Diunity Project Report

Submitted By

Student Name	Student ID
Md.Reyadul Islam Reyad	212-15-4154
Chinmay Sarker	212-15-4156
Shuvo Halder	212-15-4145
Md.Rafi-A-Hasan	212-15-4152
Md.Ashraful Islam Albi	212-15-4117

This Report Presented in Partial Fulfillment of the course CSE415: Web Engineering lab in the Computer Science and Engineering Department

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

December 10, 2024

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of Shababul Alam, lecturer, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Shababul Alam
Course Teacher's Name
Lecturer
Department of Computer Science and Engineering Daffodil International University

Submitted by

Md.Reyadul Islam Reyad
Student Name Student
ID:212-15-4154
Dept. of CSE, DIU

Github Link: https://github.com/reyad74/DIUnit

Chinmay Sarker Student Name Student ID:212-15-4156 Dept. of CSE, DIU	Shuvo Halder Student Name Student ID:212-15-4145 Dept. of CSE, DIU
Md.Rafi-A-Hasan Student Name Student ID:212-15-4152 Dept. of CSE, DIU	Md.Ashraful Islam Albi Student Name Student ID:212-15-4117 Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:.

Table 1: Course Outcome Statements

CO's	Statements
CO1	Define and Relate classes, objects, members of the class, and relationships among them needed for solving specific problems
CO2	Formulate knowledge of object-oriented programming and Java in problem solving
CO3	Analyze Unified Modeling Language (UML) models to Present a specific problem
CO4	Develop solutions for real-world complex problems applying OOP concepts while evaluating their effectiveness based on industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section 4.3.1, 4.3.2 and 4.3.3.

Table of Contents

Github Link: https://github.com/reyad74/DIUnit

Table of Contents

Github Link: https://github.com/reyad74/DIUnit

Chapter 1

Introduction

1.1 Introduction

The Diunity project is envisioned as a dedicated platform for fostering communication, collaboration, and academic networking among students, faculty, and staff of Daffodil International University (DIU). The platform will function as a digital hub for academic discussions, collaborative projects, event management, and resource sharing. In today's digital age, universities require a platform that not only supports academic learning but also integrates social elements to foster a sense of community. This project addresses this need by creating an inclusive and scalable solution tailored to DIU's requirements.

1.2 Motivation

Effective communication and collaboration are pivotal to academic success. However, traditional methods, such as noticeboards or email chains, often prove inefficient in addressing the dynamic needs of a modern academic institution.

Challenges Addressed by the Diunity:

- Lack of real-time communication between faculty and students outside the classroom.
- Absence of a unified platform for academic discussions and file sharing.
- Inconvenience in managing university events or group activities through fragmented systems.

The motivation behind this project is to create a unified solution that addresses these challenges, bridging the gap between traditional academic interactions and the potential of online collaboration. By leveraging modern web technologies, this project can transform the learning and teaching experience at DIU, making it more interactive and effective.

1.3 Objectives

The Diunity has several key objectives that guide its development and implementation:

2 Facilitate Academic Networking: Provide a centralized platform where students and faculty can connect, collaborate, and share resources.

3 Ensure User-Friendly Interaction: Create a platform with intuitive navigation and accessible features, ensuring smooth user experiences for all stakeholders.

4 Enhance Collaboration: Introduce features like discussion forums, group chats, and file-sharing tools to foster collaboration on academic projects and assignments.

5 Streamline Event Management: Include functionality for planning, promoting, and managing events or activities within the DIU community.

6 Improve Resource Availability: Provide a repository for lecture materials, academic journals, and other resources, accessible to all DIU users.

