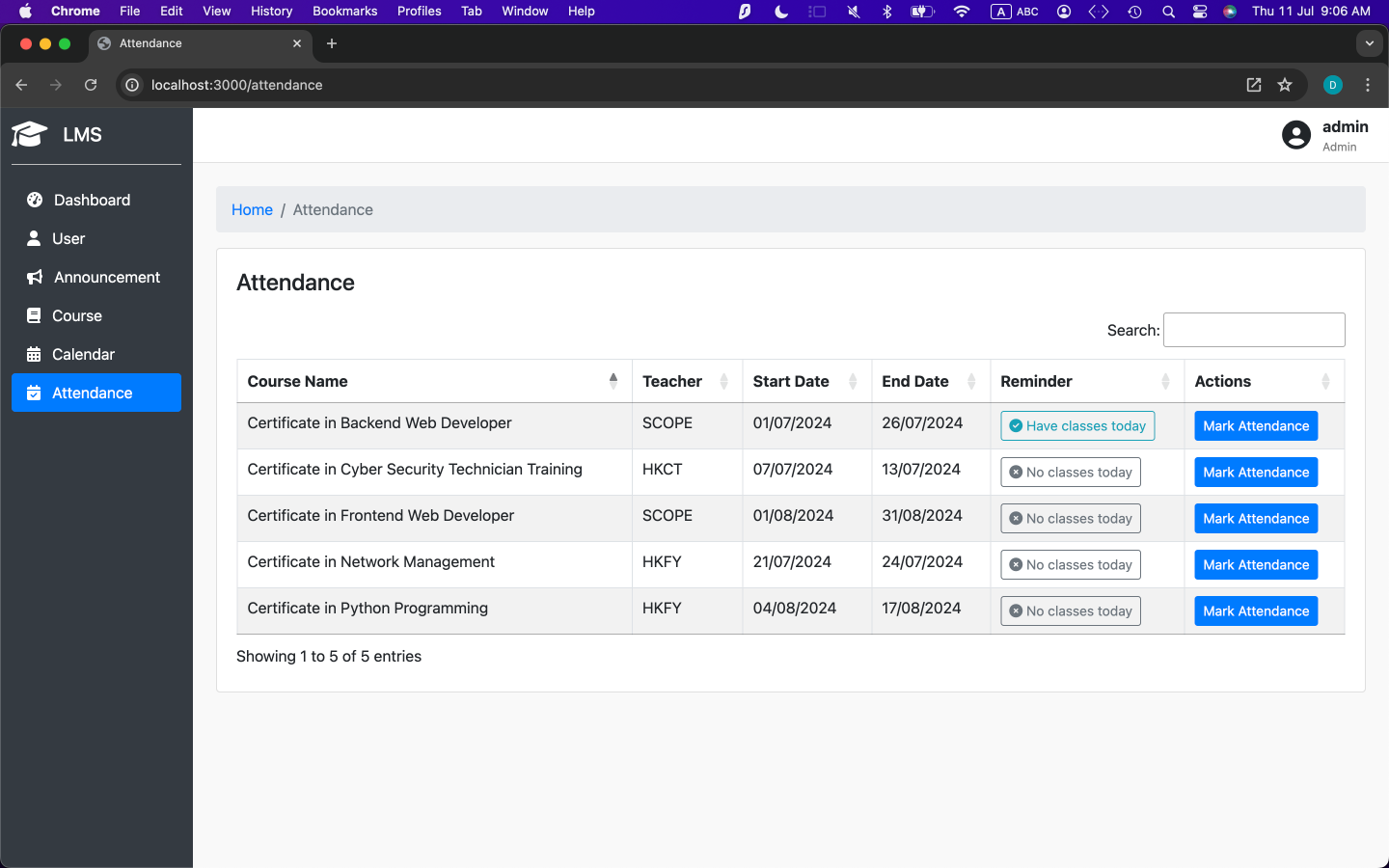


Figure 19. Course Attendance List



