

Stratify Automatic Testing Documentation

- Third party products
 - Mocha and Chai in conjunction with supertest are used for automatic testing of server side functions.
 - Jasmine and Karma are used for automatic testing of client side functions.
- Implementation
 - Backend
 - The project's public directory contains a test subfolder that has individual files containing categorically organized tests. All backend tests can be run with a single command and their output as well as success/failure status are displayed in the terminal. The test_helper file contains code that assists in the execution of all tests. Mongoose allows us to connect to and interact more easily with mongodb. Supertest allows us to test http requests easily. Mocha drives our main testing and Chai is used for more advanced assertions.
 - Frontend
 - Tests are implemented inside each component's .spec.ts file. All frontend tests can be run with a single command and their output as well as success/failure status are displayed in the browser's console.
- Coverage:
 - Backend tests cover all functions located in the app.js file. This means that our backend tests have full coverage of server-side functions. Frontend tests ensure components are rendering correctly and calling functions appropriately. All front-end components are covered.
- Running tests:
 - Backend
 - Install mocha, nodemon, mongoose, chai, and supertest are all installed.
 - Navigate to your mongodb bin directory, ~/mongodb-community@4.0/4.0.12/bin

- Run the following command: `./mongod`
 - This will start the local instance of the database.
- Inside the public directory run: `ng build --watch`
- Inside the public directory run: `npm run testMongo`
 - Note: it is important that you do not have nodemon running when the tests are executed.

■ Frontend

- Run “ng test” in the public directory.
- What parts of your system still have to be tested manually:
 - None