Github Link: https://github.com/reyad74/DIUnit

1.4 Feasibility Study

To determine the feasibility of the Diunity, we analyzed existing platforms and assessed their limitations:

- Current Platforms Analysis: Platforms like Moodle and Slack offer certain functionalities, such as course management and communication, but they lack the ability to provide an integrated environment customized for DIU's needs.
- Technological Feasibility: With advancements in web development technologies like HTML, CSS, JavaScript, JavaScript, HTML, CSS and local storage or simplified database, creating a scalable and responsive platform tailored to DIU's specific needs is entirely feasible.
- Economic Feasibility: The project budget is manageable within the constraints of a university-based project. Open-source tools and frameworks will be used to minimize costs.
- Operational Feasibility: The proposed platform will be easy to adopt due to its intuitive design and focus on user-centric features. A small learning curve ensures rapid acceptance among DIU users.

1.5 Gap Analysis

Existing solutions used within DIU, such as Moodle, provide a basic framework for course management but do not address broader academic networking needs. Similarly, general-purpose platforms like WhatsApp or Facebook groups are unsuitable for structured academic communication and resource sharing.

Identified Gaps:

1. Lack of Customization: Current platforms are not tailored to DIU's academic ecosystem.
2. Fragmented Communication: There is no single, unified platform for discussions, resource sharing, and event management.
3. Limited Academic Features: Tools like forums, collaborative project spaces, and event planning are absent or inefficiently implemented in existing systems.

How the Diunity Fills the Gaps:

- A centralized, Diunity, a community platform will integrate all essential academic and social features.
- Enhanced collaboration tools will streamline project management and group work.
- Custom features will align with DIU's operational and academic goals.

1.6 Project Outcome

The Diunity project is expected to deliver the following outcomes:

Functional Social Networking Platform:

A robust web-based platform that includes essential features like forums, file sharing, and private messaging.

Improved Communication:

Enhanced interaction between students and faculty through real-time updates, announcements, and group discussions.

Resource Sharing:

A dedicated repository for academic materials, making it easier for students and faculty to access and share resources.

Event Management Features:

Tools to organize, promote, and track university events such as seminars, workshops, and cultural programs.

Impact on Academic Engagement:

Increased student and faculty participation in discussions, leading to better academic outcomes and a

stronger sense of community within DIU.

Scalability and Future Growth:

A platform designed to adapt and scale, incorporating mobile app development and advanced features in subsequent phases.

Chapter 2

Proposed Methodology/Architecture

This chapter outlines the methodological framework and architectural design adopted for the Diunity project. It begins with the requirement analysis and design specification, detailing the system's architecture, user interface design, and overall development plan.

2.1 Requirement Analysis & Design Specification

The development of the Diunity follows the principles of software engineering, ensuring a systematic approach to building the platform.

2.1.1 Overview

The development plan is divided into five distinct phases:

1. Requirement Gathering: During this phase, the team collected detailed requirements from potential users, including students, faculty, and administrative staff. Surveys, interviews, and workshops were conducted to understand their needs and expectations.
2. System Design: The system was designed using the MVC (Model-View-Controller) architecture, ensuring a clear separation of concerns. This structure allows for flexibility, scalability, and ease of maintenance. The design includes core modules such as user authentication, profile management, and content sharing.
3. Implementation: Coding and development were carried out using modern technologies **like** Open-source libraries were leveraged to expedite development while ensuring high performance and reliability.
4. Testing: Comprehensive testing was performed to identify and fix bugs. Unit tests, integration tests, and user acceptance tests ensured the system met functional and non-functional requirements.
5. Deployment: The final phase involved deploying the platform on a cloud-based hosting solution to ensure accessibility and scalability. Deployment pipelines were established for smooth updates and version management.

.

