**LMS**

**Department of Software Engineering, San Jose State University**

**CMPE 202: System Software Engineering**

**Professor Gopinath Vinod**

**Teaching Assistant Siddanth Nagpal**

**May 8, 2024**

| **Recursive Rebels - Team Members** | |
| --- | --- |
| Lekhana Gadde | lekhana.gadde@sjsu.edu |
| Shireesh | shireesh.vennamaneni@sjsu.edu |
| Ashish | satyaashish.veda@sjsu.edu |
| Vinay | srivinay.appari@sjsu.edu |

# Table of Contents

**Chapter 1 Introduction**

1.1 Abstract

1.2 XP Core Values

**Chapter 2 Project Requirements**

2.1 Functional Requirements

2.2 Student Requirements

2.3 Administrator Requirements

**Chapter 3 System Design and Architecture**

3.1 System Overview and High-Level Design

3.2 UML Diagram

3.3 Component / Activity Diagram

3.6 Technology and resource requirements

**Chapter 4 Application Functionality**

4.1. User Screenshots

4.2. Faculty Screenshots

4.3 Admin Screenshots

**Chapter 5 AWS Implementation**

5.1. AWS Architecture

**Chapter 6 Scrum reports**

6.1 Scrum reports (backlogs)

**Chapter 7 Conclusion**

7.1 Links and References

**Abstract :**

This document outlines the development of a Learning Management System (LMS) designed to enhance educational experiences through a user-friendly platform. Learning management system is designed to manage faculties, students, and administrators based on the course management, content delivery, assessment and communication facilities according to system needs. The report defines the system under consideration, how the project was initiated, the development life cycle, architecture, implementation strategies, testing methodologies, deployment plans.

**XP Core Values**

Throughout the project, our team adhered to the XP Core Values of **Communication and Feedback**, which were essential for collaboration, continuous improvement, and shared understanding among team members.

**Communication:**

We placed a strong emphasis on maintaining open and consistent communication channels to ensure the seamless flow of information within the team. Our primary modes of communication included video conferencing platforms like Zoom and Google Meet, where we conducted regular meetings to discuss project progress, share updates, and address any challenges or concerns that arose. These meetings provided a platform for transparent and open dialogue, allowing team members to express their ideas, seek clarification, and provide feedback. Additionally, we utilised collaboration tools like whatsapp groups and email for quick updates and asynchronous discussions. By maintaining clear and diverse communication channels, we were able to stay aligned with project goals, promptly resolve issues, and ensure that everyone was well-informed and engaged throughout the project lifecycle.

**Constructive Feedback**:

Feedback was a cornerstone of our team's success, as it enabled us to continuously improve and adapt our approach. We encouraged an environment where team members felt comfortable providing and receiving feedback in an open and respectful manner. During our meetings, we actively solicited feedback on project progress, ideas and proposed solutions, encouraging everyone to share their perspectives and suggestions for improvement. Furthermore, we conducted regular retrospectives to reflect on our processes, identify areas for growth and discuss strategies for enhancement. By actively seeking and incorporating feedback into our work, we were able to iteratively refine our approach, address challenges effectively and deliver a high-quality solution that met the project requirements and exceeded expectations.

1. **User Roles and Access Control:**

* The system will support three distinct user roles: Faculty, Students, and Administrators.
* Each role will have designated permissions and access levels, ensuring that users can only perform actions relevant to their role.

2. **Homepage:**

* Faculty members will have a personalized homepage displaying the courses they are teaching in the current and previous semesters.
* Courses listed will include both published and unpublished ones.
* Students and Administrators will also have dedicated homepages providing pertinent information and functionalities suited to their roles.

3. **Course Management:**

* Faculty will be able to enrich the syllabus section of each course with content.
* They will have access to features such as viewing student lists, assigning and viewing grades, adding assignments, quizzes, and making announcements.

4. **Enrollment:**

* Students will have visibility into the courses they are enrolled in for both the current and past semesters. (The enrollment process itself is considered external to the application.)

5. **Content Access:**

* Students can access course content provided the course is published.
* This content may include syllabi, assignments, quizzes, and announcements.

6. **Grades Management:**

* Students can review their grades for each course they are enrolled in.
* Faculty will have the capability to assign grades to students for each course they teach.

7. **Profile Management:**

* Students will be empowered to manage their profiles, including setting notification preferences.
* Administrators will have access to manage user profiles and system settings.

8. **Admin Functions:**

* Administrators can oversee courses taught by faculty for each semester.
* They are responsible for assigning courses to faculty members for upcoming semesters.
* Additionally, they can view student lists for each course, although grades will remain concealed.

9. A**PI and UI Integration:**

* The application will expose APIs to facilitate all functionalities listed above.
* The web or mobile UI will interact with these APIs to ensure a seamless user experience across all roles.

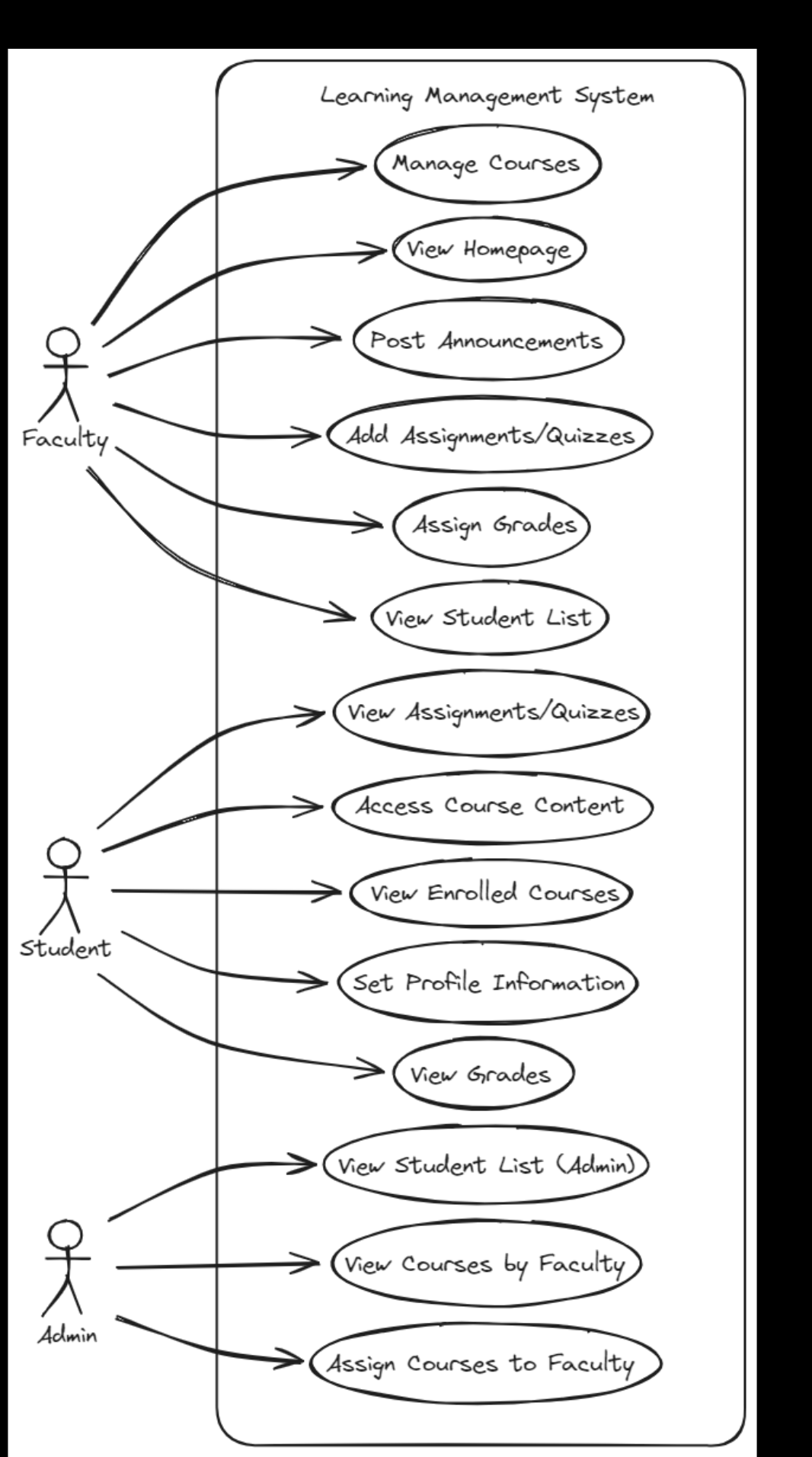
10. **Cloud Deployment:**

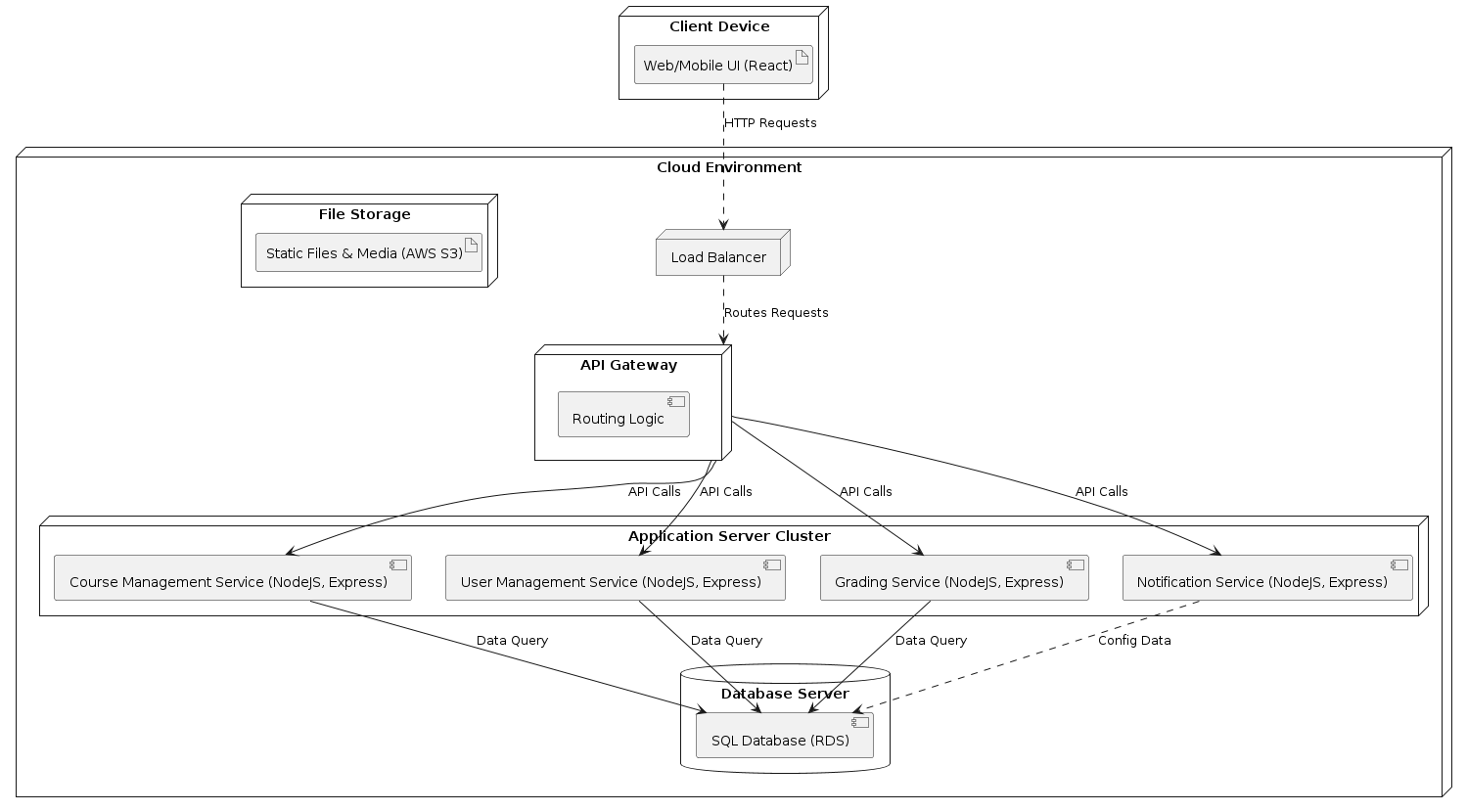
* Deployment will be on a cloud platform (e.g., AWS) utilizing an Auto Scaled EC2 Cluster with a Load Balancer to ensure scalability and reliability.

