

Automatic Testing

Frontend Testing

The structure for our testing on the frontend is relatively simple. We write tests for each individual React component in its own separate file at the same level. For testing on the frontend of our application, we have two main external libraries. First up is Jest, which is a test runner and assertion library that we use to create, run, and automate tests. The other library is Enzyme, which is specifically used for programmatically simulating user input on React web applications. We also use some smaller libraries like sinon to help create mock functions that can be spied on (check inputs, how many times it was called, etc). About 75% of our test plan is fully automated and can be run by using the command `npm test` when in the root directory of the project. The other 25% is manually tested by a human because it was too complicated to mock all of the output of the Google Maps API. Anything that has to deal with the map and the pins on the map is unfortunately stuck in manual testing.

Backend Testing

Tests for the backend are stored in a separate testing folder where they are then split up by module. We use Jest again on the backend for testing because it is easy to understand and quick to get up and running. We also make use of smaller libraries like SuperTest to help with making HTTP requests from the automated test cases. All of our backend tests are completely automated, because there was nothing finicky to test like there was on the frontend. We completely cover tests on authentication, account functions like saving and rating, and search functions. You can use the exact same `npm test` command to run the tests for our backend code. The only thing is that our backend code and frontend code are separated into 2 different repositories, so you need to individually test each of them.