**PMTool Test Automation Suite**

**Objective** The objective of this test automation suite is to validate the functional requirements of the PMTool web application accessible via [PMTool web application link].

# Acceptance Criteria

# User Account Management

* + New users can successfully create an account with their name, email, and password. Optional fields include company and address.
  + Proper validation messages should be displayed for missing required fields during user creation or update.

# Project Management

* + Users can perform CRUD operations (Create, Read, Update, Delete) on projects.
  + Validation ensures unique project names and proper error handling for empty fields during project creation and editing.

# Task Management

* + Users can perform CRUD operations on tasks associated with specific projects.
  + Tasks can be assigned statuses (TO DO, IN PROGRESS, IN REVIEW, DONE) and can include labels and file attachments.
  + Validation checks for missing required fields (summary and description) during task creation and editing.

# TaskDB Functionality

* + TaskDB page allows users to view all tasks across projects.
  + Users can sort tasks by summary and perform searches, ensuring accurate results display.

# User Settings Management

* + Users can update personal settings including name, email, password, company, address, and enable two-factor authentication (2FA).
  + Validation ensures fields are not left blank during updates (name, email, password).

**Test Scenarios** The following test scenarios ensure comprehensive coverage of the PMTool application:

* **User Signup & User Login**
  + Includes successful signup with and without optional fields, and failure scenarios for missing email, password, and name.
  + Tests successful login, login failures due to invalid credentials, and password change scenarios.

# Project Management

* + Tests for adding, editing (name, description), and deleting projects, including cancelation of delete actions.

# Task Management

* + Covers various scenarios for adding tasks with different statuses, labels, and attachments.
  + Ensures validation for missing summary and description fields.

# TaskDB Functionality

* + Tests for viewing tasks, adding tasks from the dashboard, and verifying their presence in TaskDB.
  + Includes testing task deletion and ensuring their absence in TaskDB.

# User Settings Management

* + Tests for successful changes in name, email, password, company, address, and enabling 2FA.
  + Includes validation tests for blank entries in name, email, and password fields.

# Test Cases

Each test case below corresponds to the scenarios outlined above. Positive and negative cases are included to ensure comprehensive validation.

**1. User Signup & User Login**

## Test Case 1.1: Successful Signup without Optional Fields

### Description: Test successful signup with only required fields.

### Precondition: None.

### Steps:

* 1. Navigate to the signup page.
  2. Enter a random name, email, and password.
  3. Submit the signup form.

### Expected Result: The user sees a message indicating successful registration.

## Test Case 1.2: Successful Signup with Optional Fields

### Description: Test successful signup with required and optional fields.

### Precondition: None.

### Steps:

* 1. Navigate to the signup page.
  2. Enter a random name, email, password, company, and address.
  3. Submit the signup form.

### Expected Result: The user sees a message indicating successful registration.

## Test Case 1.3: Signup Failure due to Missing Email

### Description: Test signup failure when email is missing.

### Precondition: None.

### Steps:

* 1. Navigate to the signup page.
  2. Enter a random name and password.
  3. Leave the email field empty and submit the form.

### Expected Result: The user sees an error message indicating that the email field is required.

## Test Case 1.4: Signup Failure due to Missing Password

### Description: Test signup failure when password is missing.

### Precondition: None.

### Steps:

* 1. Navigate to the signup page.
  2. Enter a random name and email.
  3. Leave the password field empty and submit the form.

### Expected Result: The user sees an error message indicating that the password field is required.

## Test Case 1.5: Signup Failure due to Missing Name

### Description: Test signup failure when name is missing.

### Precondition: None.

### Steps:

* 1. Navigate to the signup page.
  2. Enter a random email and password.
  3. Leave the name field empty and submit the form.

### Expected Result: The user sees an error message indicating that the name field is required.

## Test Case 1.6: Successful Login with Valid Credentials

### Description: Test login with correct credentials.

### Precondition: The user must be registered and verified.

### Steps:

* 1. Navigate to the login page.
  2. Enter the registered email and password.
  3. Submit the login form.

### Expected Result: The user is redirected to the Dashboard page.

## Test Case 1.7: Login Failure due to Invalid Credentials

### Description: Test login failure with incorrect credentials.

### Precondition: None.

### Steps:

* 1. Navigate to the login page.
  2. Enter an invalid email and password.
  3. Submit the login form.

### Expected Result: The user sees an error message indicating invalid login info.

## Test Case 1.8: Login Failure due to Missing Email

### Description: Test login failure when the email field is empty.

### Precondition: None.

### Steps:

* 1. Navigate to the login page.
  2. Enter a password and leave the email field empty.
  3. Submit the login form.