11. **Database and Mock Data:**

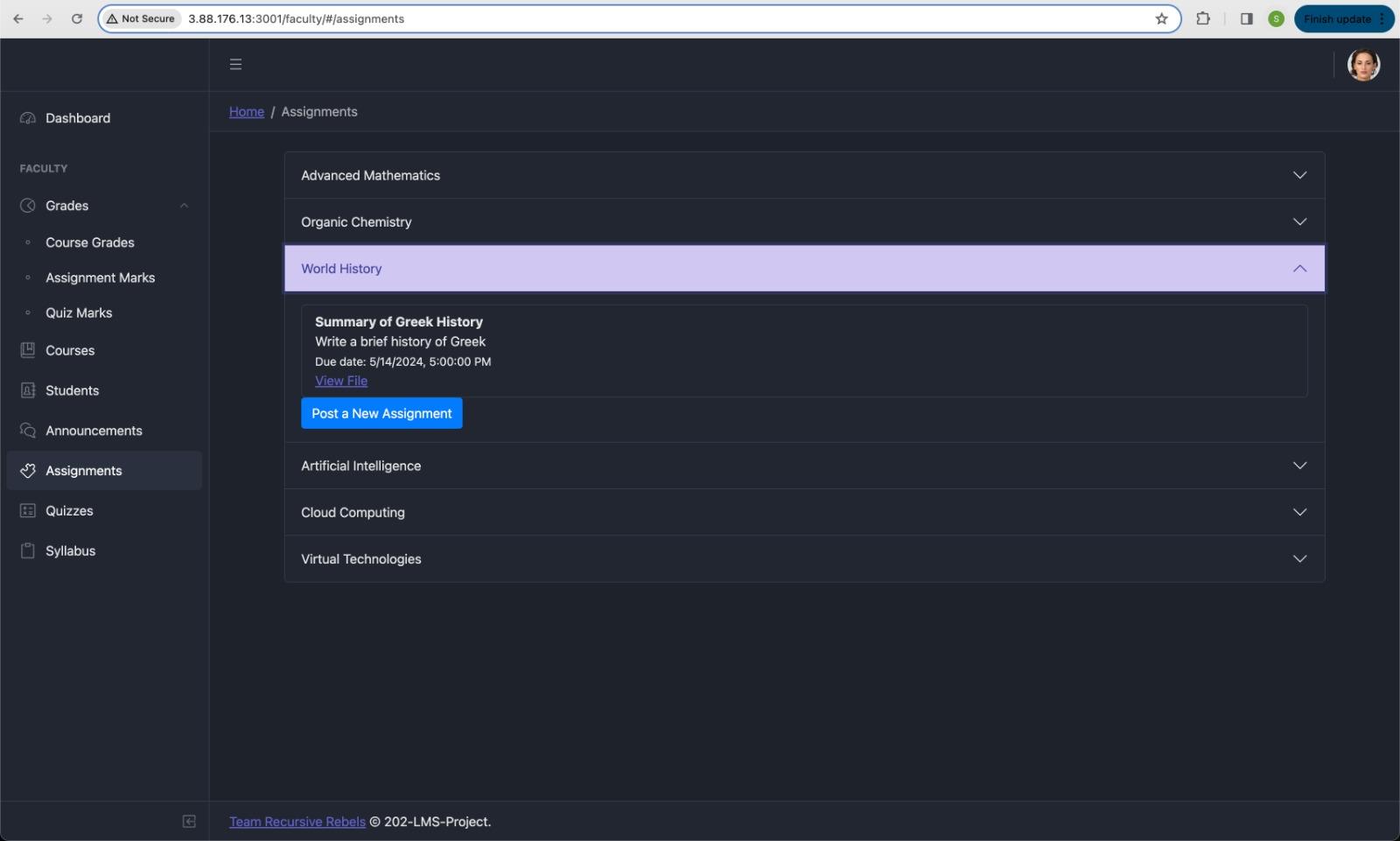
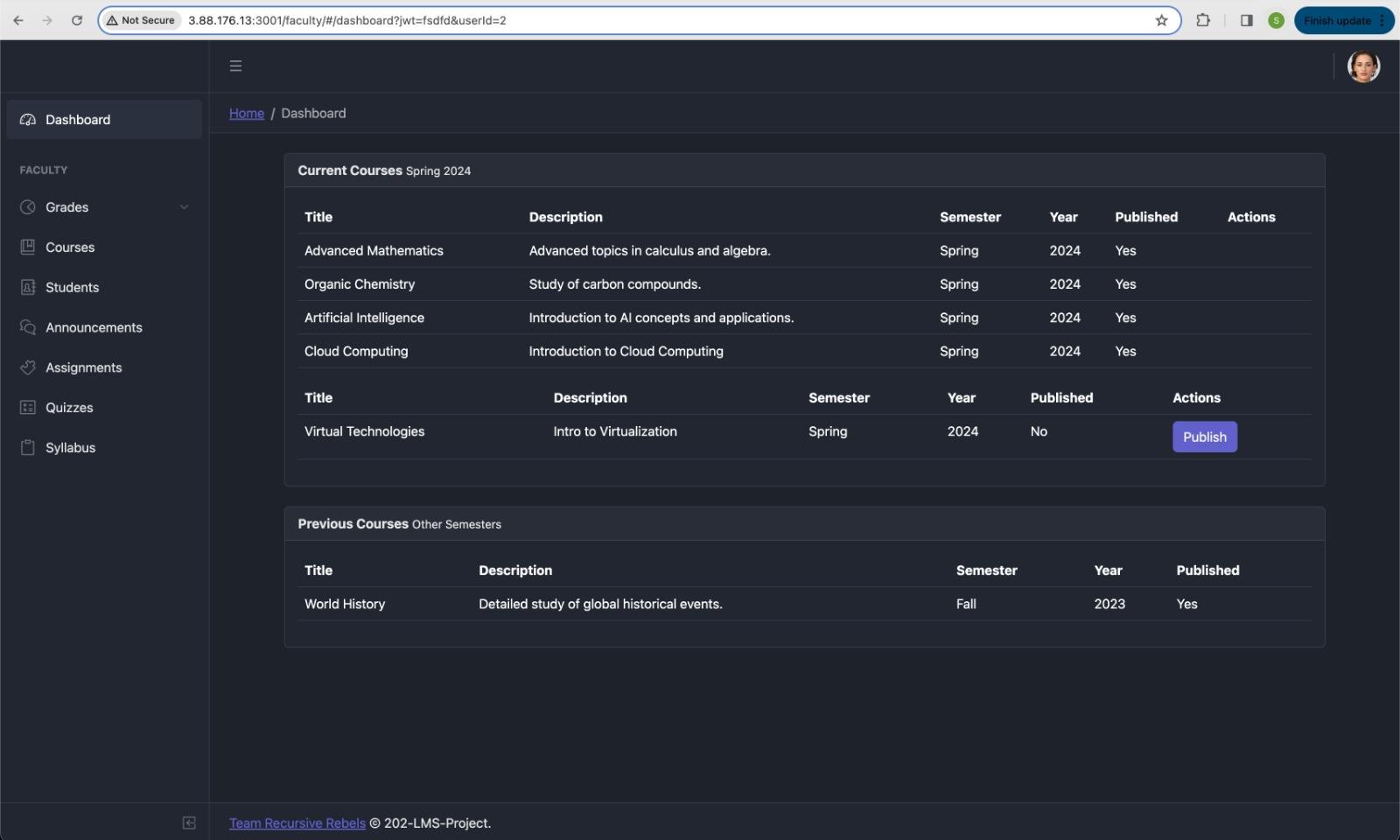
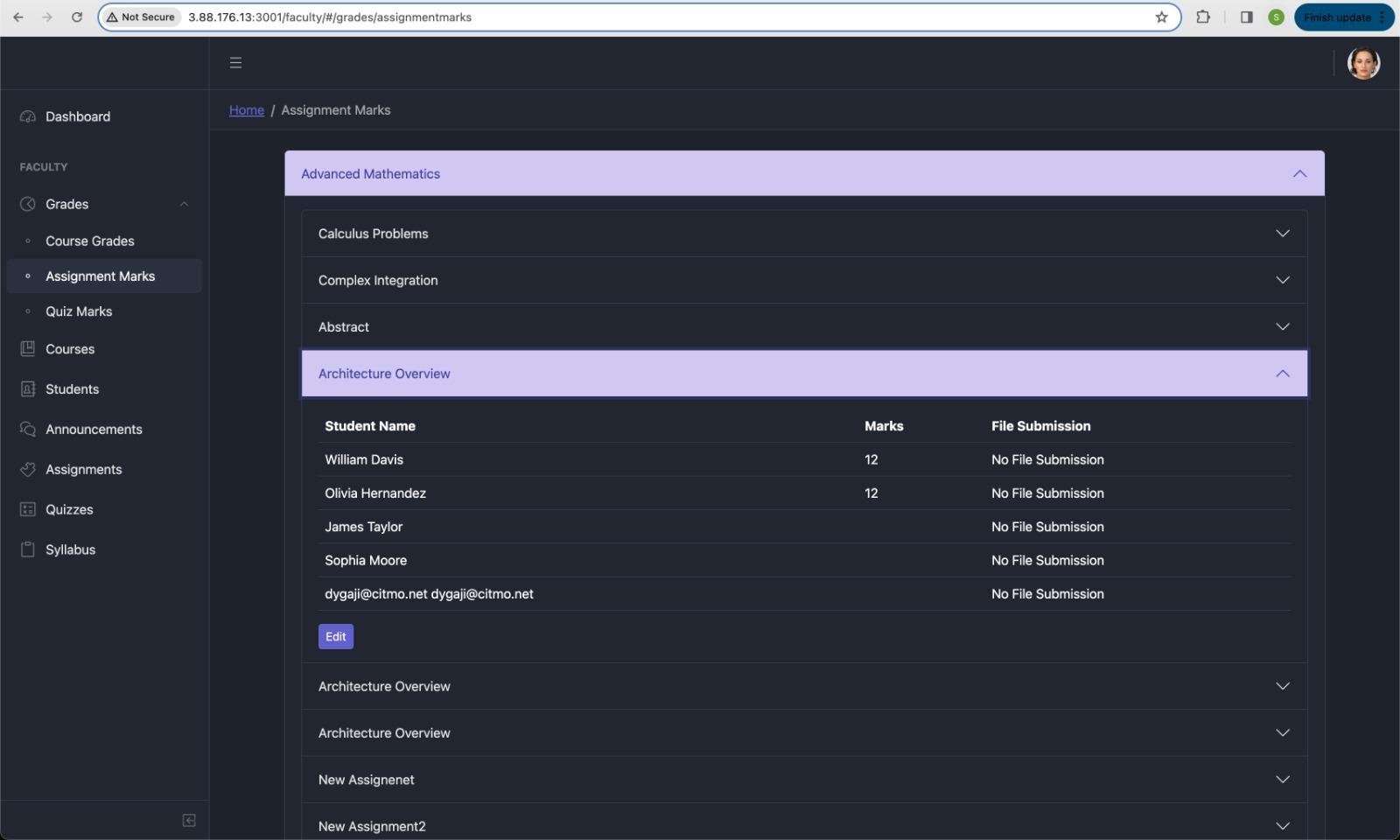
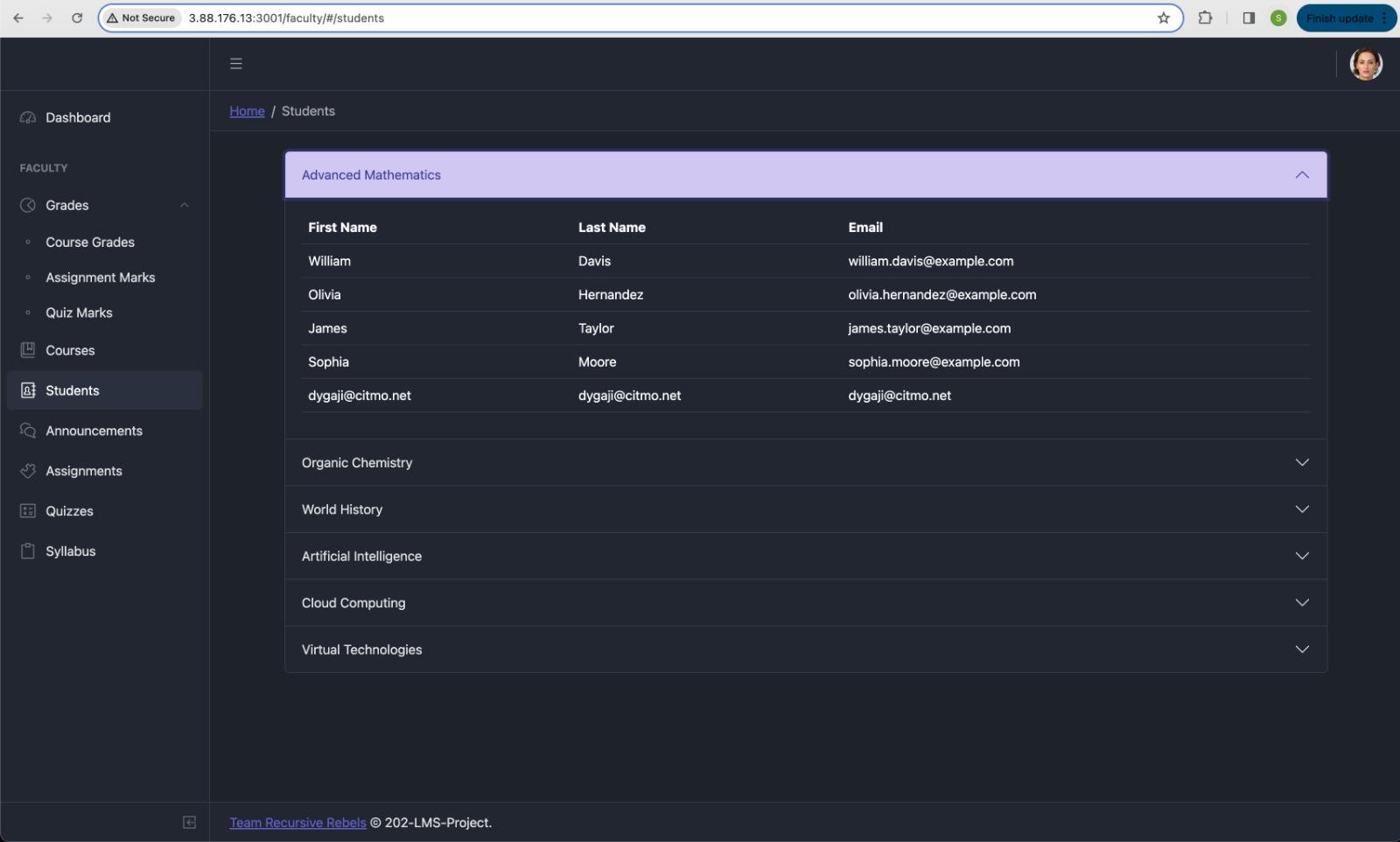
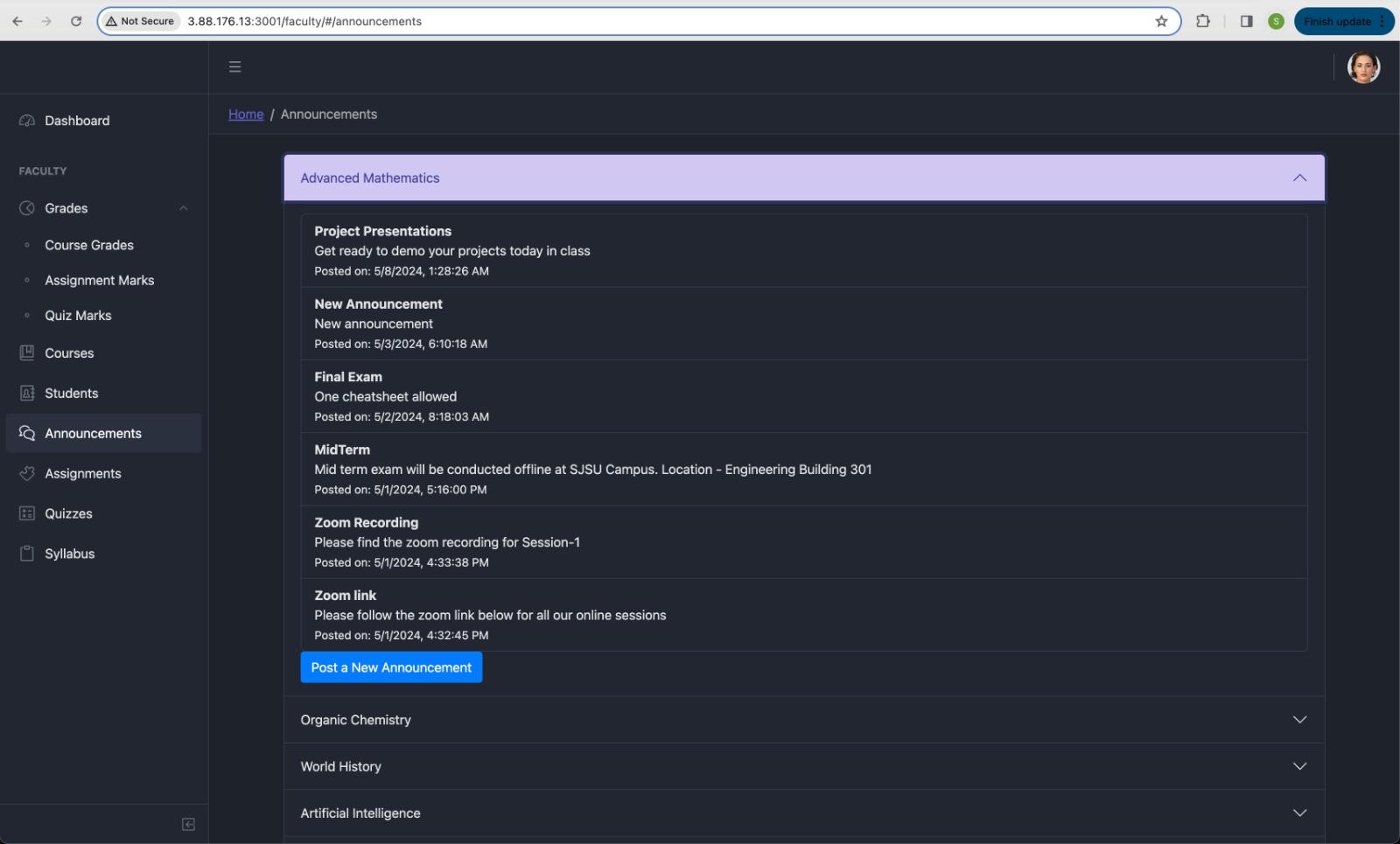
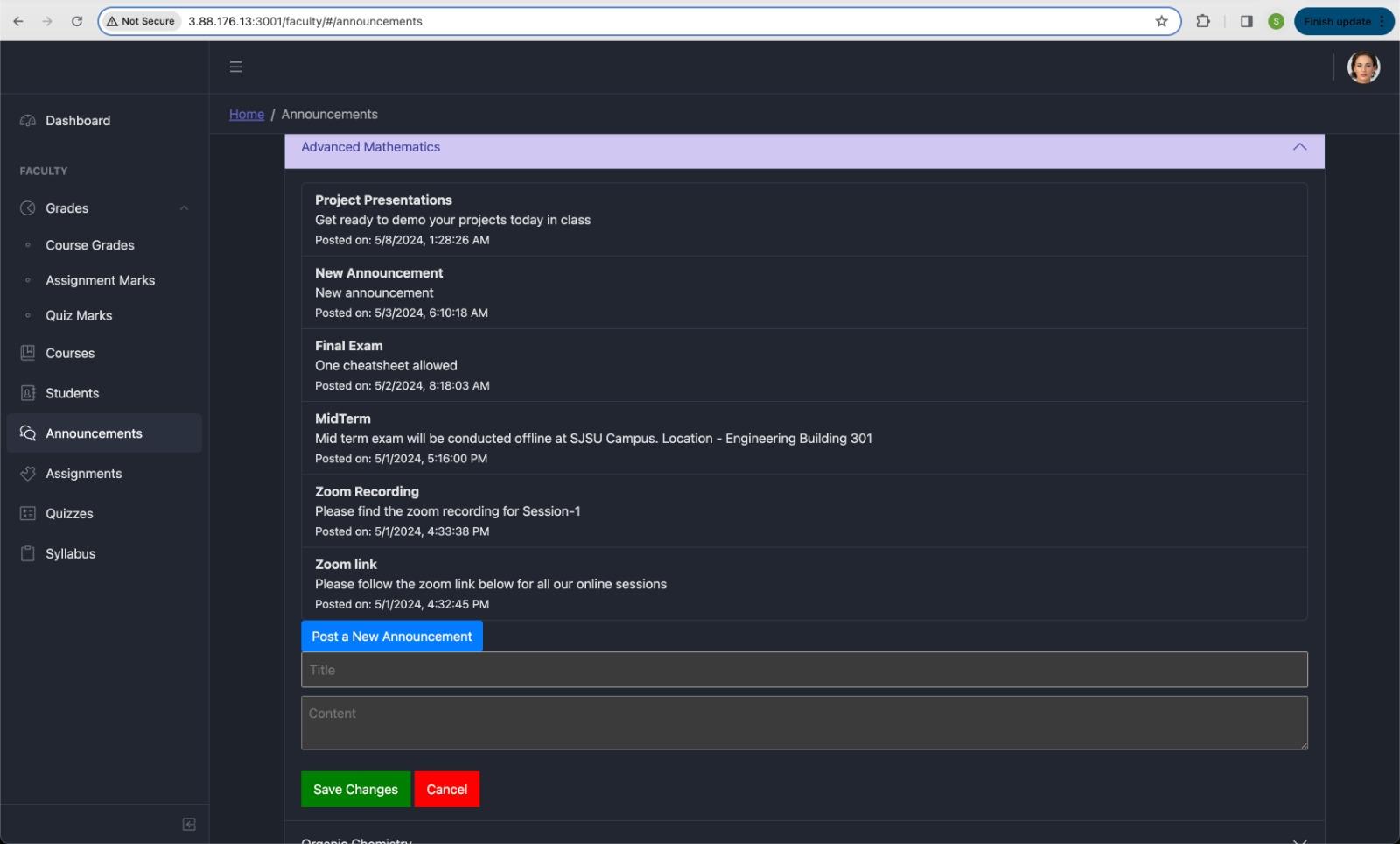
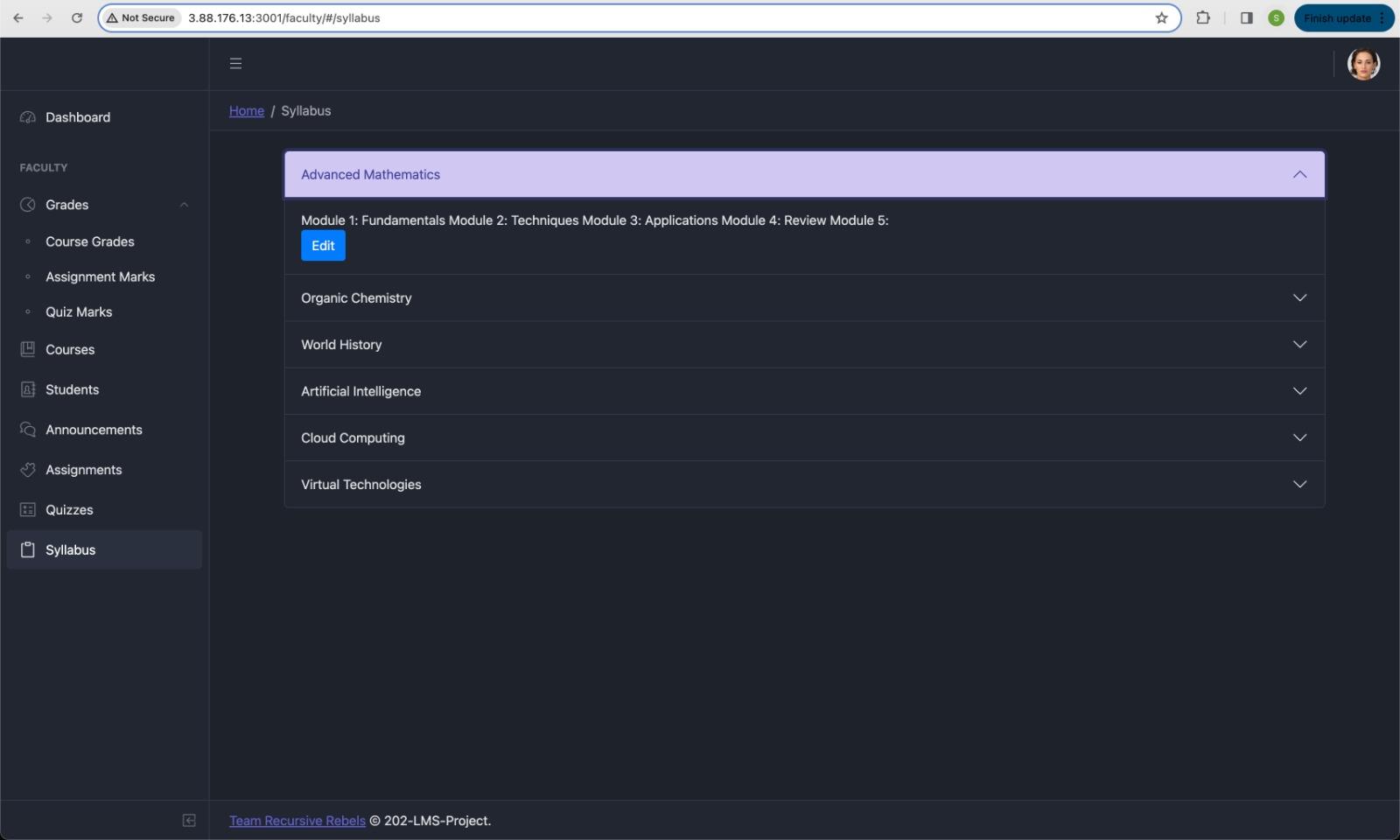
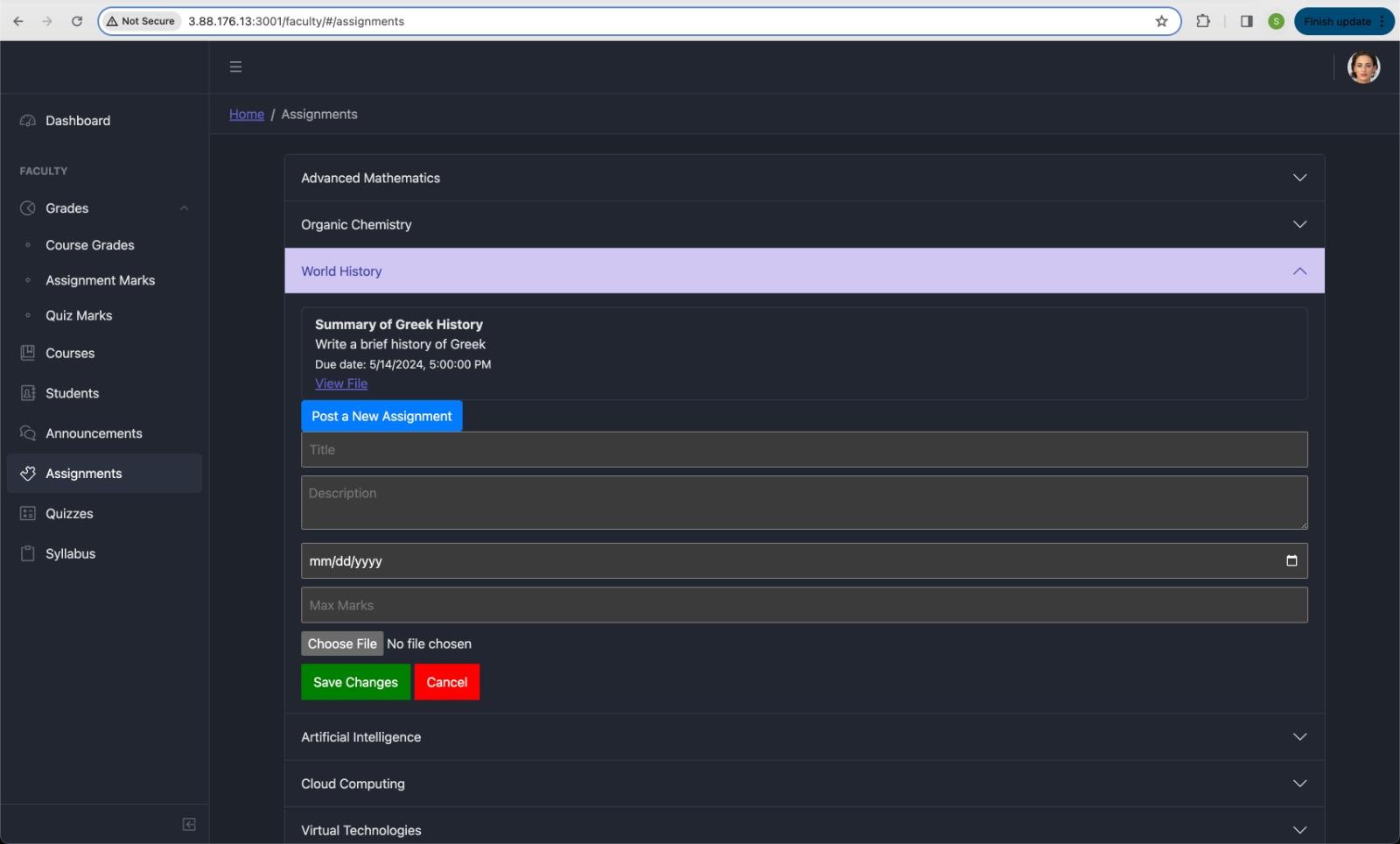