Figure 20. Course Attendance Record

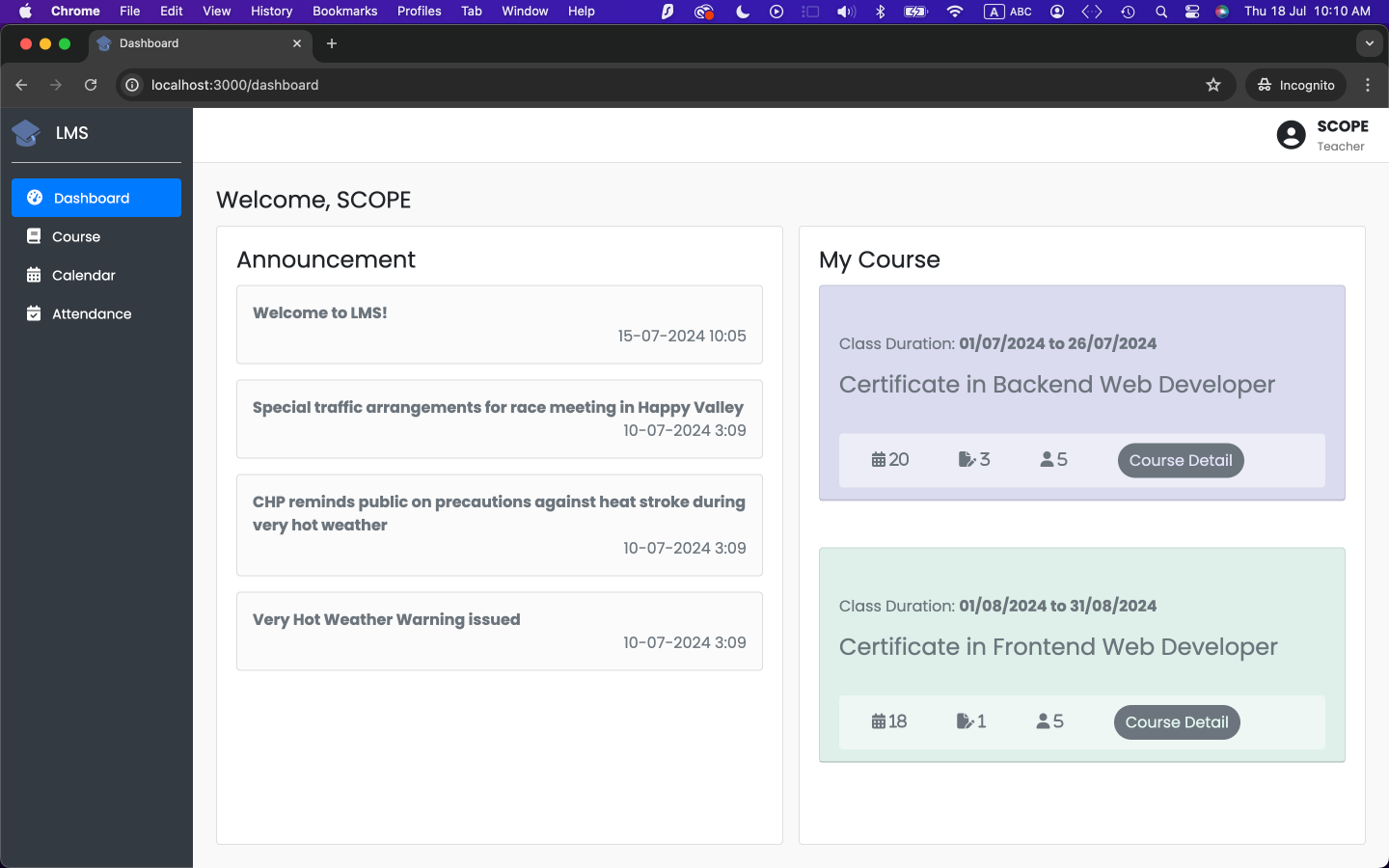
