

INSTRUCTIONS:

1. Create 1 document per team, within the milestones folder in the project directory, that describes how, at least, 3 features within your finished product will be tested.
2. The test plans should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.
3. The test plans should include a description of the test data that will be used to test the feature.
4. The test plans should include a description of the test environment that will be used to test the feature.
5. The test plans should include a description of the test results that will be used to test the feature.
6. The test plan should include information about the user acceptance testers.

LOGIN

In order to test the /login endpoint, the test case will send the server a POST request with the following data:

- Email
- Password

The positive test case will send a valid email and password, and the server will return a valid status and the user's `user_id`.

The negative test case will send either an invalid email or an invalid password. The server will return an error, with a message stating that the "Email or password is incorrect".

In order to test this feature, our PostgreSQL database has been prepopulated with dummy data concerning users. Therefore, for the positive test case, we will use the user created during the /register test case. For the negative test case, we will use a user from the dummy data with an incorrect password.

We will be using the testing frameworks mocha and chai to test the login feature.

User Acceptance Criteria:

- It is mandatory for a user to enter their email address and password.
- If the email exists in the *users* table, then the password will be verified.
 - If the password is verified, the server will return a success message.
 - If the password is invalid, the server will return a failure message.

REGISTER

In order to test the /register endpoint, the test case will send the server a POST request with the following data:

- Email
- Password

The positive test case will send a valid email and password, and the server will return a valid status and insert a new user with the information given into the user query.

The negative test case will send either an invalid email or already taken username. The server will return an error, with a message stating that there was an “Error registering user, please enter a unique username and email”.

In order to test this feature, our PostgreSQL database has been prepopulated with dummy data concerning users. Therefore, for the positive test case, we will use new data not already in the database . For the negative test case, we will use invalid email addresses, and emails and usernames that already exist in the database.

We will be using the testing frameworks mocha and chai to test the register feature.

User Acceptance Criteria:

- It is mandatory for a user to enter a valid and unique email address and a password.
- If the email is not a valid email address, the server will return an error message
- If the email or username already exists in the *users* table, the server will return an error message

POPULATE PROFILE

In order to test the /register endpoint, the test case will send the server a POST request with the following data:

- Bio
- Classes
- Times Available
- Contact Info

The positive test case will send the following data, and the server will return valid status and insert the given information into the user query and display it on the user’s profile page

The negative test case will send either invalid class/classes or invalid available time. The server will return an error with a message stating “Error adding your classes, please enter a valid class name” and/or “Error adding your available times, please enter valid times”

In order to test this feature, our PostgreSQL database has been prepopulated with dummy data concerning users. Therefore, for the positive test case, we will use new data not already in the database . For the negative test case, we will use invalid classes and times available that already exist in the database.

We will be using the testing frameworks mocha and chai to test the populate profile feature.

User Acceptance Criteria:

- It is not mandatory for a user to enter information into any of the fields but if they input times or classes they must be valid
- If either the times or classes are invalid the server will return an error message