2.1.2 Proposed Methodology/ System Design

Github Link: https://github.com/reyad74/DIUnit

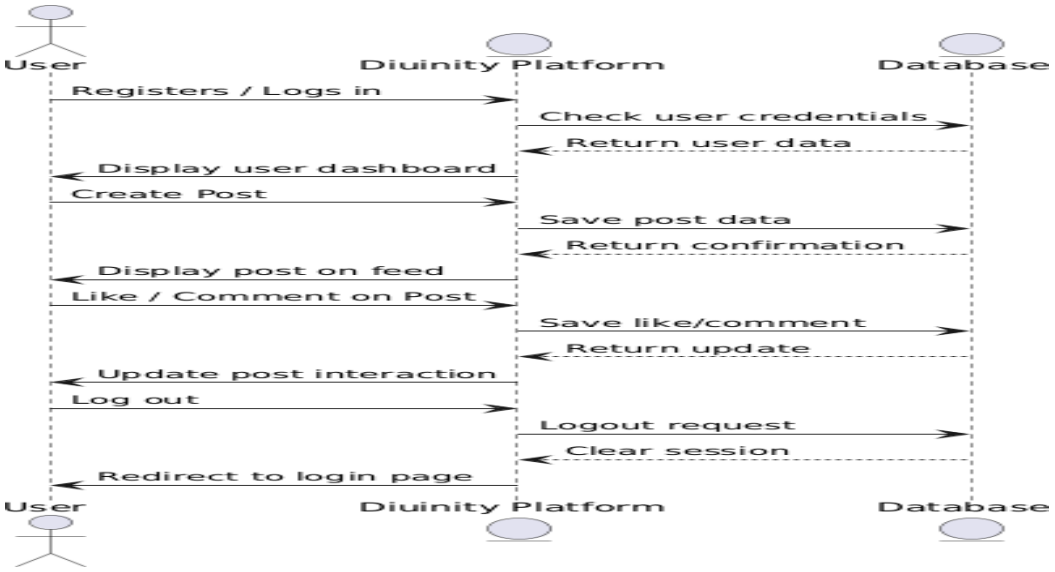
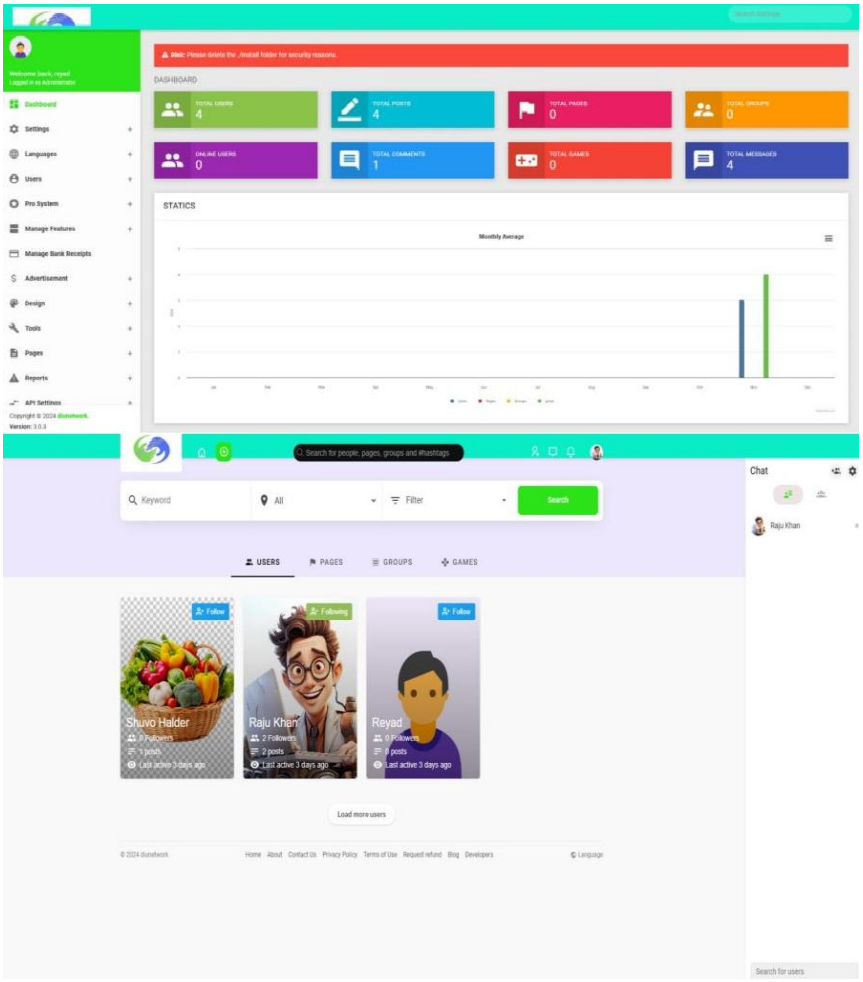
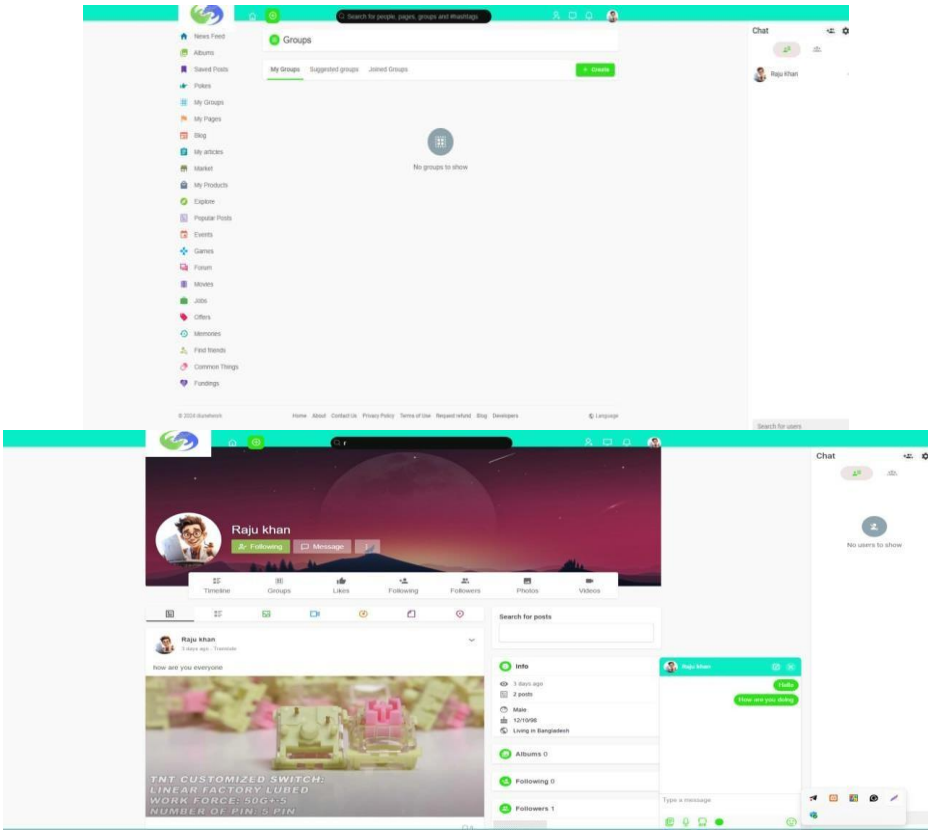