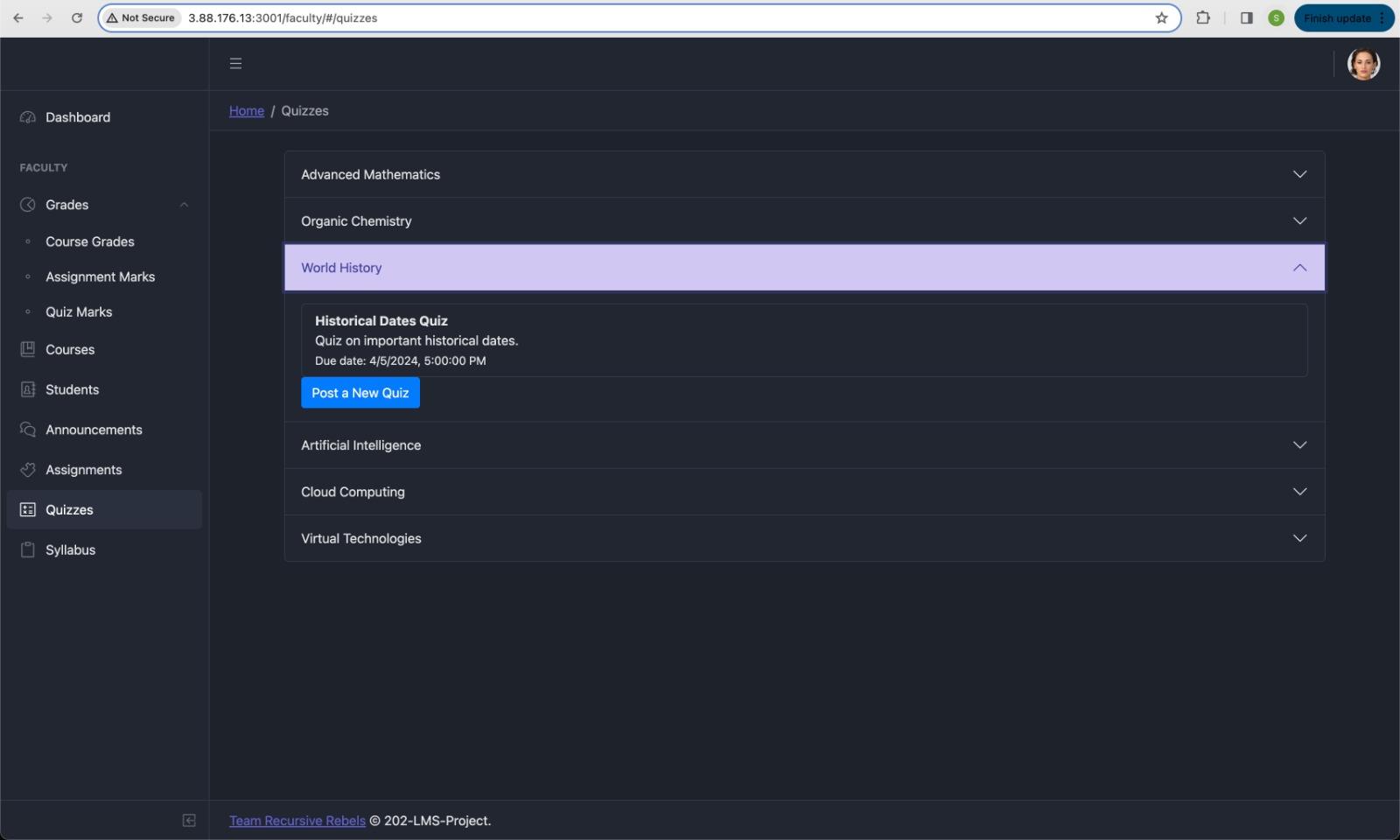
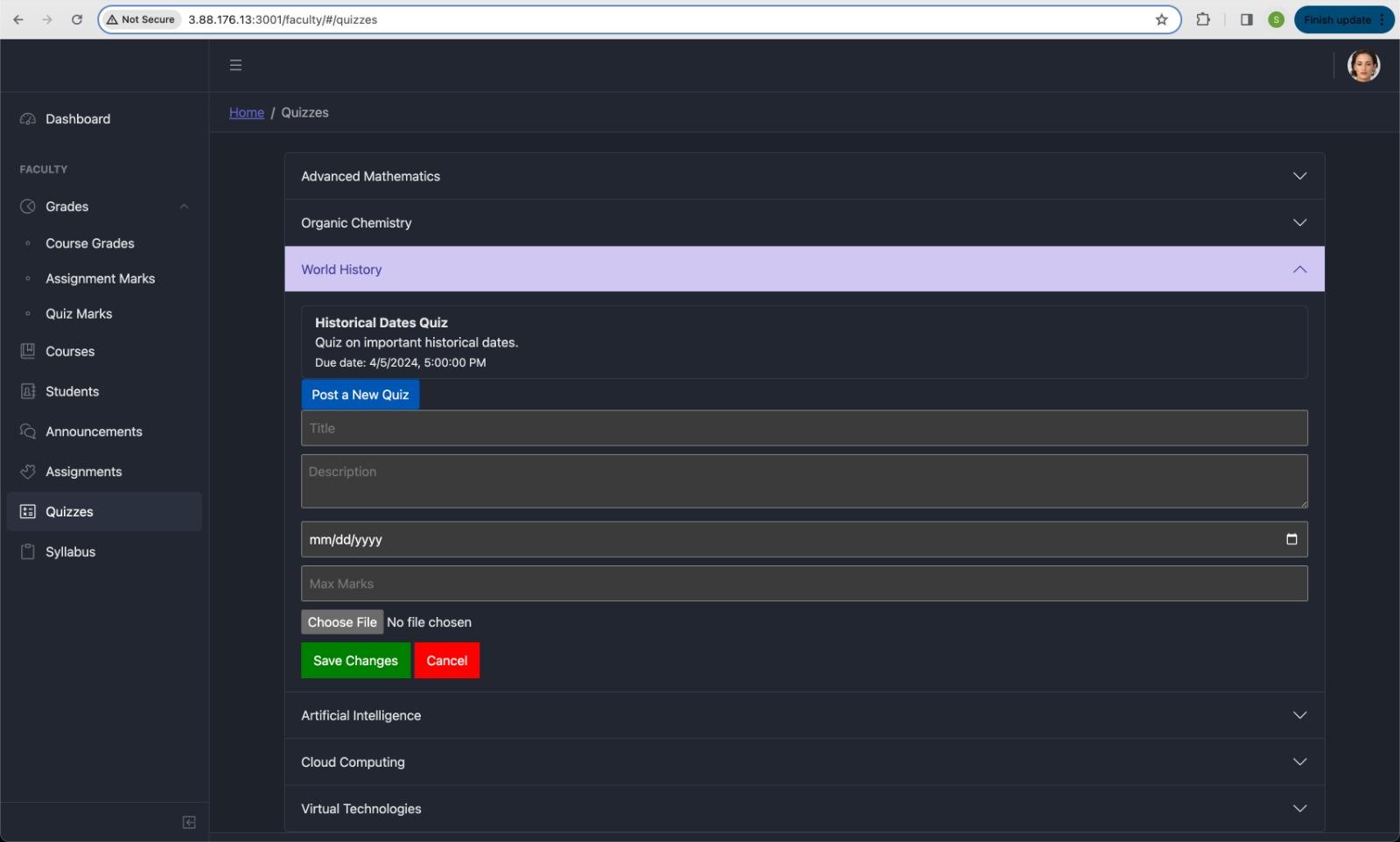
* A database will be employed to store pertinent information such as course details, faculty data, student records, and grades.
* Mock data will be provided for testing and development purposes, aiding in the creation of a robust and reliable system.