### Expected Result: The user sees an error message indicating invalid login info.

## Test Case 1.9: Login Failure due to Missing Password

### Description: Test login failure when the password field is empty.

### Precondition: None.

### Steps:

* 1. Navigate to the login page.
  2. Enter a registered email and leave the password field empty.
  3. Submit the login form.

### Expected Result: The user sees an error message indicating invalid login info.

## Test Case 1.10: Successful Login after Password Change

### Description: Test login after successfully changing the password.

### Precondition: The user must be registered and verified.

### Steps:

* 1. Log in with the current credentials.
  2. Navigate to the settings page and change the password.
  3. Log out and attempt to log in with the new password.

### Expected Result: The user can log in with the new password and sees a success message.

**2. Project Management**

## Test Case 2.1: Add New Project Successfully

### Description: Test the successful addition of new projects.

### Precondition: User must be logged in.

### Steps:

* 1. Navigate to the projects page.
  2. Add a new project with a name and description.
  3. Verify the project appears in the project list.

### Expected Result: The new project is listed with the correct name and description.

## Test Case 2.2: Edit Project Name and Description Successfully

### Description: Test the successful editing of a project's name and description.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to edit.
  2. Change the project name and description.
  3. Save the changes and verify the project details.

### Expected Result: The project shows the updated name and description.

## Test Case 2.3: Edit Project Name Successfully

### Description: Test the successful editing of a project's name only.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to edit.
  2. Change the project name.
  3. Save the changes and verify the project details.

### Expected Result: The project shows the updated name and retains the original description.

## Test Case 2.4: Edit Project Description Successfully

### Description: Test the successful editing of a project's description only.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to edit.
  2. Change the project description.
  3. Save the changes and verify the project details.

### Expected Result: The project retains the original name and shows the updated description.

## Test Case 2.5: Edit Project with No Changes

### Description: Test editing a project without making any changes.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to edit.
  2. Save the project without changing any details.
  3. Verify the project details remain unchanged.

### Expected Result: The project details remain unchanged.

## Test Case 2.6: Delete Existing Project Successfully

### Description: Test the successful deletion of an existing project.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to delete.
  2. Confirm the deletion.
  3. Verify the project is removed from the project list.

### Expected Result: The project no longer appears in the project list.

## Test Case 2.7: Cancel Delete Existing Project

### Description: Test cancelling the deletion of an existing project.

### Precondition: At least one project exists.

### Steps:

* 1. Select an existing project to delete.
  2. Cancel the deletion.
  3. Verify the project remains in the project list.

### Expected Result: The project remains in the project list.

**3. Task Management**

## Test Case 3.1: Add task with required fields only:

### Description: Test adding a task with only summary and description fields filled.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

* 1. Navigate to the project where the task needs to be added.
  2. Add a task with a summary and description.
  3. Verify that the task is successfully added with the default status 'TO DO'.

### Expected Result: The task is successfully added with default status 'TO DO'.

## Test Case 3.2: Add task with 'TO DO' status:

### Description: Test adding a task with 'TO DO' status.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task specifying 'TO DO' status.
3. Verify that the task is successfully added with 'TO DO' status.

### Expected Result: The task is successfully added with 'TO DO' status.

## Test Case 3.3: Add task with 'IN PROGRESS' status:

### Description: Test adding a task with 'IN PROGRESS' status.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task specifying 'IN PROGRESS' status.
3. Verify that the task is successfully added with 'IN PROGRESS' status.

### Expected Result: The task is successfully added with 'IN PROGRESS' status.

## Test Case 3.4: Add task with 'IN REVIEW' status:

### Description: Test adding a task with 'IN REVIEW' status.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task specifying 'IN REVIEW' status.
3. Verify that the task is successfully added with 'IN REVIEW' status.

### Expected Result: The task is successfully added with 'IN REVIEW' status.

## Test Case 3.5: Add task with 'DONE' status:

### Description: Test adding a task with 'DONE' status.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task specifying 'DONE' status.
3. Verify that the task is successfully added with 'DONE' status.

### Expected Result: The task is successfully added with 'DONE' status.

## Test Case 3.6: Add task with random labels:

### Description: Test adding a task with random labels.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task with random labels.
3. Verify that the task is successfully added with specified labels.

### Expected Result: The task is successfully added with specified random labels.

## Test Case 3.7: Add task with all available labels:

### Description: Test adding a task with all available labels.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task with all available labels.
3. Verify that the task is successfully added with all available labels.