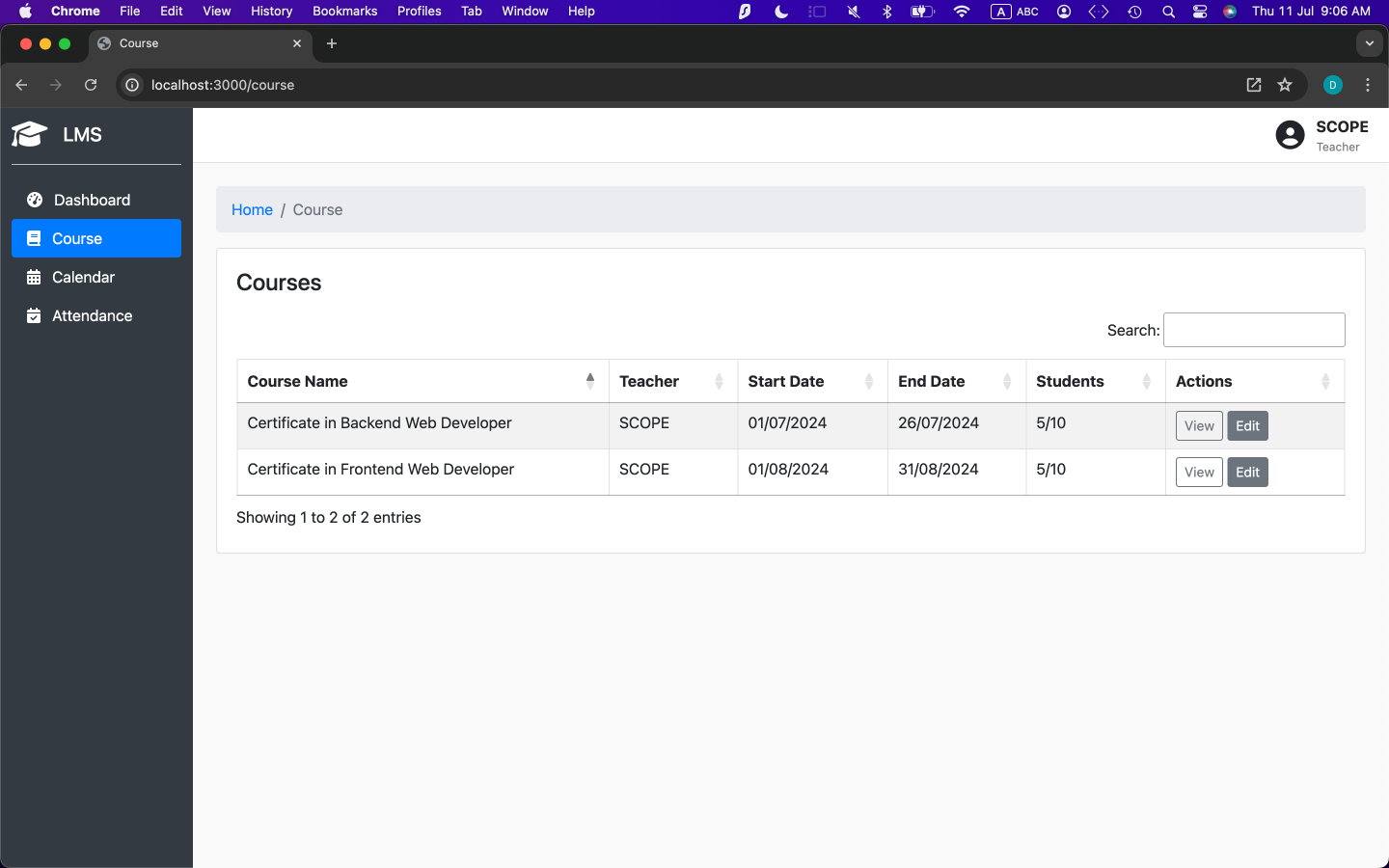


Figure 21. Teacher View - Dashboard

Figure 22. Teacher