**4. Application Functionality :**

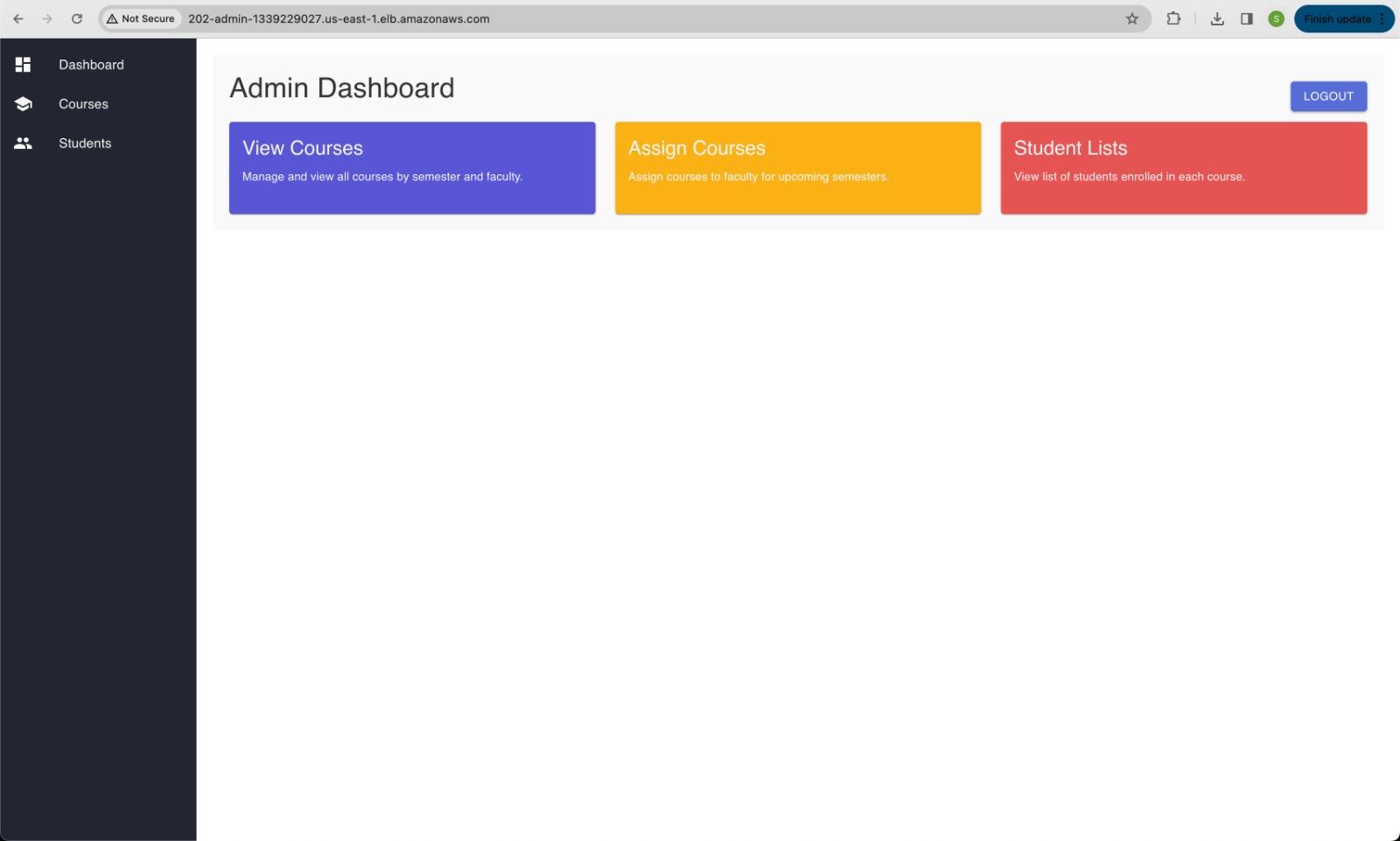


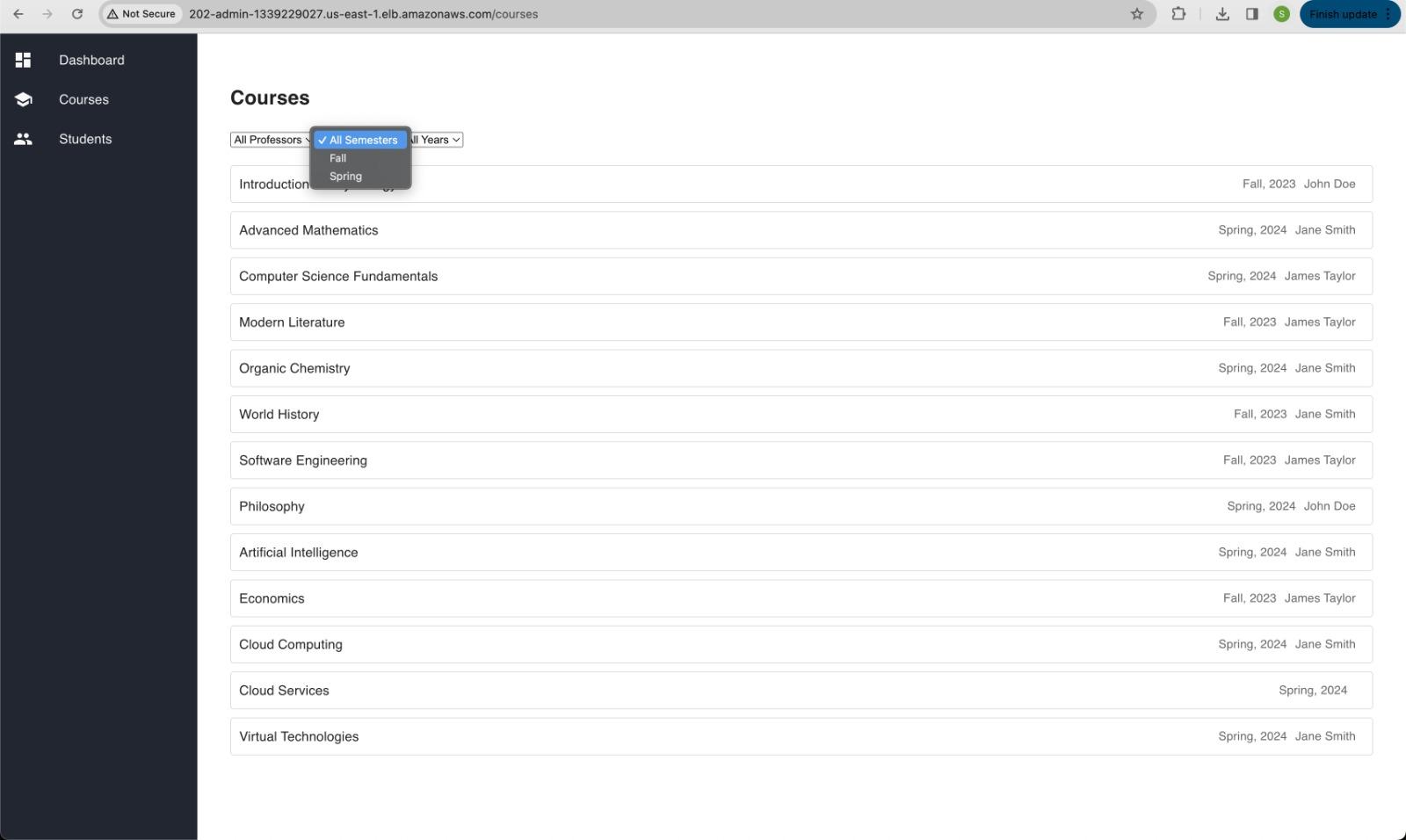
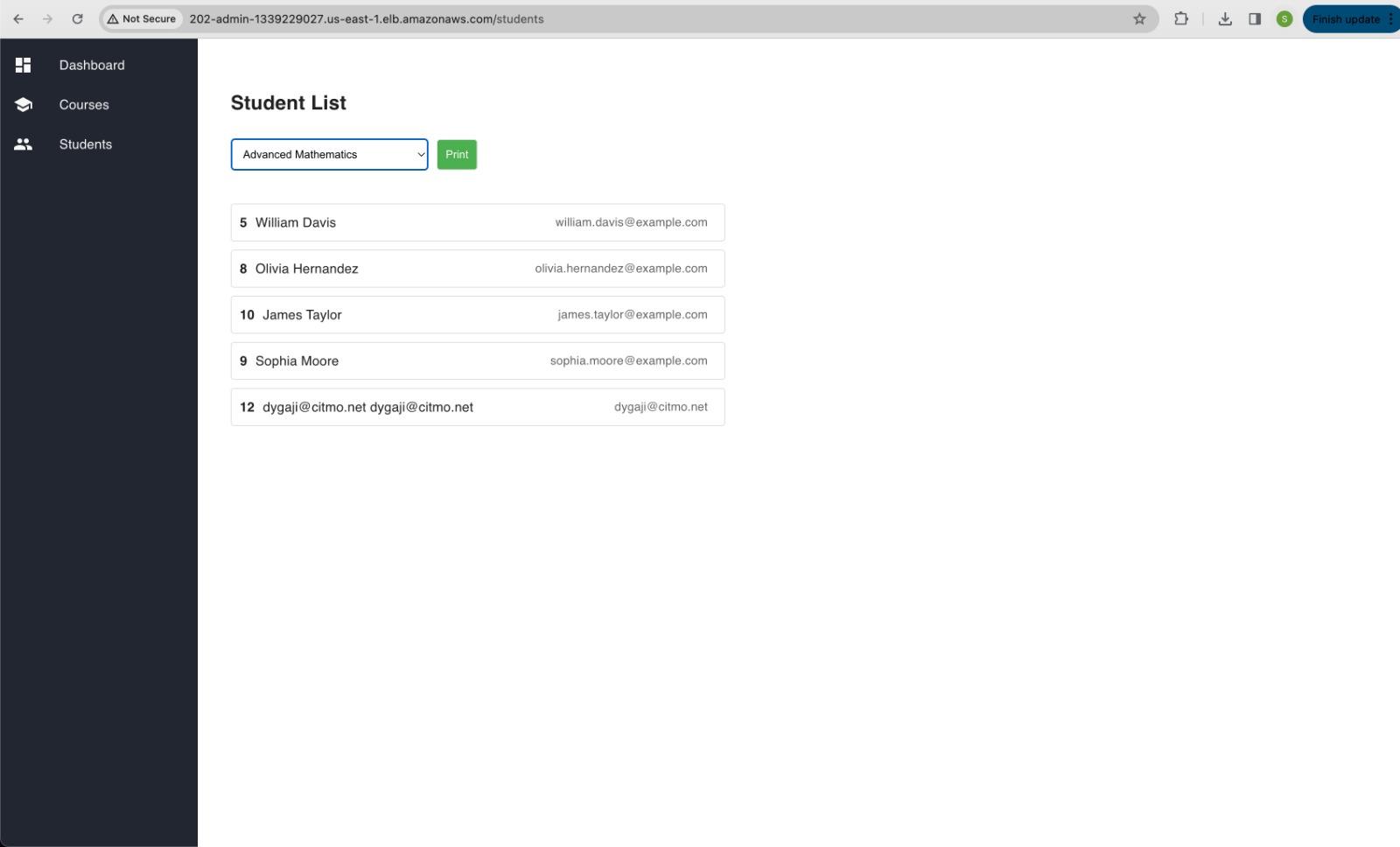
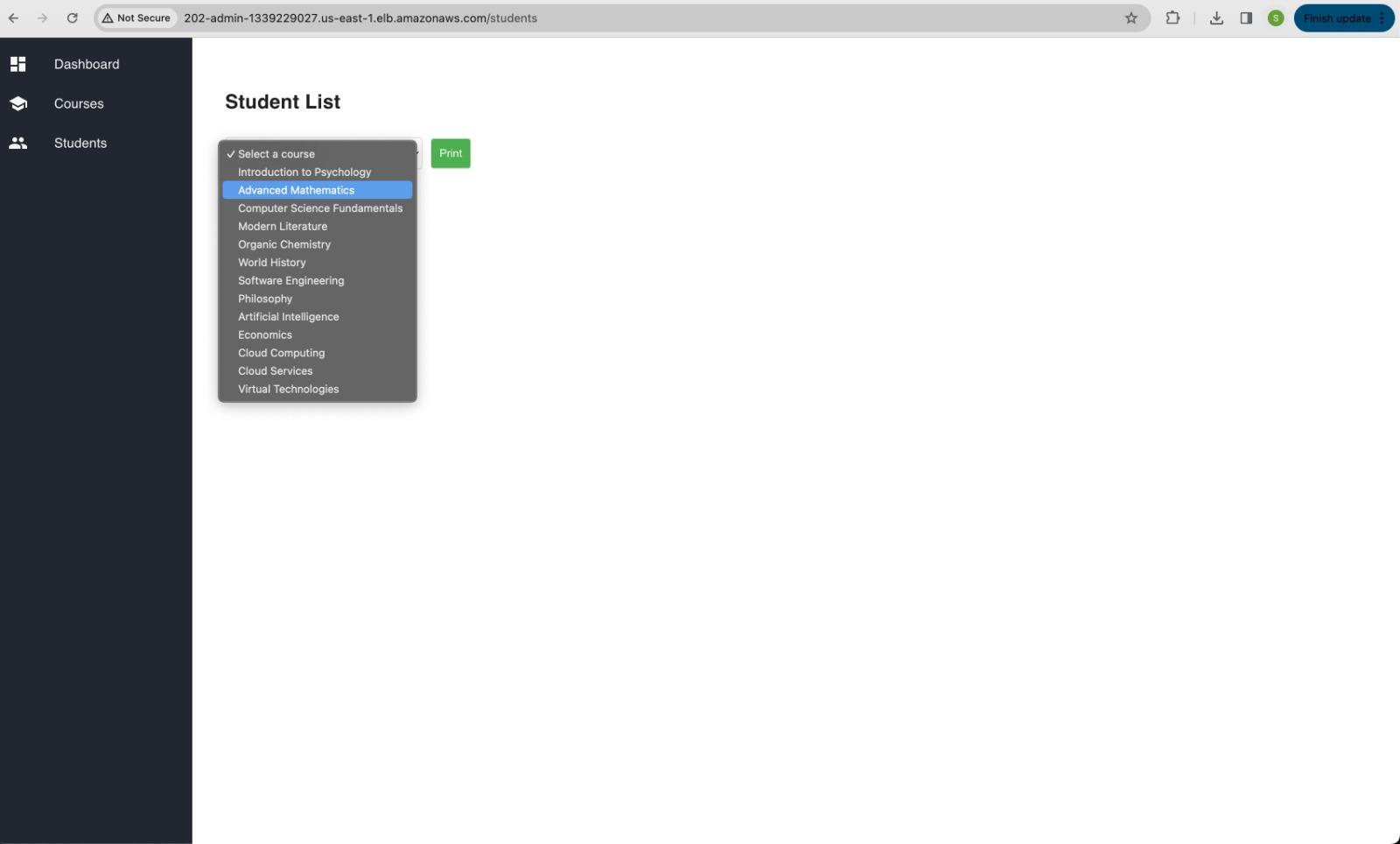


