

**CS673 Software Engineering**  
**Team 4 - Fantastic 5**  
**Project Proposal and Planning**

<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Brenna Mahn	Team Leader	<u>Brenna Mahn</u>	<u>May 13, 2024</u>
Savien (Brooke)Love	Design & Implementation Leader	<u>Savien Brooke Love</u>	<u>May 13, 2024</u>
Pinwen Mu	Configuration Leader	<u>Pinwen Mu</u>	May 14, 2024
Natanim Eibrahim	Security Leader	<u>Natanim Eibrahim</u>	May 14, 2024
Fehmi Baltaci	Requirement Leader, QA Leader	<u>Fehmi Baltaci</u>	<u>May 13, 2024</u>

**Revision history**

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
<u>1.0</u>	Brenna Mahn, Savien (Brooke) Love, Pinwen Mu, Natanim Eibrahim, Fehmi Baltaci	<u>05/14/20204</u>	<u>Iteration 0 Draft</u>
<u>1.1</u>	Brenna Mahn, Savien (Brooke) Love, Pinwen Mu, Natanim Eibrahim, Fehmi Baltaci	<u>05/28/2024</u>	<u>Iteration 1</u>
<u>2.0</u>	Brenna Mahn, Savien (Brooke) Love, Pinwen Mu, Natanim Eibrahim, Fehmi Baltaci	<u>6/11/2024</u>	<u>Iteration 2</u>
<u>3.0</u>	Brenna Mahn, Savien (Brooke) Love, Pinwen Mu, Natanim Eibrahim, Fehmi Baltaci	<u>6/17/2024</u>	<u>Iteration 3</u>

[Overview](#)

[Related Work](#)

[Proposed High level Requirements](#)

[Management Plan](#)

[Objectives and Priorities](#)

[Risk Management \(need to be updated constantly\)](#)

[Timeline \(need to be updated at the end of each iteration\)](#)

[Configuration Management Plan](#)

[Tools](#)

[Deployment Plan if applicable](#)

[Quality Assurance Plan](#)

[Metrics](#)

[Code Review Process](#)

[Testing](#)

[Defect Management](#)

[References](#)

[Glossary](#)

## 1. Overview

In today's rapidly evolving educational landscape, the need for a dynamic and interactive learning platform is necessary. The proposed online project-based learning platform aims to offer this service by providing a comprehensive, user-friendly environment that supports learning through real-world projects

This project aims to develop a user-friendly and engaging online learning platform to support project-based learning. We will focus on the frontend (UI) development, creating a seamless and intuitive interface for the platform. We aim to provide a better engagement experience for our target users, ensuring seamless interaction with the backend systems. By creating an intuitive and visually appealing platform, we enhance user satisfaction and facilitate smoother navigation and interaction with the platform's comprehensive features.

## Motivation

The main initiative for this project is driven by the need to bridge the gap between theoretical knowledge and real world application, enhancing the overall educational experience through technology. We aim to enhance learning outcomes by promoting practical applications of theoretical knowledge, increase engagement and collaboration among learners, and provide continuous assessment and feedback.

## Purpose

The primary purpose of this online platform is to function as a central hub for educators, students, and organizational members, facilitating the management and collaboration on project-based learning activities. Ultimately enhance the learning experience by providing a comprehensive platform that integrates user and project management, content organization, collaboration tools, and assessment features, along with timely notifications, thereby optimizing communication and resource sharing.

## Target Customers and Potential Users

The platform is primarily targeted at educational institutions, including universities, colleges, and K-12 schools. It is also suitable for training organizations and corporate entities that utilize project-based learning methodologies for employee training.

## Potential Users

- **Students:** Students will benefit from a dedicated space designed to gain practical experience and skills in their fields of study. The platform provides a user-friendly interface that allows seamless access to project details, learning materials, and collaboration tools. Students can effortlessly collaborate with peers, communicate with other users, and track their progress.
- **Educators:** Implementing project-based learning modules effectively. Our focus on clear and intuitive design will enable educators to easily create, edit, and track project details, simplifying the management and evaluation of educational activities.
- **Institution members:** Administrative staff involved in overseeing and facilitating educational activities.

## Basic Functionality with a Frontend Focus

As the team is concentrated on frontend development, our efforts are directed towards creating a user-friendly and visually appealing interface that supports the following functionalities:

1. **User Management Interface:** Easy-to-navigate screens for registration, login, and profile management. Clear visual cues for roles, permissions, and team assignments.
2. **Project Management Interface:** Intuitive project creation and management tools, with features like seamless search functions and an interactive rating system.

