How to run tests –
Unit tests are written and tested using mocha test framework. This framework allows to create unit tests for testing functions and APIs and validating the correct outcome.
The test files mentions all the functions and what the test should do as an output.
Test files are dependent on each other. As tests require authorization, we need to have both managers and developers in the database so next tests can be done.
Run the below test files in order -

1. Register.test.js-

This is a test file for the developer and manager registration feature in the application. The test uses the Chai assertion library and Supertest library to simulate HTTP requests to the application and verify the responses. The test checks if a new developer or manager can be registered successfully and if the registered user is added to the database and redirected to the correct page.
To run this test -

1. Ensure that the dependencies for testing, such as Chai and Supertest, are installed in your project.
2. In the terminal, navigate to the folder /tests of your project.
3. Run the test file using the command - npx mocha register.test.js
4. The test results will be displayed in the terminal.

(To run the tests below, your database needs to have the registered developer and manager from the register test)
2. Dev.test.js -

This is a test file for the authentication and authorization of developers while accessing the "loadHome" endpoint. It uses the "supertest" library to make HTTP requests to the endpoint and the "chai" library to write assertions for the expected behavior of the endpoint.
The first test case logs in a developer by making a POST request to the "/login/developer" endpoint with the developer's credentials. The second test case checks if the endpoint returns a 200 status code when the user is authenticated. It sets the developer token obtained from the previous test case in the "Cookie" header of the HTTP request. The third test case checks if the endpoint returns a 401 status code when the user is not authenticated. It does not set any authorization header in the HTTP request.

5. Ensure that the dependencies for testing, such as Chai and Supertest, are installed in your project.
6. In the terminal, navigate to the folder /tests of your project.
7. Run the test file using the command - npx mocha dev.test.js
8. The test results will be displayed in the terminal.

3. Chatroom.test.js

These tests are testing the Manager login and Chat Room API routes of the application. The first test is testing the Manager login functionality by sending a POST request to "/login/manager" with valid email and password. The test checks if the response status is 200 and sets the managerToken for use in subsequent tests.

The second test is testing the case where the email or password is invalid, expecting a 400 response status code.
The second part of the test suite tests the Chat Room API routes. The first test is testing the creation of a new chat room by sending a POST request to "/chatRoom/create" with a chatRoom object and managerToken set in the request headers. The test expects the response status code to be 201, and for the response body to have an "_id" property, a "name" property matching the "name" property of the chatRoom object sent in the request, and a "description" property matching the "description" property of the chatRoom object sent in the request.
The test then checks if the chat room was successfully saved to the database by checking for the existence of the saved chat room with the same name, description, and owner.
The second test in the Chat Room API test suite tests the case where the user is not authorized, by sending a POST request to "/chatRoom/create" without setting the managerToken in the request headers. The test expects a 401 response status code.

9. Ensure that the dependencies for testing, such as Chai and Supertest, are installed in your project.
10. In the terminal, navigate to the folder /tests of your project.
11. Run the test file using the command - npx mocha chatRoom.test.js
12. The test results will be displayed in the terminal.