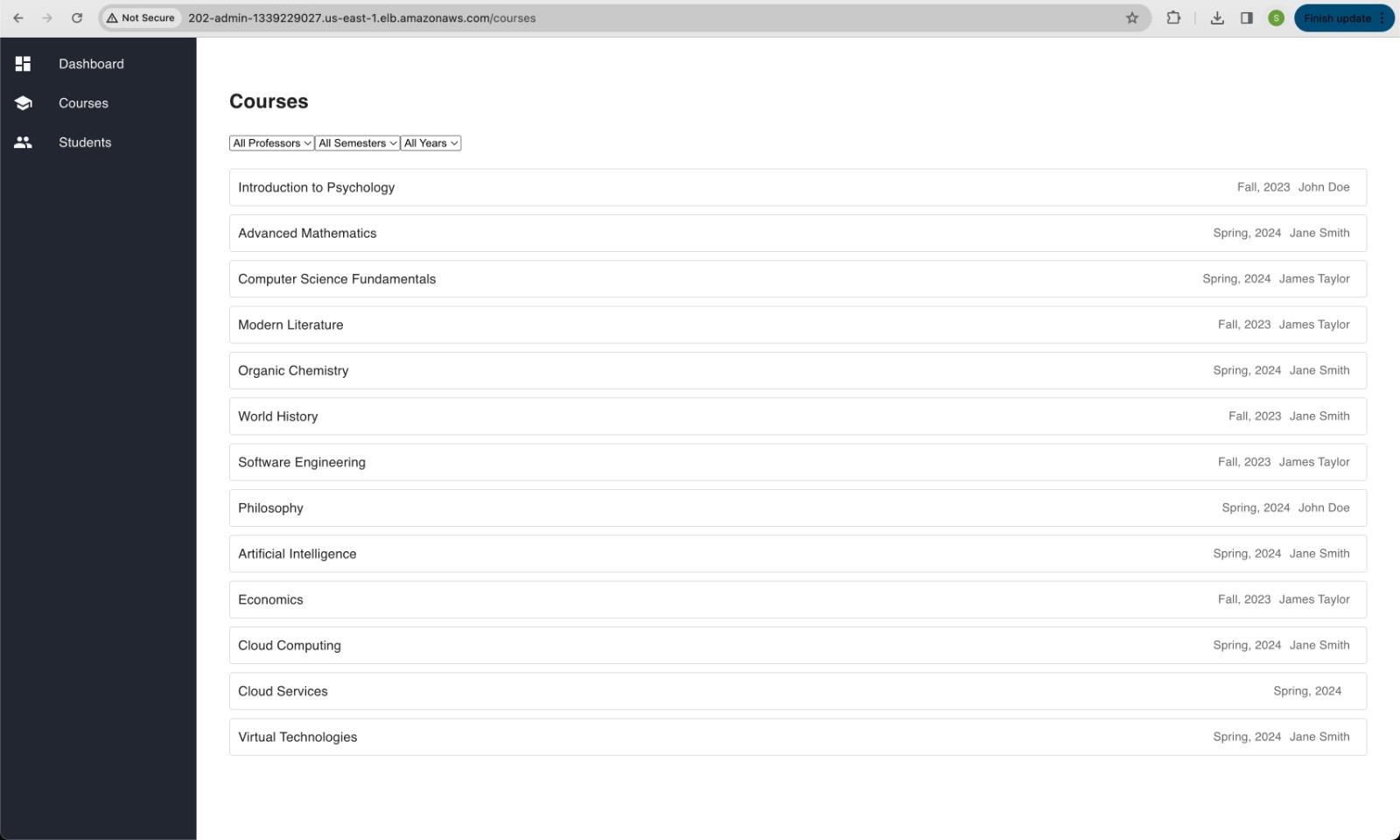
**Faculty Screen :**

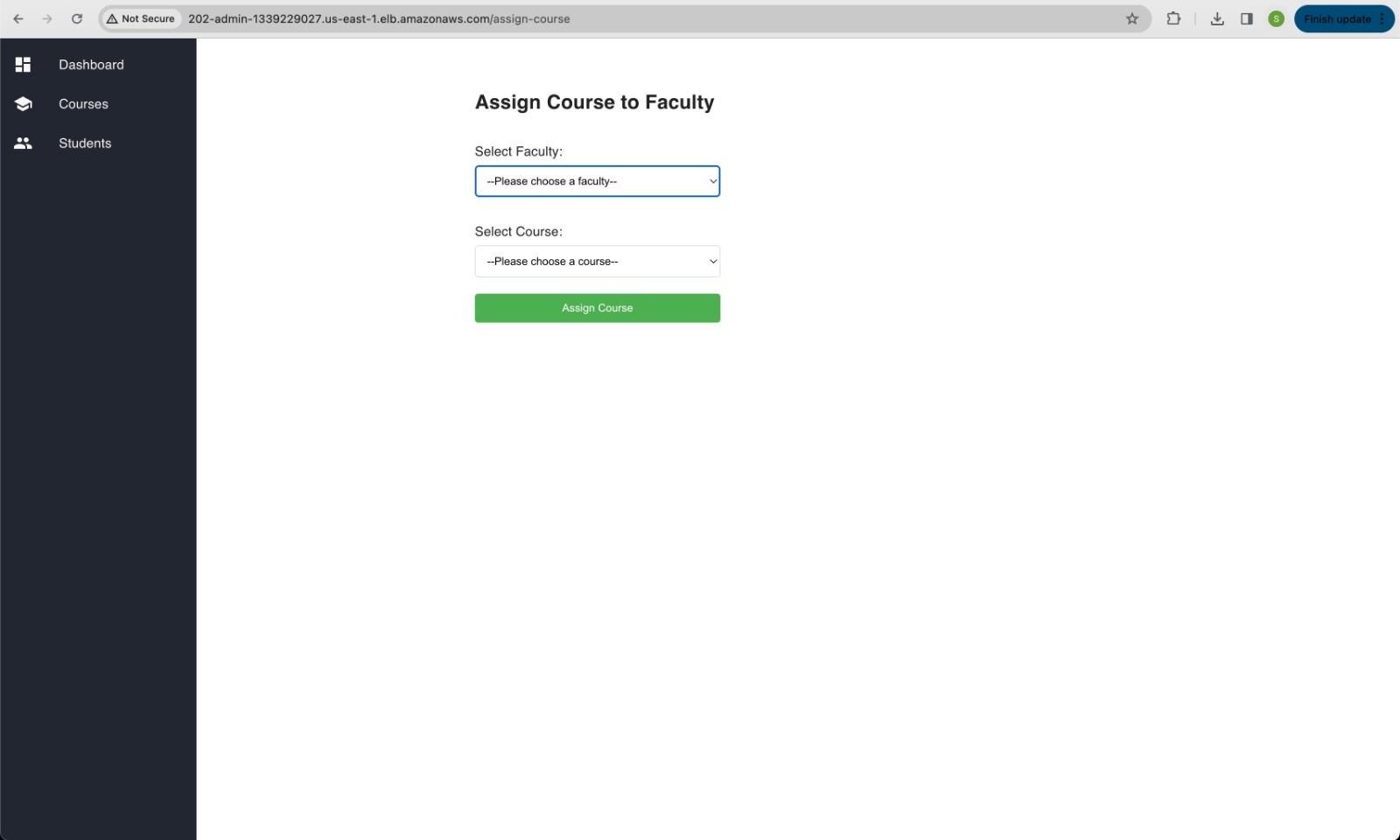


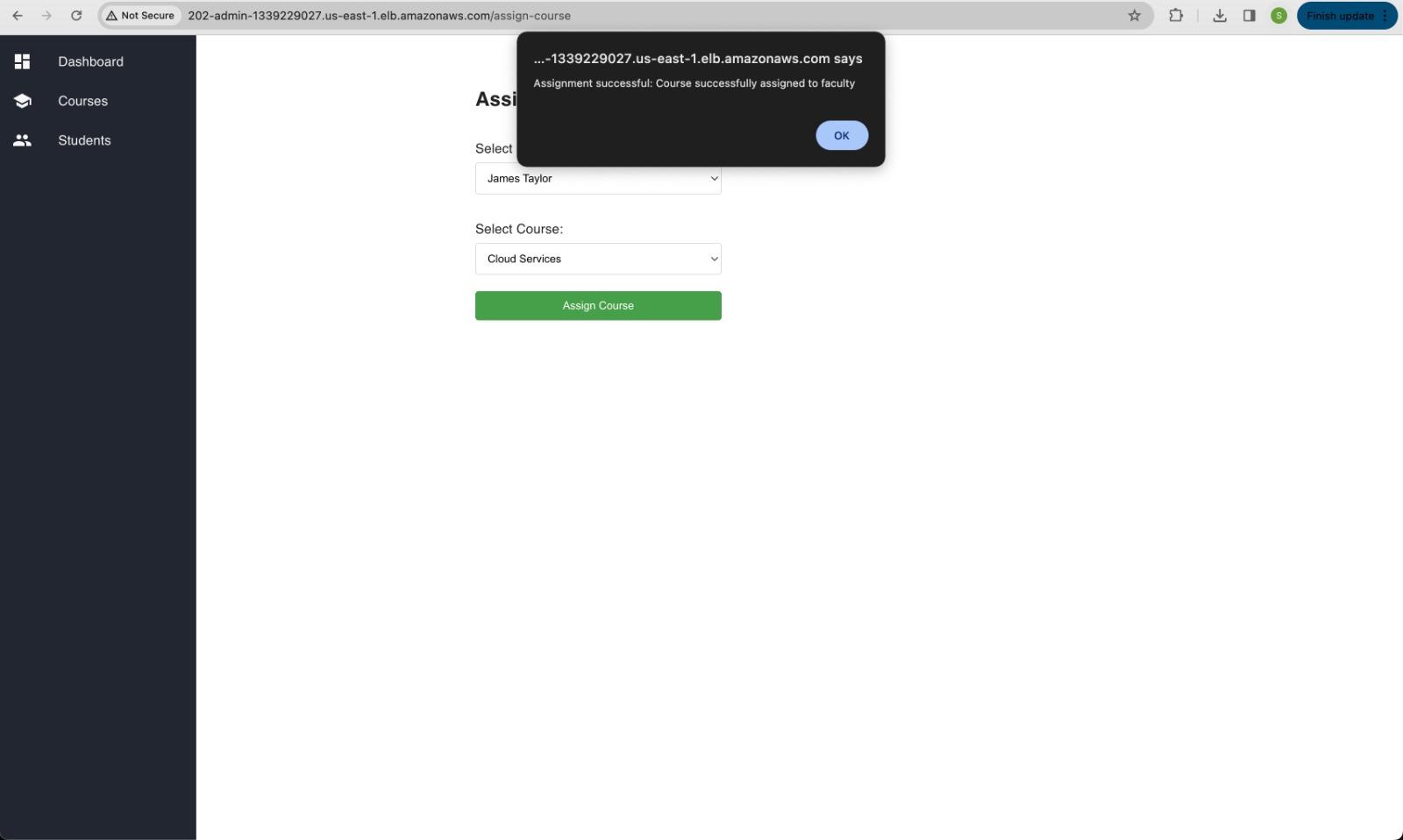