3. **Content/Resource Management Interface:** Streamlined management of educational content and blogs. Designing user interfaces for students to access learning materials and potentially allowing educators to upload additional resources.
4. **Collaboration:** Developing interfaces for real-time chat, discussion forums, and document sharing functionalities, ensuring smooth communication and collaboration among students.
5. **Assessment and Analytics Dashboard:** Visually rich and interactive dashboards for tracking progress, receiving feedback, and viewing reports.
6. **Notification System:** We'll design notification pop-ups or alerts to inform users about deadlines, feedback, announcements, and other important updates.

### Technology Stack

- **Frontend Framework:** React.js
- **CSS PreProcessor:** SASS
- **React Component Libraries:** Material UI, Formik
- **JavaScript Libraries:**
- **Version Control System:** Git
- **Project Management Tool:** Pivotal Tracker, Docker
- **Test Framework:** Python-Behave-Selenium

Throughout the project, we'll collaborate with the backend development team to ensure seamless integration between the UI and the platform's functionalities. Our focus on the frontend will be crucial in creating a user-friendly and engaging platform that empowers educators and students to leverage the benefits of project-based learning.

## 2. Related Work

During our comprehensive online research, one of the platforms identified that closely aligns with the concept of our proposed project is [TalentLMS](#). This platform is a well-rounded learning management system (LMS) crafted to support the creation and management of educational courses and content. TalentLMS stands out due to its array of features designed to simplify the training process, making it versatile and easily adaptable to various educational settings.

Based on my analysis of the TalentLMS website, here are some key features it offers:

- **Course Creation and Delivery:** TalentLMS enables the creation and delivery of online courses, supporting various media formats such as videos and documents.
- **Progress Tracking:** The platform provides robust tools to monitor learner progress, including tracking course completion rates and the amount of time spent on each module.

- Video Conferencing and Live Interaction: The platform supports video conferencing, allowing for live interactions, and integrates seamlessly with various third-party applications to enhance its utility.

Our project distinguishes itself from TalentLMS in several key areas:

- Focus on Project-Based Learning: Unlike TalentLMS, which serves as a general-purpose LMS suitable for various training scenarios, our platform is specifically designed to support project-based learning. This approach involves students engaging in real-world projects to acquire new skills and knowledge, providing a practical application of theoretical concepts.
- Enhanced Collaboration Features: To facilitate effective teamwork, our platform will include comprehensive collaboration tools such as real-time chat, discussion forums, and document sharing. These features are tailored to support group dynamics and interactive learning, which are essential for project-based learning environments.
- Advanced Assessment and Analytics: Our platform emphasizes tools for educators to assess learning outcomes and monitor progress on projects. It is equipped to handle detailed analytics on student engagement and project completion, which differs from the traditional e-learning assessments like quizzes and exams predominantly offered by TalentLMS.

Overall, our project offers a unique set of features that are specifically tailored to the needs of project-based learning. While TalentLMS is a powerful LMS, it is not designed to meet the specific needs of educators and students who are engaged in project-based learning activities.

### 3. Proposed High level Requirements

#### a. Functional Requirements

##### i. Essential Features

- User registration:  
As a user I want to be able to register a user account so that I can join the platform
- User login:  
As a user I want to be able to log in to my account so that I can access the platform
- Project creation:  
As a project lead I want to be able to create a project so that I can initiate new tasks or initiatives
- Project Editing: As a project lead I want to be able to edit a project so that I can keep information up to date
- Task Assignment and Deadline Setting:

As a project lead I want to be able to assign tasks and set deadlines so that the project progresses smoothly

- Team roles and permission management:

As a project lead I want to be able to set team roles and permissions so that I can manage access

- Project Search:

As a user I want to be able to search for projects so that I can find relevant project information easily

## ii. Desirable Features (the nice features that you really want to have too):

- User profile:

As a user I want to create and edit my profile so that other users can identify me and understand my role

- Project Analytics:

As a user I want to view project analytics so that I can evaluate project progress

- Deadline Notifications:

As a user I want to receive notifications about upcoming deadlines related to projects or tasks so that I can stay informed and meet deadlines effectively

- Update Notifications:

As a user I want to receive notifications about project updates so that I can stay informed about any changes

## iii. Optional Features (additional cool features that you want to have if there is time):

- Learning Materials:

As a user I need to be able to manage learning materials for educational purposes so that I can access relevant resources for my learning

- Blog Publishing:

As a user I need to be able to manage and publish blog content so that I can share my insights and knowledge with others