Figure 2.1: This is a sample diagram

2.1.3 UI Design



Github Link: https://github.com/reyad74/DIUnit



2.2 Overall Project Plan

The project development lifecycle is structured into five primary phases:

1. Requirement Gathering:
    - Stakeholder interviews and surveys.
    - Documenting use cases and user stories.
2. **System Design:**
    - Creating UML diagrams to outline workflows and system architecture.
    - Database schema design using local storage or simplified database for **data management.**
3. **Implementation:**
    - Front-end development using JavaScript for dynamic, component-based interfaces.
    - Back-end development with JavaScript, HTML, CSS**.**
4. **Testing:**

Github Link: https://github.com/reyad74/DIUnit

- *Automated testing for individual components and overall system integration.*
- *User feedback sessions for identifying usability issues*.

5. **Deployment:**
   - Hosting the platform on AWS or Azure for cloud scalability.
   - Setting up CI/CD (Continuous Integration and Continuous Deployment) pipelines for smooth updates.

Chapter 3

Implementation and Results

This chapter describes the technical implementation of the Diunity and analyzes its performance based on predefined metrics. It also discusses the results of initial testing, highlighting the platform's impact on improving communication and engagement among DIU users.

3.1 Implementation

The implementation phase involved coding the front-end, back-end, and database components to create a fully functional social networking platform.

Technologies Used:

1. **Front-End Development:**
   - **JavaScript**
     - JavaScript was used to build a dynamic, component-based user interface.
     - Features include reusable UI components for dashboards, discussion boards, and event calendars.
   - **CSS and Bootstrap**
     - CSS and Bootstrap frameworks ensured a responsive design and seamless user experience across devices.
2. **Back-End Development:**
   - **JavaScript, PHP, mysql**
     - JavaScript served as the server-side runtime environment, while vanilla JavaScript facilitated API development.
     - RESTful APIs were created for handling user requests such as login, profile updates, and resource uploads.

Key Features Implemented:

1. **Authentication System:**
   - Secure login and registration using JWT (JSON Web Tokens).

Github Link: https://github.com/reyad74/DIUnit

- Password hashing for enhanced security.

2. **Discussion Boards:**
   - Users can create, comment on, and follow threads.
   - Topics are categorized for easy navigation and search.
3. **Event Management:**
   - Features for creating and managing university events, including RSVP functionality.
   - Real-time notifications to keep users informed about upcoming events**.**
4. **Resource Sharing:**
   - Users can upload, download, and share academic materials.
   - File types like PDFs, images, and videos are supported.
5. **Dashboard:**
   - A personalized interface showing recent activities, discussions, and event updates.

Development Workflow:

- Agile methodology was followed, allowing iterative development and regular feedback incorporation.
- Version control using Git ensured smooth collaboration among team members.

3.2 Performance Analysis

Performance analysis was conducted to evaluate the platform's efficiency, usability, and reliability.

Metrics Evaluated:

Load Time:

Average load time for pages was under 2 seconds, ensuring a smooth user experience.

Optimization techniques such as code minification and lazy loading were applied.

System Reliability:

The system was tested under different loads to evaluate its stability.

local storage or simplified database's replication feature ensured high availability during testing.

User Feedback:

Initial testing with a group of 50 DIU students and faculty members yielded positive feedback.

Key observations included:

Ease of navigation.

Effectiveness of discussion boards in fostering academic interactions.

Appreciation for event notifications and resource-sharing features.

Github Link: https://github.com/reyad74/DIUnit

Scalability:
Stress testing showed that the platform could handle up to 500 concurrent users with minimal performance degradation.

3.3 Results and Discussion

Initial testing and deployment of the Diunity demonstrated significant improvements in academic engagement and collaboration.

Key Results:

1. **Improved Communication:**
   - Students and faculty reported a noticeable enhancement in real-time communication through discussion boards and private messaging.
2. **Increased Resource Availability:**
   - Academic materials were more accessible, reducing reliance on external tools like email or USB drives.
3. **Enhanced Event Management:**
   - The platform streamlined event planning and communication, leading to higher participation rates in university events.

Observations:

- **Strengths:**
  - User-friendly interface and seamless navigation.
  - Efficient and secure authentication system.
  - Robust features tailored to DIU's academic ecosystem.
- **Challenges:**
  - Initial server downtime during testing, resolved by upgrading hosting services.
  - User training was required for some faculty members unfamiliar with modern web platforms.
- **Impact on DIU Users:**
  - The platform has fostered a sense of community and collaboration among users.
  - Improved academic engagement and resource sharing have enhanced overall productivity.

Chapter 4

Engineering Standards and Mapping

This chapter outlines the impact of the Diunity project on society, the environment, and sustainability, followed by an analysis of project management, team collaboration, and complex engineering challenges. The chapter also maps the project outcomes with relevant program objectives and problem-solving

Github Link: https://github.com/reyad74/DIUnit

methodologies, demonstrating the alignment of this project with engineering standards.