View - Teached Course List

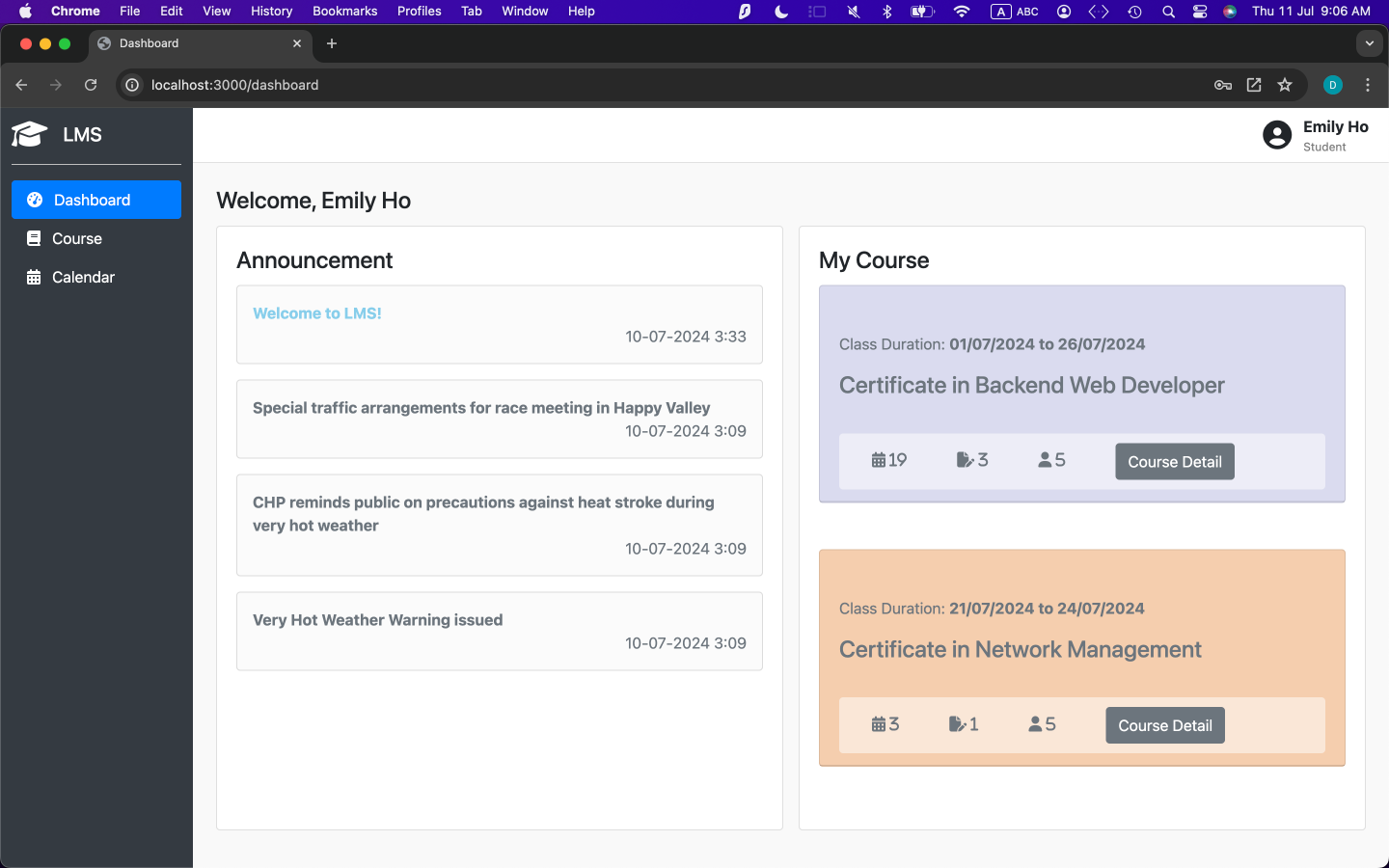


Figure 23. Student