- Content Moderation:

As a project lead I need to be able to moderate and manage learning content so that the platform remains organized and relevant

- Live chat:

As a user I want to be able to live chat with other users so that we can communicate quickly to collaborate

- Discussion Forum:

As a user I want to be able to participate in discussion forums so that I can contribute to discussions and learn from others

- Document Share:

As a user I want to be able to share documents with team members so that we can collaborate on projects effectively

b. Nonfunctional Requirements

i. Performance requirements

- a. Speed and Responsiveness: Ensure the interface responds swiftly to user interactions, with minimal delay in loading content or executing actions.

ii. Usability requirements

- a. User-friendliness: The user interface of the platform should be intuitive, with a clear and logical layout that is easy for new users to understand.
- b. Accessibility: The user interface must comply with accessibility standards to ensure the platform is usable by people with disabilities, including keyboard navigability and screen reader support.

iii. Scalability requirements

- a. Adaptability: Design the frontend to handle varying loads and an increasing number of users, ensuring it can scale without degradation in performance.

iv. Security requirements

- a. Data Protection: Implement secure coding practice to protect user data to prevent common vulnerabilities like SQL injection, cross-site scripting (XSS)
- b. Authentication and Authorization: The platform should have secure user authentication and ensure user data and actions are protected based on their roles.

v. Compatibility requirements

- a. Cross-Platform Compatibility: Ensure the frontend works seamlessly across different browsers and devices, including mobiles, tablets, and desktops.

## 4. Management Plan

### a. Objectives and Priorities

1. Complete All Proposed Essential Features: To fully implement all core functionalities outlined in the project scope, ensuring the platform meets the fundamental requirements for user management, project management, collaboration, and assessment services.
2. Maintain High Quality Throughout the Development Process: To uphold high standards in code quality, user interface design, and overall user experience throughout the development lifecycle.
3. Ensure the Software is Free of Known Bugs: To rigorously test the platform to identify and resolve all critical bugs before the product goes live, ensuring stability and reliability.
4. Adhere to Security Best Practices: To integrate comprehensive security measures throughout the platform, protecting user data and ensuring compliance with relevant regulations.
5. Deploy the Software Successfully: To ensure that the software deployment is seamless, with the platform being fully operational and accessible to all users upon launch.
6. Ensure Scalability and Performance: To build the platform with scalable architecture and efficient performance benchmarks to handle increasing loads and future expansions.
7. Foster User Engagement and Satisfaction: To design the interface and interactions in a manner that is intuitive and user-friendly, promoting regular and sustained engagement from the users.

### b. Risk Management

Our project team has identified several key risks that could impact the development and deployment of our software. To manage these risks effectively, we have adopted a proactive risk management approach. Key risks include unclear requirements, not enough testing, not enough time for integration and deployment, integration complexities, deployment failures, security vulnerabilities, and unforeseen bugs. Our strategy involves regular risk assessment meetings to update and refine our risk management strategies. We also plan to use automated testing and continuous integration tools to detect and resolve issues early. Security audits and adherence to best security practices will be routine to safeguard against vulnerabilities. Additionally, we will maintain a detailed risk management log in a Google Sheet, which will be updated constantly to reflect the current status and actions taken for each identified risk.

For detailed risk management updates and ongoing tracking, please refer to our dedicated Google Sheet in the project folder: [Link to the Google Sheet](#). This sheet



contains a comprehensive breakdown of each risk, including risk title and description, estimated likelihood of occurrence and estimated impact, estimated retirement cost, priority and responsible engineers, detailed plan and status updates, ensuring transparency and accountability throughout the project lifecycle.

c. Timeline (this section should be filled in iteration 0 and updated at the end of each later iteration)

Iteration	Functional Requirements(Essential/Disable/Option)	Tasks (Cross requirements tasks)	Estimated/real person hours
1	User Registration User Login Project Creation Project Editing	Primarily UI Development	20 15 25 10
2	Prepare to integrate UI components with other teams' services UI Design Development Testing	UI Development, Backend Integration, Testing	15 20 15
3	Integrate UI components from Iter1 with other teams' services Project Search	UI Development, Backend Integration, Testing	15 20 20

## 5. Configuration Management Plan -Lillian

### a. Tools

For the frontend UI development, our team will use the following tools:

- Version Control: Git and GitHub
- IDE: Visual Studio Code (VS Code)
- Frontend Frameworks/Libraries: React.js
- Package Managers: npm
- CI/CD: GitHub Actions