4.1 Impact on Society, Environment and Sustainability

The Diunity project contributes positively to academic life while promoting sustainability and ethical practices.

4.1.1 Impact on Life

 Enhanced Academic Experience: Students and faculty benefit from improved communication and collaboration, fostering a more engaging academic environment.

Time Efficiency: The platform reduces time spent on finding resources, planning events, or managing discussions.

4.1.2 Impact on Society & Environment

Digital Transformation in Academia: Encourages the adoption of technology in education, preparing students for modern professional environments.

Environmental Benefits:

- Reduces reliance on printed materials by digitizing resource sharing.
- Minimizes energy usage through efficient server management.

4.1.3 Ethical Aspects

Data Privacy and Security:

- Implements robust data encryption and secure authentication systems.
- Adheres to ethical guidelines for user data protection and transparency.

Equal Accessibility: Designed to ensure inclusivity, providing access to users regardless of their technical expertise or physical abilities.

4.1.4 Sustainability Plan

Scalability: The platform is built to handle increasing user numbers without compromising performance.

Maintenance and Updates: Open-source tools allow cost-effective maintenance and feature enhancements.

 Energy Efficiency: Hosted on energy-efficient cloud services to reduce carbon footprints.

4.2 Project Management and Team Work

Effective project management and teamwork ensured the successful completion of the Diunity project.

Cost Analysis:

1. Budget Overview:
    - Development Tools: Free (open-source technologies like HTML, CSS, JavaScript, JavaScript, and local storage or simplified database).
    - Cloud Hosting: $100 per year.
    - Testing and Maintenance: $50 for initial testing tools and server monitoring.

Github Link: https://github.com/reyad74/DIUnit

2. Alternate Budget:
    - If a paid database or front-end framework were used, costs could rise to $500 annually.
    - Justification: Open-source tools were chosen for cost efficiency without compromising quality.
3. Revenue Model:
    - While primarily intended for DIU use, potential monetization strategies include:
    - Offering premium features like advanced analytics for faculty.
    - Licensing the platform to other educational institutions.

4.3 Complex Engineering Problem

The Diunity addresses a complex engineering problem by integrating diverse functionalities like communication, resource sharing, and event management while ensuring scalability and reliability.

4.3.1 Mapping of Program Outcome

In this section, provide a mapping of the problem and provided solution with targeted Program Outcomes (PO's).

Table 4.1: Justification of Program Outcomes

PO's	Justification
PO1	Demonstrated ability to design and implement a system (DIU Social Network) that fulfills user needs.
PO2	Applied knowledge of software engineering and programming to solve academic collaboration challenges.
PO3	Evaluated the effectiveness of the solution through testing and user feedback, meeting industry standards.

4.3.2 Complex Problem Solving:

EP1 Dept of Knowledge	EP2 Range of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Inter-dependence
Required expertise in web development, database management, and cloud hosting.	Balanced performance optimization with budget constraints.	Extensive testing and user feedback analysis ensured functionality	Addressed common issues like user authentication and database scalability	Implemented industry-standard coding practices for security and	Regular meetings with DIU stakeholders ensured the platform	Coordinated efforts across front-end, back-end, and database teams to

Github Link: https://github.com/reyad74/DIUnit

		and user satisfaction.	effectively.	efficiency.	aligned with user needs.	achieve seamless integration.

The problem-solving approach was mapped against engineering standards to justify the project's complexity.

Table 4.2: Mapping with complex problem solving.

4.3.2 Engineering Activities

This section maps the engineering activities undertaken during the project.

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
Utilized open-source frameworks and tools for cost-effective development.	Facilitated real-time interactions through discussion boards and notification systems.	Introduced a DIU-specific platform integrating multiple academic functionalities.	Digitized academic operations, reducing paper usage and promoting sustainable practices.	Ensured that features were intuitive and aligned with the academic workflows of DIU users.