View - Dashboard

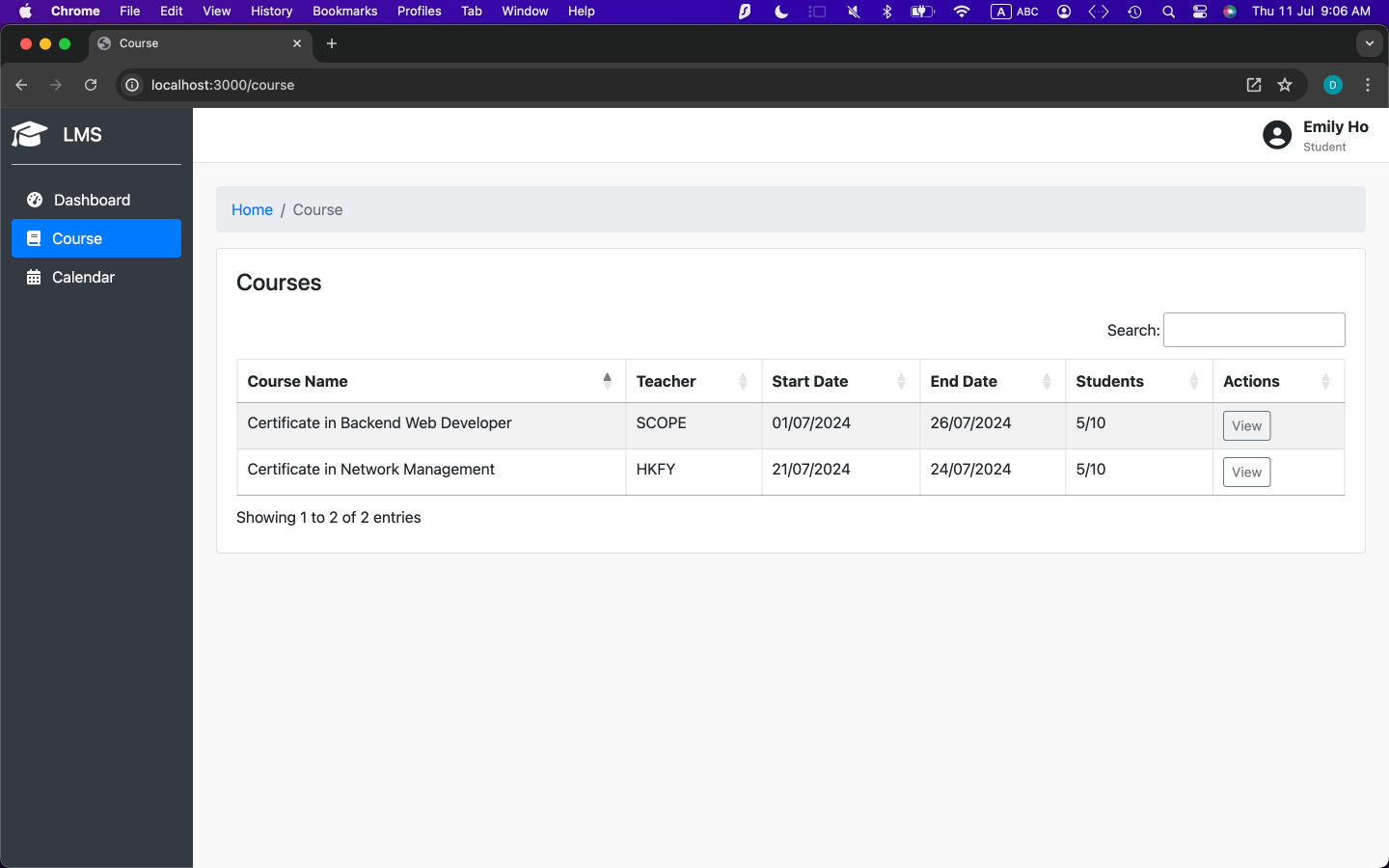


Figure 24. Student