- Containerization: Docker

## **b. Code Commit Guideline and Git Branching Strategy**

- Branching:
  - Iteration Branches: Each iteration of the project will have its own branch, named 'iterX'.
  - Feature Branches: Each developer will create a branch for their current assigned ticket with the format: feature\_<first initial last initial>\_<feature name>.
  - Main Branch: The main branch will continue to hold the official release history. It will only be updated from iteration branches at the end of each successful iteration, ensuring that the code in main is always stable and production-ready.
  - Bugfix Branches: Branches named bugfix/ui-<bug\_name> for urgent fixes
- Pull Requests: Mandatory for every change, and must be reviewed by the Design & Implementation Leader or Team Leader before merging.
- Commit Messages: Should include the type of change (feature, fix, update), a concise description, and the UI screen or component affected.

## **c. Deployment Plan if applicable**

- Environment Coordination:
  - Backend Integration: Regularly meet with backend teams to ensure that the frontend interfaces (APIs) are fully compatible with backend services.
  - Environment Setup: Ensure that the testing environment mirrors the AWS production settings
- Automated Deployment:
  - Build Process: Use GitHub Actions to automate the frontend's build and testing processes.
  - Continuous Deployment: Configure GitHub Actions to automatically deploy successfully built code to AWS. Before deployment, ensure all configuration files and environment variables are adapted for the AWS production environment.
- Version Control and Rollback:
  - Semantic Versioning: Manage releases using semantic versioning, for example, using the format major.minor.patch.
  - Deployment Version Management: Use tags and version numbers on AWS to manage deployment versions, facilitating tracking and rollback to previous stable versions.

## 6. Quality Assurance Plan

### a. Metrics

Defect Rate and Test Coverage will serve as our primary metrics for evaluating the application's quality and testing effectiveness.

Metric Name	Description
Defect Rate	Quantifies the number of defects or bugs identified against the total number of features implemented. It will provide more information about the applications stability and the effectiveness of the development/testing process.
Test Coverage	Measures the size of the application's features that are tested by our test cases. It is calculated like this: $\text{test\_case\_amount} / \text{total\_number\_of\_features}$ This metric will help ensure that we have enough testing coverage for the application.

### b. Coding Standard

We will follow the industry-standard coding guidelines;

- Consistent Naming Conventions: Clear and consistent naming for variables, functions, and classes that are being used in the framework.
- Proper Commenting: Necessary comments will be added to all function and class
- Often Refactoring Code: Code that is used will be refactored as we build the framework.
- Adherence to Best Practices: We will follow established coding guidelines and principles, enhancing code readability, reducing errors, and facilitating collaboration among team members.

### c. Code Review Process

- Branches will be named consistently like this: *feature-<initials>-<small details for the feature>*
- The developer who worked on the feature will create a PR for their feature branch to request to merge it into the current iteration branch.
- One of the teammates of the developer and the team lead will review the code and leave comments for the code. If code needs to be updated, developers will update the code as required.
- After reviewers have reviewed the code and approved the merge, the branch

will get merged by the configuration lead into the current iteration branch. (Before the merge all checks for the branch should pass and the code should have a proper coverage of unit tests.)

- After the current iteration is complete, reviewed, and tested (Smoke and Regression Suites will be run) by the QA Lead, it will be merged into the main branch by the configuration lead.

#### d. Testing

There will be multiple layers of testing happening in our development process.

- Test-Driven Development (TDD) will be practiced in our team. Our tests will be ready even before the implementation. With this, we will make sure that the requirements will be met.
- Unit testing for the code will be written by developers using Jest and will get merged when they're merging the branches. Unit testing will also be tested when PR is created.
- Manual test cases will be written for all of the features in Gherkin language (using Given, When, Then, And like we see in the class.)
- Automation testing should be done for those test cases that are crucial first then we can move to other features. (Smoke Suite and then Regression Suite.)
- Selenium-Behave with Python will be used for automation framework.
- Postman will be used for API testing. (And this will happen when API's are all done and there is enough data to test.)

#### e. Defect Management

For the defects that are found during testing the application;

1. We'll log a ticket in the pivotal tracker backlog and record it on GitHub.
2. This backlog will be reviewed by the whole team and those defects will get fixed in order as the developer would like.
3. After the defect is fixed;
  - a. The ticket will be tested and closed after it's verified.

## 7. References

[1] "TalentLMS," EPIGNOSIS. [online] Available: <https://www.talentlms.com>. [Accessed: 05/13/2024]

## 8. Glossary

(Any acronym used in the document should be explained here)