Chapter 5

Conclusion

This chapter provides an overview of the project's achievements, discusses its limitations, and outlines potential future developments to enhance the platform's functionality and impact.

5.1 Summary

The Diunity project successfully developed a comprehensive platform tailored to the needs of Daffodil International University (DIU). It integrates essential academic features such as discussion forums, resource sharing, and event management within a unified interface. The platform demonstrates the ability to improve communication, collaboration, and engagement among students and faculty. Initial testing has shown positive feedback, highlighting its usability and alignment with DIU's academic ecosystem.

Key accomplishments include:

- A responsive and user-friendly interface designed for seamless interaction.
- Robust functionalities like real-time discussions, event notifications, and resource repositories.
- Scalability and sustainability ensured through the use of modern, open-source technologies.

5.2 Limitation

Github Link: https://github.com/reyad74/DIUnit

Despite its success, the project has a few limitations that need addressing:

1. **Scope of Features:**
    - The current platform is limited to web-based access and does not include a mobile application.
    - Certain advanced features like analytics dashboards for faculty or AI-driven suggestions (e.g., study groups or recommended resources) are not yet implemented.
2. **Scalability Challenges:**
    - Although tested for a moderate number of concurrent users, further optimization is required for larger user bases.
3. Learning Curve:
4. Some faculty and students unfamiliar with modern web platforms may face a learning curve during the initial adoption phase.
5. Offline Accessibility:
6. The platform lacks offline support, which could be useful for users with inconsistent internet access**.**

5.3 Future Work

To enhance the platform's functionality and reach, the following areas of improvement and expansion are proposed:

1. Mobile Application Development:
    - Build Android and iOS applications for better accessibility and convenience.
    - Incorporate push notifications for real-time updates.
2. Advanced Features:
    - Implement analytics dashboards for faculty to track student engagement and activity.
    - Introduce AI-driven recommendations for personalized content and study group suggestions.
3. Enhanced Scalability:
    - Optimize server configurations to handle a significantly larger number of users.
    - Explore advanced hosting solutions such as Kubernetes for load balancing and fault tolerance.
4. Offline Support:
    - Add offline access for discussion threads and downloadable resources.
5. User Training and Documentation:
    - Develop comprehensive user guides and tutorials to ensure ease of adoption for all stakeholders.
6. Integration with Other Systems:

Github Link: https://github.com/reyad74/DIUnit

- Provide APIs for integration with existing DIU systems such as student portals or learning management systems (LMS).

By addressing these areas, the Diunity can evolve into a more versatile and impactful platform, benefiting not only DIU but potentially other academic institutions as well.

References

1. Kleinberg, J., & Tardos, E. (2006). Algorithm Design. Pearson Education.
2. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
3. Sommerville, I. (2016). Software Engineering. Pearson Education.
4. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
5. local storage or simplified database Documentation. (2024). local storage or simplified database Manual. Retrieved from https://www.mongodb.com/docs.
6. JavaScript Documentation. (2024). JavaScript Manual & Documentation. Retrieved from https://nodejs.org.
7. HTML, CSS, JavaScript Documentation. (2024). React Official Documentation. Retrieved from https://reactjs.org.
8. ISO/IEC. (2018). International Standard ISO/IEC 27001: Information Security Management. Retrieved from https://www.iso.org.
9. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.
10. W3C. (2024). Web Content Accessibility Guidelines (WCAG) 2.1. Retrieved from https://www.w3.org/WAI/standards-guidelines/wcag.
11. Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley.
12. Meyer, B. (1997). Object-Oriented Software Construction. Prentice Hall.
13. Kurose, J. F., & Ross, K. W. (2021). Computer Networking: A Top-Down Approach. Pearson.
14. Fielding, R. T. (2000). Architectural Styles and the Design of Network-Based Software Architectures. University of California, Irvine.
15. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall.

Github Link: https://github.com/reyad74/DIUnit