View - Enrolled Course List

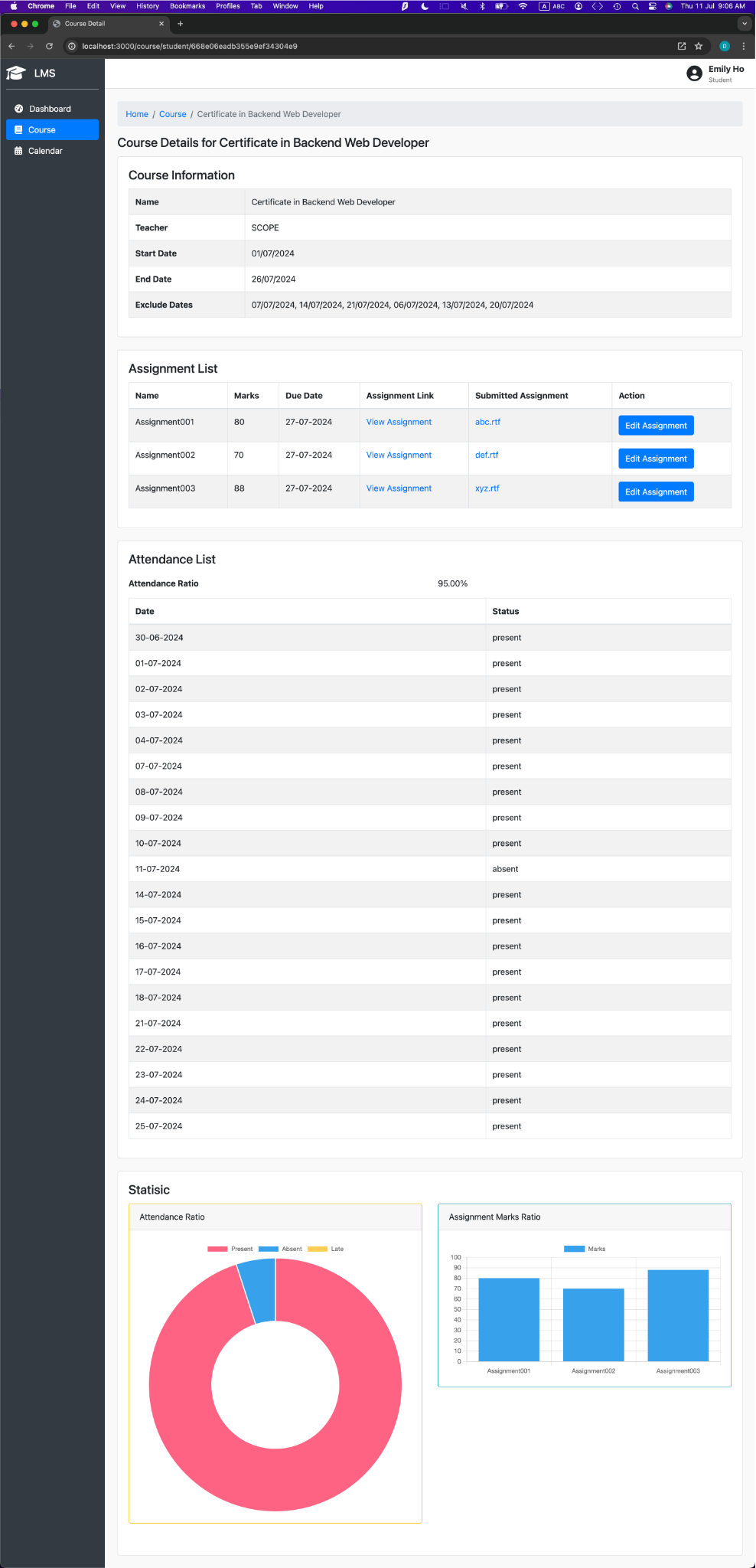
