

CS673 Software Engineering
Team 4 - Project Portal
Software Test Document

Your project Logo
here if any

<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Brenna Mahn	Team Lead	<u>BM</u>	<u>5/28/2024</u>
Natanim Eibrahim	Security Lead	<u>NE</u>	<u>5/28/2024</u>
Savien Love	Design & Implementation Lead	<u>SL</u>	<u>5/28/2024</u>
Pinwen Mu	Configuration Lead	<u>PM</u>	<u>5/28/2024</u>

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
<u>1.0</u>	Fehmi Baltaci	<u>05/28/2024</u>	<u>First Draft</u>
<u>2.0</u>	<u>Fehmi Baltaci</u>	<u>06/09/2024</u>	<u>Second Draft</u>
<u>3.0</u>	<u>Fehmi Baltaci</u>	<u>06/18/2024</u>	<u>Third Draft</u>

[Project Overview](#)[Testing Summary](#)[Test Planning](#)[Objectives](#)[Scope](#)[Testing Approach](#)[Unit Testing](#)[Objectives](#)[Testing Approach](#)[Smoke Testing](#)[Objectives](#)[Testing Approach](#)[Regression Testing](#)[Objectives](#)[Manual Testing Report](#)[Test Cases](#)

- [1. User Google Sign-In](#)
- [2. Email and Password Sign-In](#)
- [3. Display User Projects](#)
- [3. View Project Information via Side Window](#)
- [4. Display Project Start Date](#)
- [5. Display Progress Comments \(?\)](#)
- [6. Display Project Team Members \(?\)](#)

[Automated Testing Report](#)[Testing Metrics](#)[References](#)[Glossary](#)

Project Overview

This document outlines the software testing strategy for our project management application. The application includes features such as user sign-in, project listing, project details viewing, and project progress tracking. The following sections describe the test plan, test cases, and the testing approach to ensure the application meets its requirements and functions correctly.

Testing Summary

Project management application we're working on is currently in the midst of building the automated testing framework and testing phase, with a focus on ensuring the application meets its functional and performance requirements. This report will summarize the testing activities that have been planned and those that will be

manually and also with automated framework tested.

Test Planning

Objectives

- Verify that users can sign in using Google, email, and password.
- Ensure the landing page displays the correct projects accessible to the user.
- Test the functionality of viewing project details via a side window and a detailed project page.
- Validate the project details, including start date, progress comments (?), and team members, are displayed correctly.

Scope

- Functional testing of user sign-in.
- UI testing of the landing page and project cards.
- Functional and UI testing of project details viewing.
- Validation of charts and detailed information on the project page.

Testing Approach

- Analysis of requirements and acceptance criteria will be evaluated.
- Definition of Done for all test cases.
- Manual testing will be performed for initial verification.
- Automated tests will be developed using Python with Behave and Selenium for UI automation.
- Automated tests will be developed using Postman for API Automation

Unit Testing

Objectives

- Validating the functionality of individual units of code.

Testing Approach

- Test Driven Development (TDD) will be used with Jest for React.js
- Developers are responsible for writing unit tests for their code.
- The unit tests will be run for every PR in GitHub and before merge they all must pass.

Smoke Testing

Objectives

- Evaluate the crucial functionality of the application.
- Critical functionality of the application will be tested using automation framework.

Testing Approach

- Smoke tests will be automated using Python-Behave-Selenium.
- Smoke tests will run before every deployment.

Regression Testing

Objectives

- To ensure new code has not affected the functionality of the application in a bad way.
- Regression testing will be done both using automation framework and Postman.

Manual Testing Report

In this section you will find scenarios will be tested for our application.

Test Cases

1. User Google Sign-In - CANCELLED

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Google Sign-In - High
Functionality	User can sign in with their google account
Preconditions	User should have a Google Account
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the sign in page 2. Click on Sign In with Google button. 3. Enter Google credentials 4. Verify that the user can login successfully to the application and the landing page is displayed.
Postconditions	Users successfully sign in to the application with their google account.
Expected Output / Actual Output	Users successfully sign in to the application with their google account.
Pass or Fail (If Failed: Bug ID)	Not Tested
Additional notes	

2. Email and Password Sign-In

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Email and Password Sign-In - High
Functionality	User can sign in with their email and passwords
Preconditions	User should have registered account with our

	application
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the sign in page 2. Click on the Sign In button. 3. Enter valid email and password 4. Verify that the user can login successfully to the application and the landing page is displayed.
Postconditions	Users successfully sign in to the application with their email and passwords.
Expected Output / Actual Output	Users successfully sign in to the application with their email and passwords.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

3. Display User Projects

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Display User Projects - High
Functionality	Users can see the projects in the landing page.
Preconditions	User should have signed-in
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Verify that the projects are displayed as cards.
Postconditions	Projects are correctly displayed as cards.
Expected Output / Actual Output	Projects are correctly displayed as cards.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	Please note that, there should be a data for the projects. If there are no projects in the system or user has no access to them, there won't be anything to be displayed.