**Admin Screen**

****

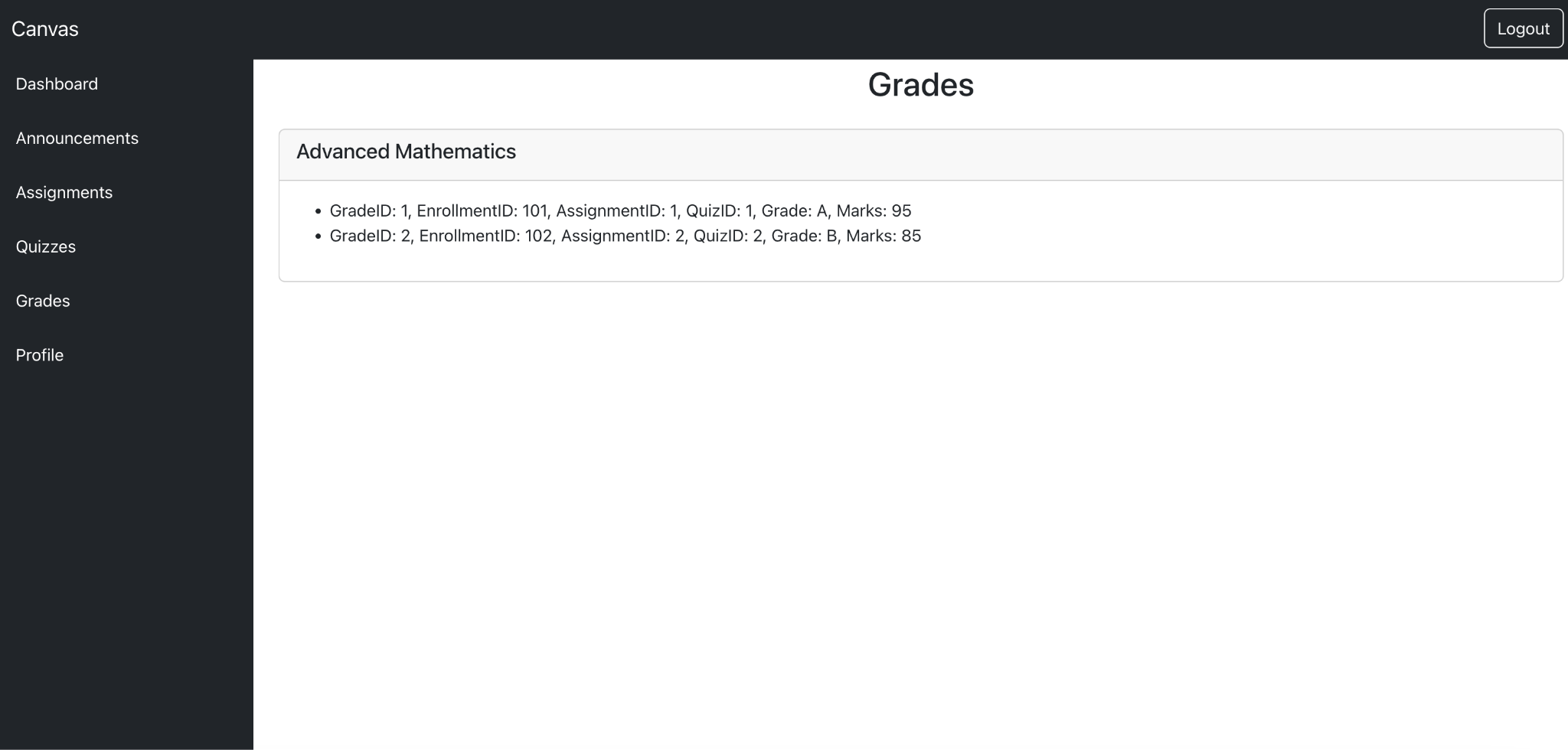
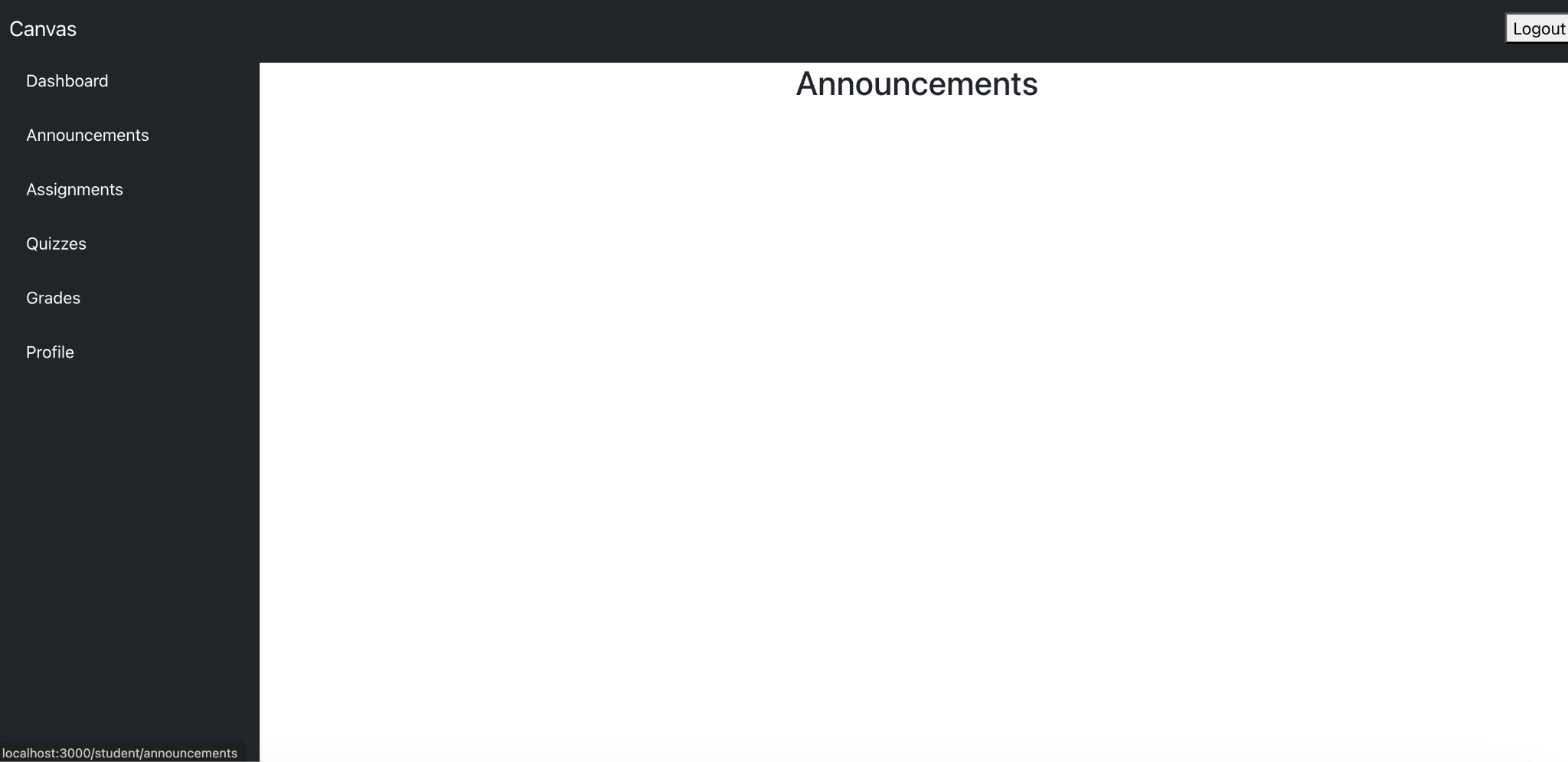
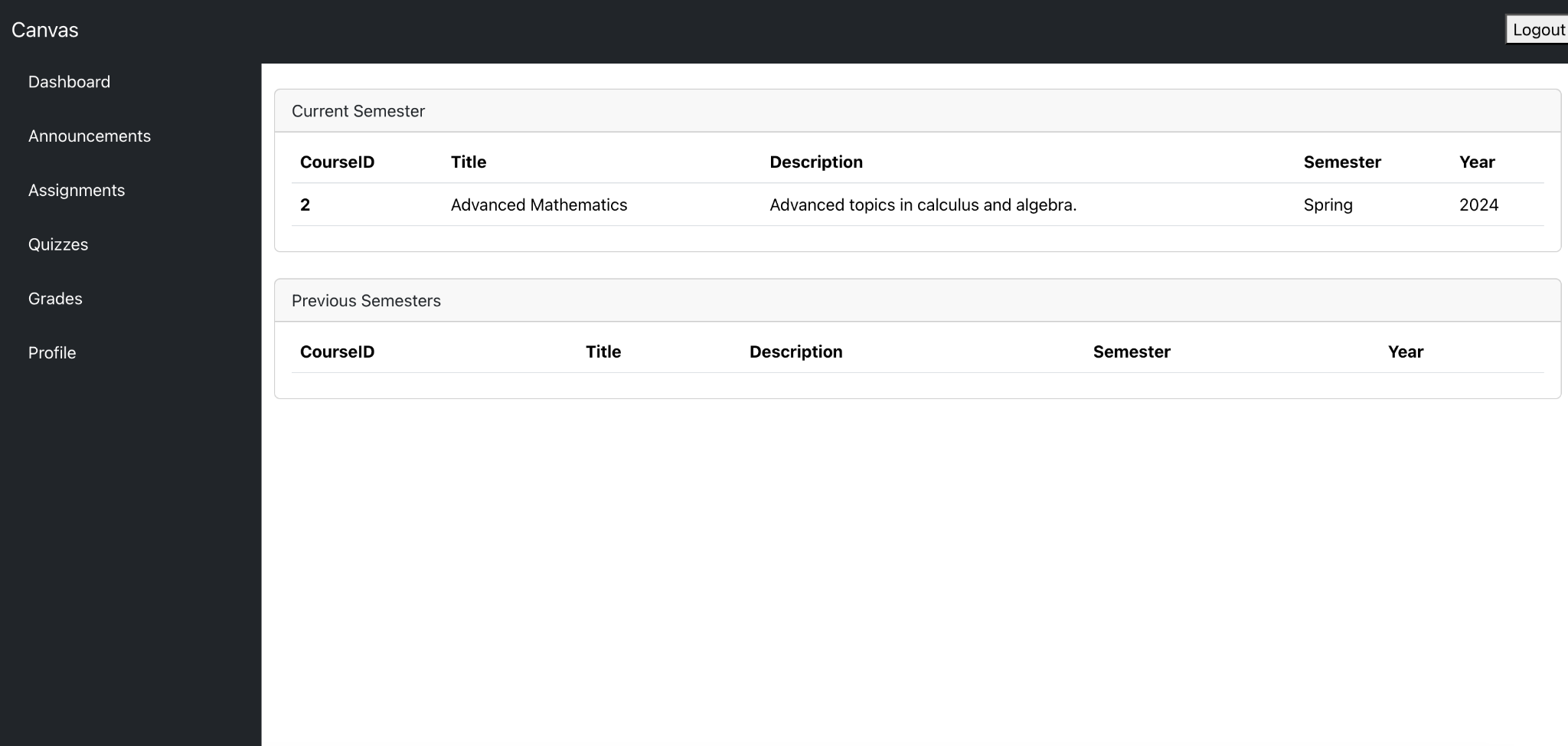
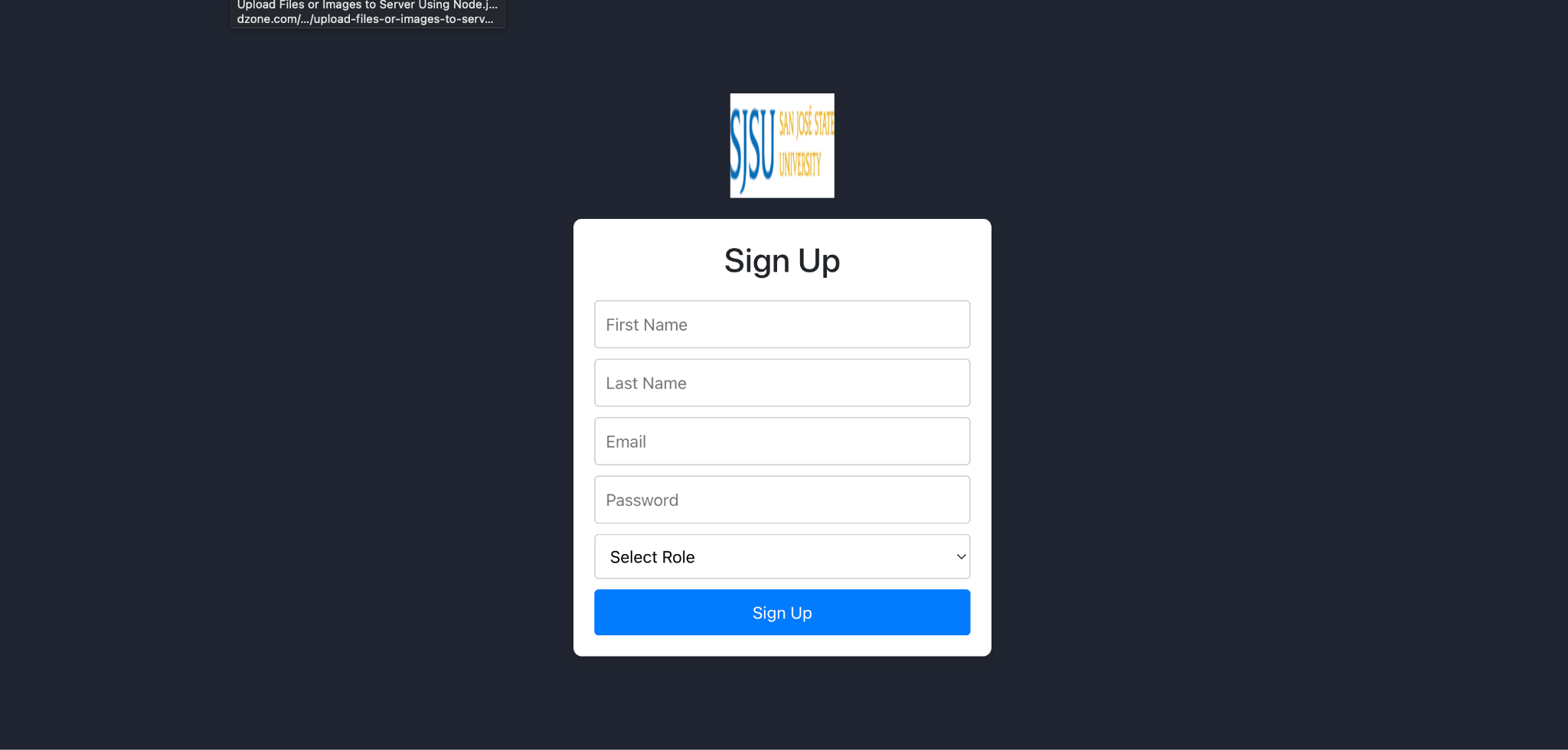