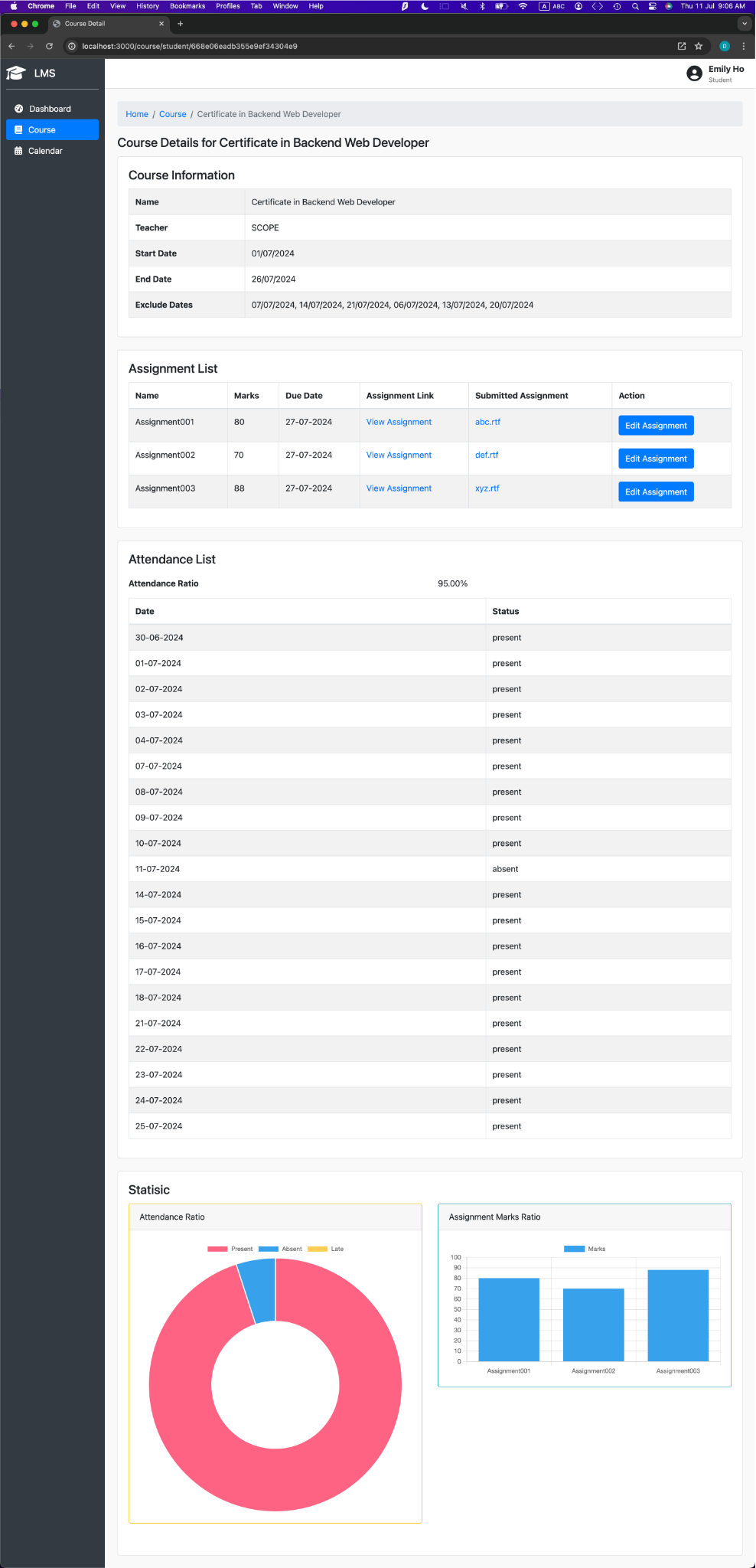


Figure 25. Student View - Enrolled Course Details