3. View Project Information via Side Window

Test Case ID, Title, Priority (High/Medium/Low)	N/A - View Project Information on Side Window -
---	---

	High
Functionality	Users can see the project's brief information in a small side window that pops-up when clicked on the project card.
Preconditions	User is on the landing page with project cards displayed.
Test Steps	<ol style="list-style-type: none"> 1. Click on a project card. 2. Verify that a side window pops up displaying brief project information.
Postconditions	Side window with project information is displayed.
Expected Output / Actual Output	Side window with project information is displayed.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

4. Display Project Start Date - CANCELLED

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Display Project Start Date - High
Functionality	Users can see the projects in the landing page.
Preconditions	User is on the detailed project page.
Test Steps	<ol style="list-style-type: none"> 1. User clicks on the title of the project that is displayed on the side window, 2. New page is displayed with all of the information for the clicked project. 3. Verify that the project start date is displayed.
Postconditions	Project start date is correctly displayed.
Expected Output / Actual Output	Project start date is correctly displayed.
Pass or Fail (If Failed: Bug ID)	Not Tested
Additional notes	

5. Display Progress Comments (?) - CANCELLED

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Display Progress Comments - Medium
Functionality	Users can see the comments in the detailed project page.
Preconditions	User is on the detailed project page.
Test Steps	<ol style="list-style-type: none"> 1. User clicks on the title of the project that is displayed on the side window, 2. New page is displayed with all of the information for the clicked project. 3. Verify that there are comments in the comments section displayed successfully.
Postconditions	Project comments are displayed.
Expected Output / Actual Output	Project comments are displayed.
Pass or Fail (If Failed: Bug ID)	Not Tested
Additional notes	Please note that, there should be a comment posted before for the project, otherwise there won't be any comment displayed to the user.

6. Display Project Team Members (?) - CANCELLED

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Display Project Team Members - Medium
Functionality	Users can see the team members that are working on the project.
Preconditions	User is on the detailed project page.
Test Steps	<ol style="list-style-type: none"> 1. User clicks on the title of the project that is displayed on the side window. 2. New page is displayed with all of the information for the clicked project. 3. Verify that there are comments in the comments section displayed successfully.
Postconditions	Project members are displayed.
Expected Output / Actual Output	Project members are displayed.
Pass or Fail (If Failed: Bug ID)	Not Tested

Additional notes	
------------------	--

7. About Page

Test Case ID, Title, Priority (High/Medium/Low)	N/A - About Page Regression - High
Functionality	Display About Us and Mission Statement sections.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the About button. 3. Verify that the user is redirected to the About page. 4. Verify that the About Us section is displayed. 5. Verify that the Mission Statement section is displayed.
Postconditions	About Us and Mission Statement sections are displayed.
Expected Output / Actual Output	About Us and Mission Statement sections are displayed.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

8. Dashboard Page

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Dashboard Regression - High
Functionality	Display projects and search functionality.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the Dashboard button. 3. Verify that the dashboard page with projects is displayed. 4. Select Class 2 from the class dropdown. 5. Enter "AI" in the search bar.

	6. Click the submit button. 7. Verify that the search results for "AI" are displayed.
Postconditions	Projects and search results are displayed correctly.
Expected Output / Actual Output	Projects and search results are displayed correctly.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

9. Navigate to the About Page

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Navigate to About Page - High
Functionality	Navigate to About page from the landing page.
Preconditions	User is on the landing page.
Test Steps	1. Navigate to the landing page. 2. Wait for 5 seconds. 3. Click on the About button. 4. Wait for 5 seconds. 5. Verify that the user is redirected to the About page.
Postconditions	User is redirected to the About page.
Expected Output / Actual Output	User is redirected to the About page.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

10. Navigate to Dashboard Page

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Navigate to Dashboard Page - High
---	---

Functionality	Navigate to Dashboard page from the landing page.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Wait for 5 seconds. 3. Click on the Dashboard button. 4. Wait for 5 seconds. 5. Verify that the user is redirected to the Dashboard page.
Postconditions	User is redirected to the Dashboard page.
Expected Output / Actual Output	User is redirected to the Dashboard page.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

10. Navigate to Login Page

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Navigate to Login Page - High
Functionality	Navigate to Login page from the landing page.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Wait for 5 seconds. 3. Click on the Login button. 4. Wait for 5 seconds. 5. Verify that the user is redirected to the Login page.
Postconditions	User is redirected to the Login page.
Expected Output / Actual Output	User is redirected to the Login page.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