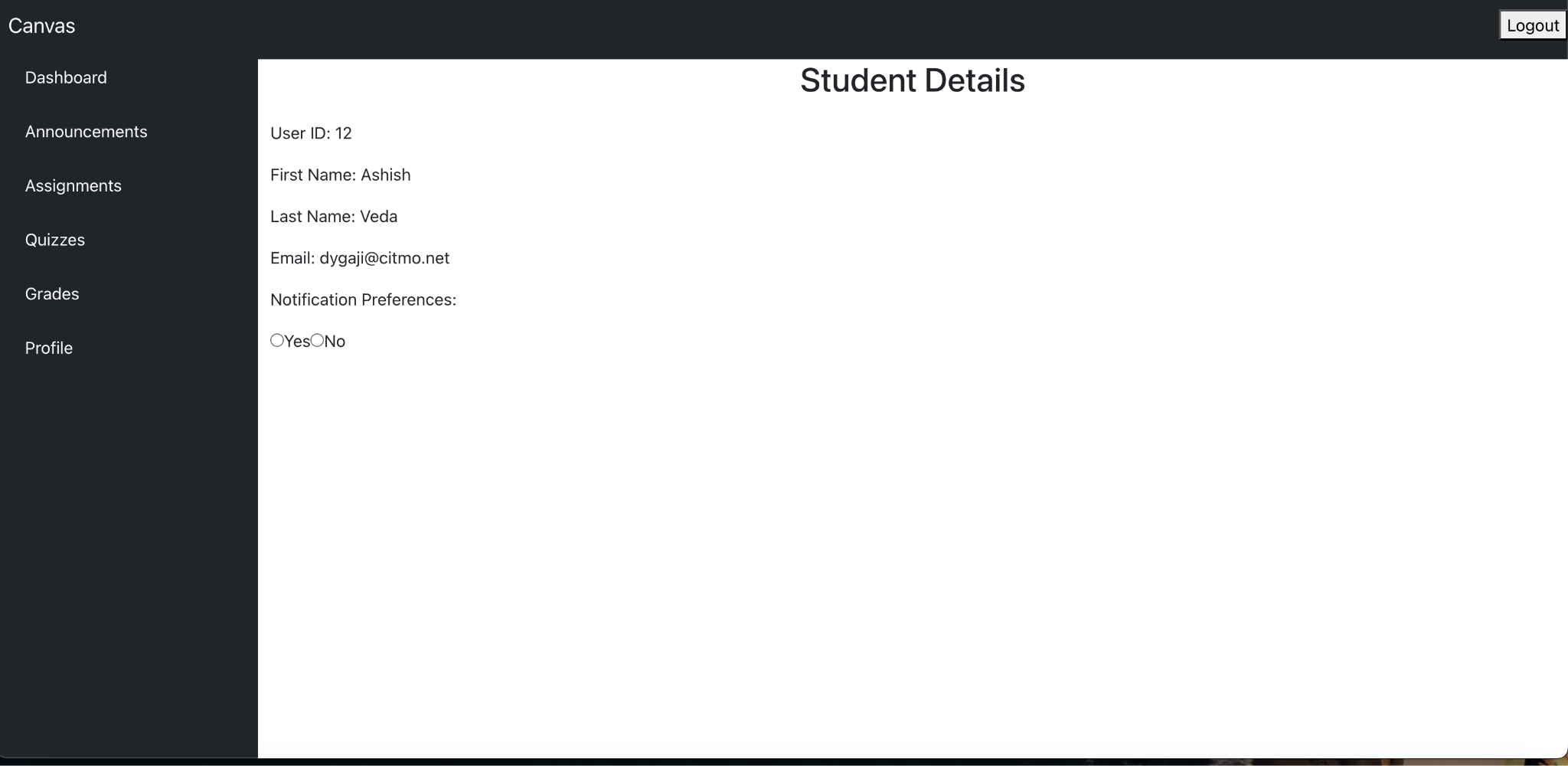
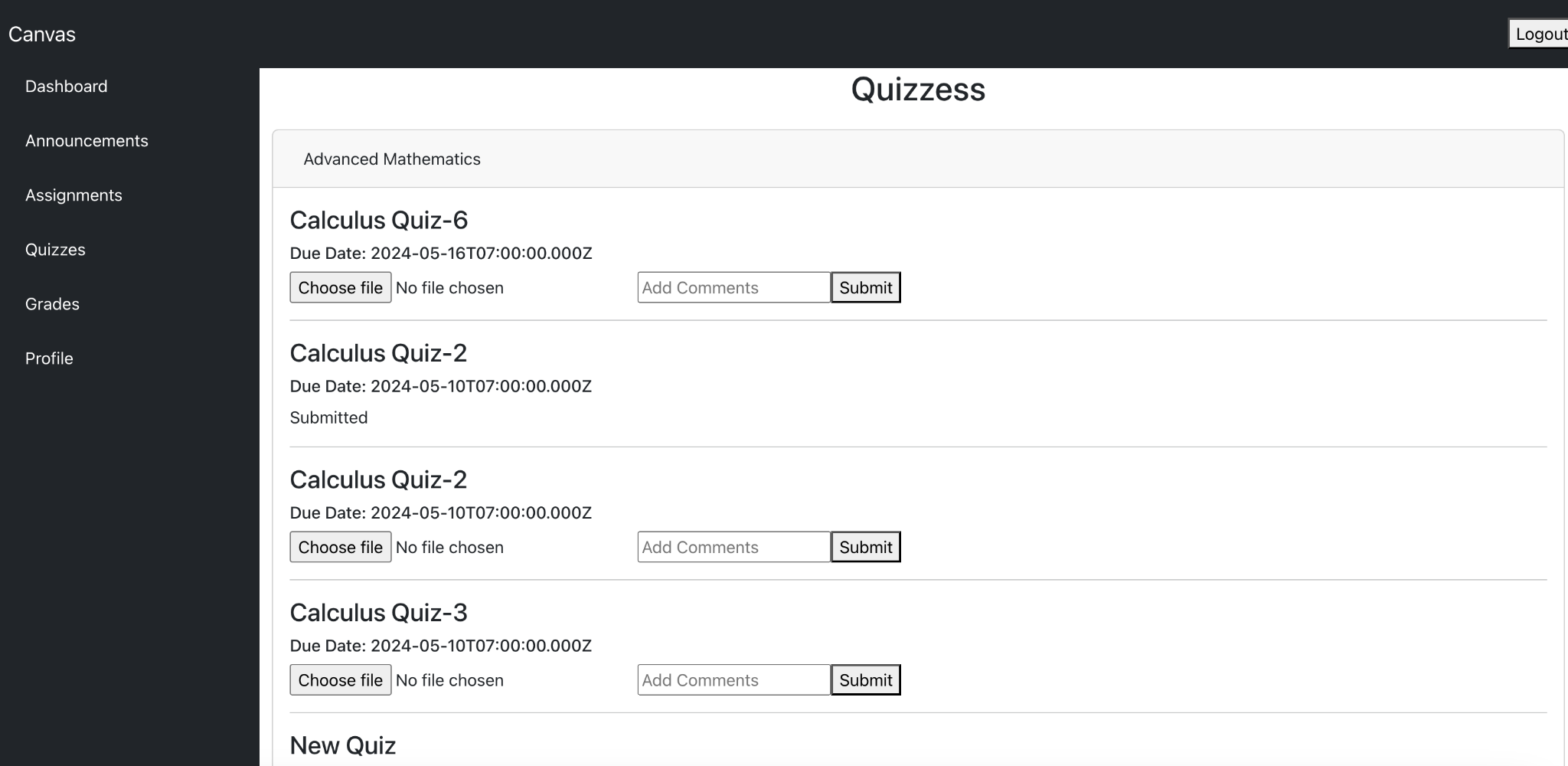
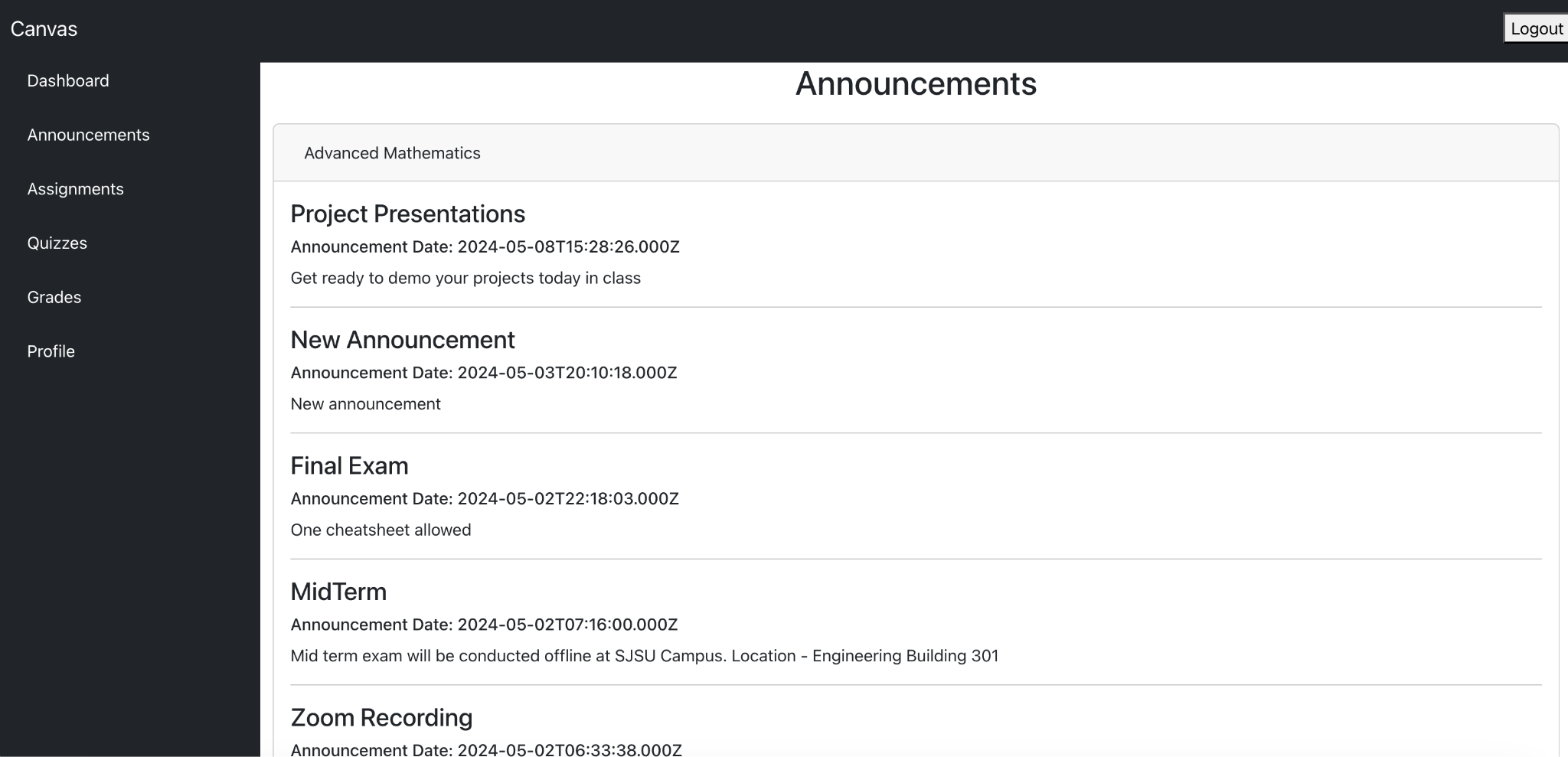
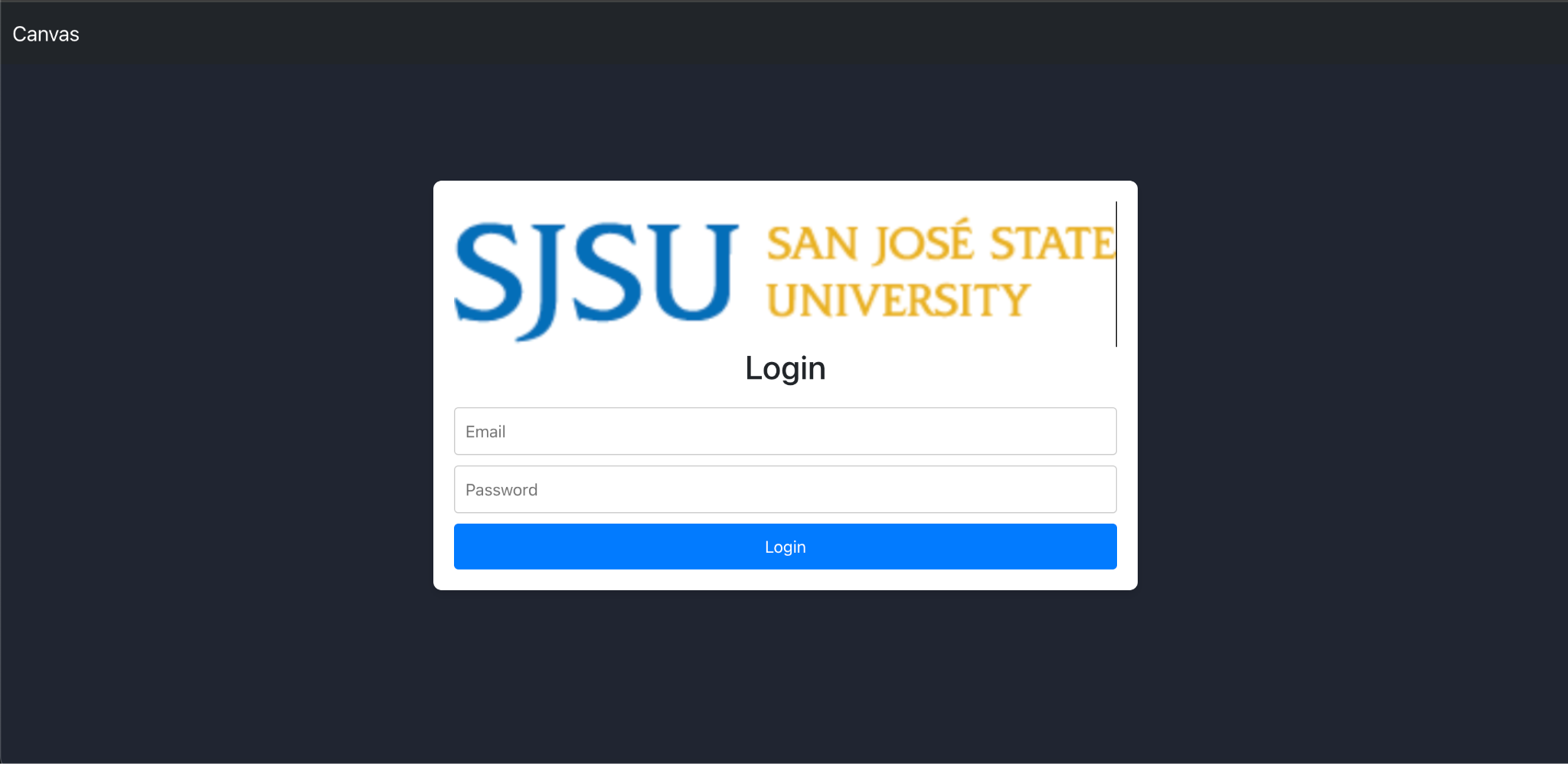
****