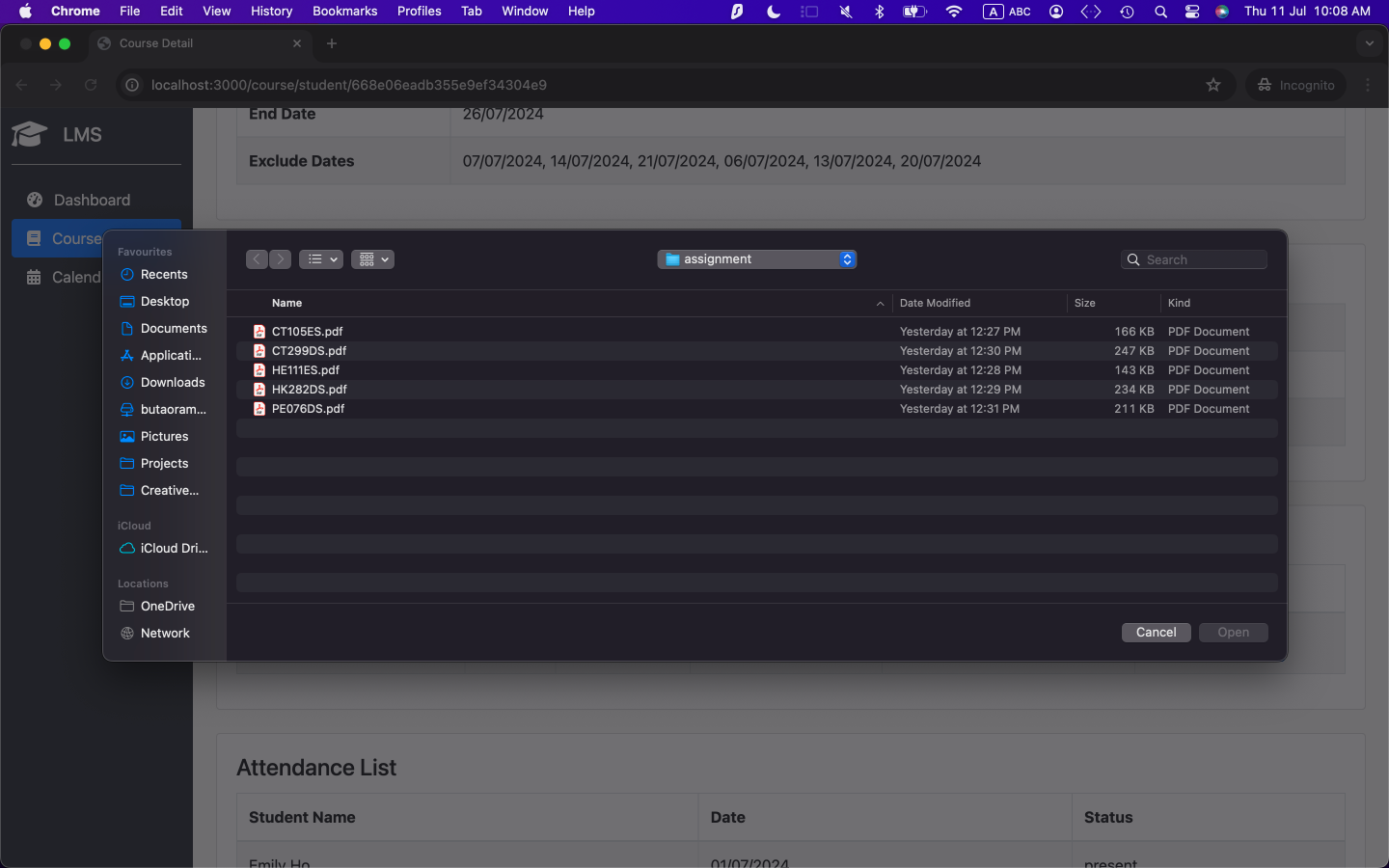


Figure 26. Student View - Course Assignment Upload