11. User visits the login page and logs in

Test Case ID, Title, Priority (High/Medium/Low)	N/A - User logs in with valid credentials - High
Functionality	User logs in with valid email and password.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the Login button. 3. Verify that the user is redirected to the Login page. 4. Verify that the Sign In button is displayed. 5. Verify that the Sign Up button is displayed. 6. Click on the Sign Up button. 7. Verify that the Sign Up Form is displayed. 8. Sign up with valid email and valid password "<password>". 9. Verify that the sign-in page is displayed. 10. Sign in with valid email and valid password "<password>". 11. Verify that the user is signed in successfully. 12. Click on the Profile button. 13. Verify that the email address is displayed on the Profile page.
Postconditions	User is signed in and email address is displayed on the Profile page.
Expected Output / Actual Output	User is signed in and email address is displayed on the Profile page.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	<ol style="list-style-type: none"> 1. Examples: <ul style="list-style-type: none"> • password • T9@bRg78BwhS&%

11. User visits the login page and tries to log in with invalid email and password

Test Case ID, Title, Priority (High/Medium/Low)	N/A - User tries to log in with invalid credentials -
---	---

	High
Functionality	User tries to sign up with invalid email and password.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the Login button. 3. Verify that the user is redirected to the Login page. 4. Verify that the Sign In button is displayed. 5. Verify that the Sign Up button is displayed. 6. Click on the Sign Up button. 7. Verify that the Sign Up Form is displayed. 8. Sign up with invalid email "<email>" and invalid password "<password>". 9. Verify that sign up is not successful.
Postconditions	Sign up is not successful.
Expected Output / Actual Output	Sign up is not successful.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	<ol style="list-style-type: none"> 1. Examples: <ul style="list-style-type: none"> • email password • invalidemail@123?email.com invalidpwd

11. Project Details Sidewindow

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Project Details Side Window - High
Functionality	Display project details in a side window.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the Dashboard button. 3. Verify that the dashboard page with projects is displayed. 4. Click on the first project card for more details. 5. Verify that the project details are displayed in

	a side window.
Postconditions	Project details are displayed in a side window.
Expected Output / Actual Output	Project details are displayed in a side window.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

11. Outline: Create a Project

Test Case ID, Title, Priority (High/Medium/Low)	N/A - Create a Project - High
Functionality	User creates a project with valid inputs.
Preconditions	User is on the landing page.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the landing page. 2. Click on the Login button. 3. Verify that the user is redirected to the Login page. 4. Verify that the Sign In button is displayed. 5. Verify that the Sign Up button is displayed. 6. Click on the Sign Up button. 7. Verify that the Sign Up Form is displayed. 8. Sign up with valid email and valid password "<password>". 9. Verify that the sign-in page is displayed. 10. Sign in with valid email and valid password "<password>". 11. Verify that the user is signed in successfully. 12. Click on Create button. 13. Verify that the form for creating a project is displayed. 14. Create a project with valid inputs. 15. Verify that the project is created successfully.
Postconditions	Project is created successfully.

Expected Output / Actual Output	Project is created successfully.
Pass or Fail (If Failed: Bug ID)	PASSED
Additional notes	

Automated Testing Report

We provide an overview of our automated testing efforts, including the location of test code in our repository, the test frameworks used, and any generated testing reports or screenshots.

- Everyone should test their code before creating a PR for merge.
- Testing methods include but are not limited to Python-Behave Framework, Postman, and Jest.
- Everyone is a manual tester in our team. They will test the front end by opening pages in the browser and checking error codes in the browser console, or checking the error code provided by IDE.
- Test codes are located under [./project-thisisteam4/automated-testing/](#)
- We aim to have as many automated tests as possible within the short time frame for our application. Manual test cases for both UI and API will also be documented within this document.

Testing Metrics

We'll report relevant metrics used for evaluation, such as the number of test cases, test coverage, defect rate, etc. These metrics will help us assess the effectiveness of our testing process and identify areas for improvement.

Metric Name	Description
Number of Test Cases	The total number of test cases written for the application including manual and automated tests.
Test Coverage	The percentage of functionality covered by testing.
Defect Rate	Number of Defects / Number of Total Test Cases
Pass/Fail Rate	Passed Test Cases / Failed Test Cases
Number of Automated Test Cases	Number of Automated Test Cases
Automation Rate	Number of Automated Test Cases / Number of Total Test Cases

References

- [Behave Documentation](#)
- [Selenium Documentation](#)
- [Jest Documentation](#)
- [Postman Documentation](#)
- [PEP 8 - Python Style Guide](#)
- [JavaScript Style Guide](#)

Glossary

We'll include a glossary of terms to ensure clarity and consistency in communication throughout the document. This glossary will define key terms and abbreviations used in the testing process.

Term	Definition
Acceptance Criteria	The conditions that a software product must satisfy to be accepted by a user or customer
Behavior-Driven Development (BDD)	A software development process that encourages collaboration among developers, QA, and non-technical participants
Test Driven Development (TDD)	A software development process where test cases are developed to specify and validate what the code will do
Unit Testing	Testing individual units or components of software
User Acceptance Testing (UAT)	A type of testing performed by the end-user or client to verify that the software meets the business requirements
Definition of Done (DoD)	A shared understanding of what it means for work to be complete, ensuring all criteria are met before a product increment is considered "done"