### Expected Result: The task is successfully added with all available labels.

## Test Case 3.8: Add task with one uploaded file:

### Description: Test adding a task with one uploaded file.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task with one uploaded file.
3. Verify that the task is successfully added with the specified file.

### Expected Result: The task is successfully added with the specified file.

## Test Case 3.9: Add task with two uploaded files:

### Description: Test adding a task with two uploaded files.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Add a task with two uploaded files.
3. Verify that the task is successfully added with the specified files.

### Expected Result: The task is successfully added with the specified files.

## Test Case 3.10: Add task with missing task summary:

### Description: Test validation for missing summary field during task creation.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Attempt to add a task without a summary.
3. Verify that an error message indicating the summary field is required is displayed.

### Expected Result: An error message indicating the summary field is required is displayed.

## Test Case 3.11: Add task with missing task description:

### Description: Test validation for missing description field during task creation.

### Precondition: User is logged in and navigated to the project where the task needs to be added.

### Steps:

1. Navigate to the project where the task needs to be added.
2. Attempt to add a task without a description.
3. Verify that an error message indicating the description field is required is displayed.

### Expected Result: An error message indicating the description field is required is displayed.

## Test Case 3.12: Viewing tasks:

### Description: Test viewing tasks after adding them.

### Precondition: User is logged in and navigated to the project where tasks were added.

### Steps:

1. Navigate to the project where tasks were added.
2. Verify that all added tasks are displayed correctly.

### Expected Result: All added tasks are displayed correctly.

## Test Case 3.13: Delete task successfully:

### Description: Test deleting a task.

### Precondition: User is logged in and navigated to the project where the task exists.

### Steps:

1. Navigate to the project where the task exists.
2. Delete the task.
3. Verify that the task is removed from the task list.

### Expected Result: The task is successfully deleted and no longer appears in the task list.

## Test Case 3.14: Cancel delete task:

### Description: Test cancelling the deletion of a task.

### Precondition: User is logged in and navigated to the project where the task exists.

### Steps:

1. Navigate to the project where the task exists.
2. Attempt to delete the task but cancel the action.
3. Verify that the task remains in the task list.

### Expected Result: The task remains in the task list after cancelling the deletion action.

## Test Case 3.15: Editing tasks:

### Description: Test editing an existing task.

### Precondition: User is logged in and navigated to the project where the task exists.

### Steps:

1. Navigate to the project where the task exists.
2. Edit the task's summary, description, status, labels, and associated files.
3. Verify that changes are reflected correctly.

### Expected Result: Changes made to the task (summary, description, status, labels, and files) are reflected correctly.

## Test Case 3.16: Edit task with missing task summary:

### Description: Test validation for missing summary field during task edit.

### Precondition: User is logged in and navigated to the project where the task exists.

### Steps:

1. Navigate to the project where the task exists.
2. Attempt to edit a task without a summary.
3. Verify that an error message indicating the summary field is required is displayed.

### Expected Result: An error message indicating the summary field is required is displayed.

## Test Case 3.17: Add task with missing task description:

### Description: Test validation for missing description field during task edit.

### Precondition: User is logged in and navigated to the project where the task exists.

### Steps:

1. Navigate to the project where the task exists.
2. Attempt to edit a task without a description.
3. Verify that an error message indicating the description field is required is displayed.

### Expected Result: An error message indicating the description field is required is displayed.

**4. TaskDB Page Functionality**

## Test Case 4.1: View Tasks in TaskDB:

### Description: Test viewing tasks in the TaskDB page.

### Precondition: User is logged in and navigated to the TaskDB page.

### Steps:

1. Navigate to the TaskDB page.
2. Verify that all expected tasks ("A\_Task", "B\_Task", "$\_Task", "Z\_Task") are present.

### Expected Result: All expected tasks are displayed correctly on the TaskDB page.

## Test Case 4.2: Add Task and Verify in TaskDB:

### Description: Test adding a task from the dashboard and verifying its presence in the TaskDB.

### Precondition: User is logged in and navigated to the dashboard.

### Steps:

1. Add a new task ("Just\_Added") from the dashboard.
2. Navigate to the TaskDB page.
3. Verify that the newly added task ("Just\_Added") is present in the TaskDB.