****

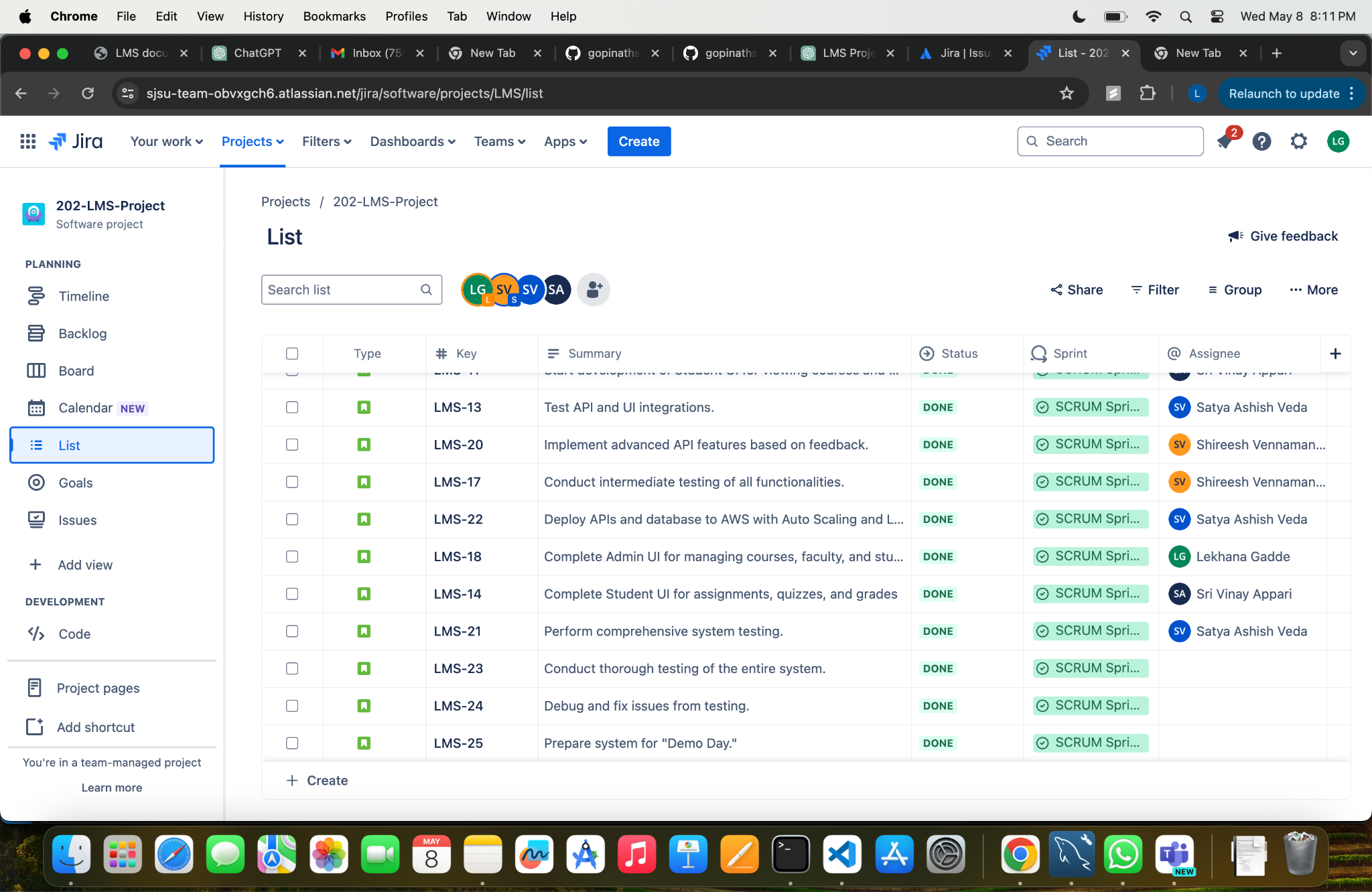
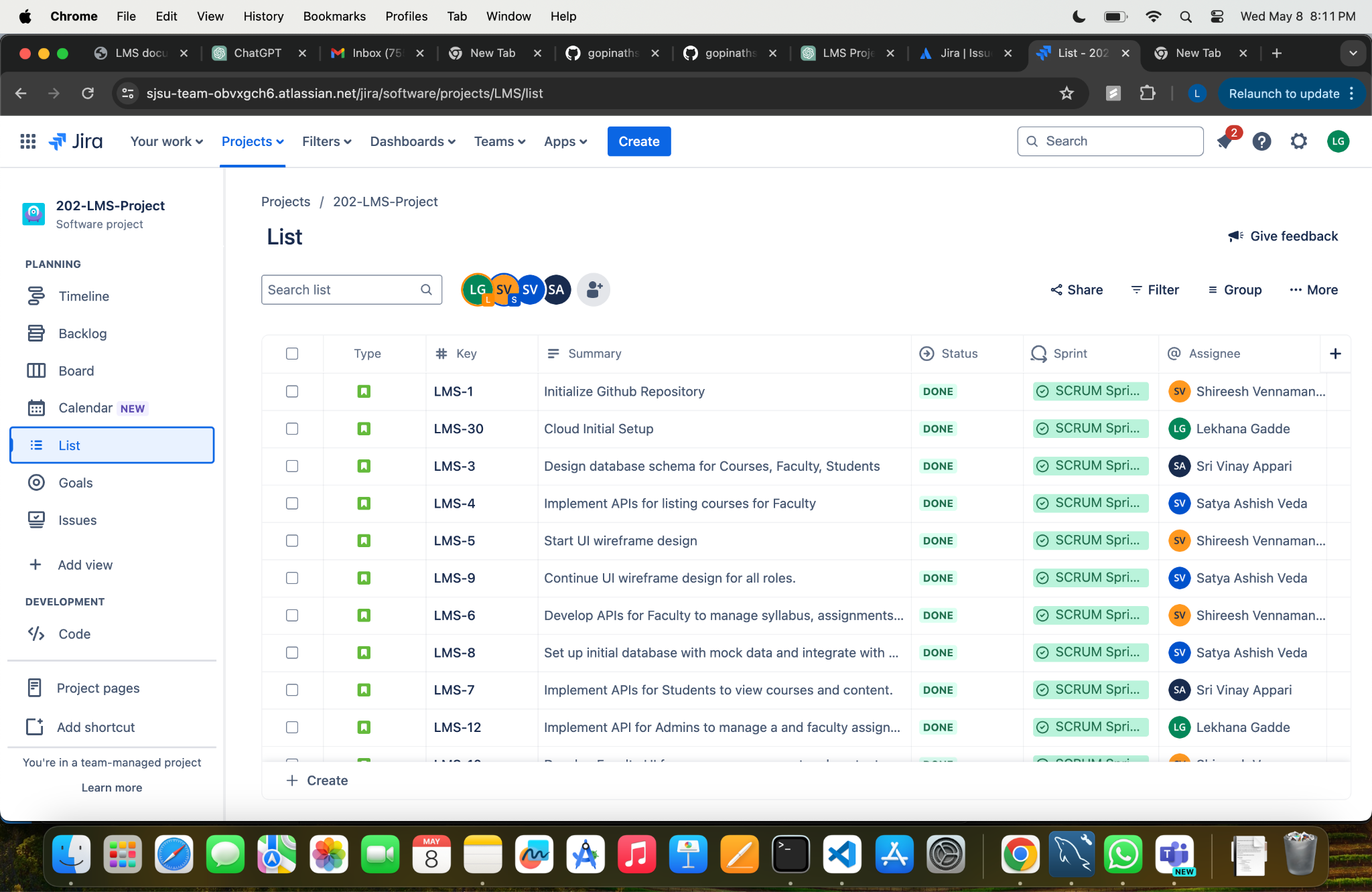
****

****

**User Screens**



**Scrum reports**



**7.** **References**

**Github url :**

**Cloud Credentials :**

Admin credentials : Fitsync-admin

Admin password: Fitsync@1

<https://304106284650.signin.aws.amazon.com/console>

**Tech Stack -**

**Front end :** React

**Backend :** Nodejs

**Database:** Mysql RDS

**Deployment** : AWS EC2 instances