### Expected Result: The newly added task ("Just\_Added") appears correctly in the TaskDB page.

## Test Case 4.3: Remove Task and Verify Absence in TaskDB:

### Description: Test deleting a task from the dashboard and verifying its absence in the TaskDB.

### Precondition: User is logged in and navigated to the dashboard.

### Steps:

1. Delete the task ("Just\_Added") from the dashboard.
2. Navigate to the TaskDB page.
3. Verify that the deleted task ("Just\_Added") is no longer present in the TaskDB.
4. Refresh the TaskDB page and verify again for the absence of the deleted task.

### Expected Result: The deleted task ("Just\_Added") is not found in the TaskDB after deletion.

## Test Case 4.4: Sort Tasks by Summary:

### Description: Test sorting tasks by summary in ascending and descending order.

### Precondition: User is logged in and navigated to the TaskDB page.

* **Steps for Ascending Order:**

1. Click on the "Sort by Summary" button to sort tasks in ascending order.
2. Wait for the page to load after sorting.
3. Retrieve the titles of the tasks after sorting.

* **Expected Result (Ascending Order):** Tasks are displayed in ascending order based on their summary.
* **Steps for Descending Order:**

1. Click on the "Sort by Summary" button again to sort tasks in descending order.
2. Wait for the page to load after sorting.
3. Retrieve the titles of the tasks after sorting.

* **Expected Result (Descending Order):** Tasks are displayed in descending order based on their summary.

## Test Case 4.5: Search Tasks:

### Description: Test searching for tasks in the TaskDB page.

### Precondition: User is logged in and navigated to the TaskDB page.

* **Steps for Individual Searches:**

1. Search for each task ("A\_Task", "B\_Task", "$\_Task", "Z\_Task") individually.
2. Verify that search results match the expected task.

* **Expected Result (Individual Searches):** Each search returns the respective task as expected.
* **Steps for Non-Existent Task Search:**

1. Search for a non-existent task ("Non\_Existent\_Task").
2. Verify that no results are returned.

* **Expected Result (Non-Existent Task Search):** No tasks are displayed as a result of the search.

**5. User Settings Management**

## Test Case 5.1: Successful Name Change

### Description: Verify that the user can successfully change their name.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Change the user's name.
  3. Save changes and verify success message.

### Expected Result: The user sees a success message confirming the name change and the updated name is displayed in settings.

## Test Case 5.2: Successful Email Change

### Description: Verify that the user can successfully change their email.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Change the user's email.
  3. Save changes and verify success message.

### Expected Result: The user sees a success message confirming the email change and the updated email is displayed in settings.

## Test Case 5.3: Successful Password Change

### Description: Verify that the user can successfully change their password.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Change the user's password.
  3. Save changes and verify success message.
  4. Logout and login again with the new password.

### Expected Result: The user sees a success message confirming the password change and can successfully log in with the new password.

## Test Case 5.4: Successful Company and Address Change

### Description: Verify that the user can successfully change their company and address.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Change the user's company and address.
  3. Save changes and verify success message.

### Expected Result: The user sees a success message confirming the company and address change and the updated company and address are displayed in settings.

## Test Case 5.5: Successful 2FA Enablement

### Description: Verify that the user can successfully enable 2FA.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Enable 2FA setting.
  3. Save changes and verify success message.

### Expected Result: The user sees a success message confirming the 2FA enablement and 2FA is shown as enabled in settings.

## Test Case 5.6: Attempt to Change Name to Blank

### Description: Verify that the user cannot change their name to blank.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Attempt to change the name field to blank.

### Expected Result: The user sees a validation error message indicating that the name field is required.

## Test Case 5.7: Attempt to Change Email to Blank

### Description: Verify that the user cannot change their email to blank.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Attempt to change the email field to blank.

### Expected Result: The user sees a validation error message indicating that the email field is required.

## Test Case 5.8: Attempt to Change Password to Blank

### Description: Verify that the user cannot change their password to blank.

### Precondition: User is logged in and on the Settings page.

### Steps:

* 1. Navigate to the Settings page.
  2. Attempt to change the password field to blank.

### Expected Result: The user sees a validation error message indicating that the password field is required.

**Automated Test Scripts**

**Tools and Libraries**

* **Programming Language**: Python
* **Framework**: Selenium for UI tests

**Instructions to Execute Automated Tests**

1. **Install WebDriver**: Ensure you have the appropriate WebDriver for your browser installed (e.g., ChromeDriver for Google Chrome).
2. **Set Up Environment**:
   * Install necessary libraries:  
     pip install .\pm\_tool\_tests/requirement.txt
3. **Run Scripts**: Execute the test scripts from your IDE or command line:
   * pytest --html=report.html .\pmtool\tests\
4. **Review Results**: Check the console output and logs for test results and any assertion errors.