🧪 Testing And Project Setup

Contents

Project Setup

This project uses the MERN stack. Node is assumed to already be installed, for official node instillation refer to Node.js — Download Node.js®

1. Clone or fork the repository on GitHub.
2. First set up the front end, navigate the path **client/app** and in the app folder, open terminal and do **npm install** to install all the front end package dependencies.
3. Set up the backend by going back to the root directory. Then navigate the path **/server** and do **npm install**.
4. Before further setup ensure that you have flake8, scikit-learn, numpy, and pandas installed using pip install. These are relevant python libraries that the project relies on. If you do not have python installed, then you must install it first.
5. Create a **.env** file in the /server directory. Inside the .env file structure it as follows

```
DATABASE_URI =
PORT =
AI_API_KEY=
ACCESS_TOKEN_SECRET = ''
REFRESH_TOKEN_SECRET = ''
```

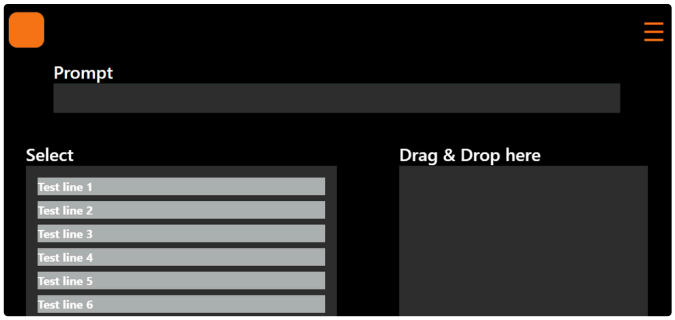
5. Now create a MongoDB Atlas account and create a free cluster and collection. For more information MongoDB Atlas Database | Multi-Cloud Database Service
6. In MongoDB Atlas, create a new key for the database collection.
7. Copy the MongoDB Atlas key URL and paste into the DATABASE_URI in the .env file.
8. Enter some available port in the PORT variable of the .env file.
9. Create a new Gemini AI account and generate an API key see Get a Gemini API key | Google AI for Developers for more information.
10. Copy the AI key into the AI_API_KEY variable in the .env file. Now the set up is complete.
11. To run the project, open two terminal instances. On the first terminal, navigate into client/app and type **npm run start** On the second terminal navigate to /server where all the backend files are and type **node index.js**

Front End Testing Expectations

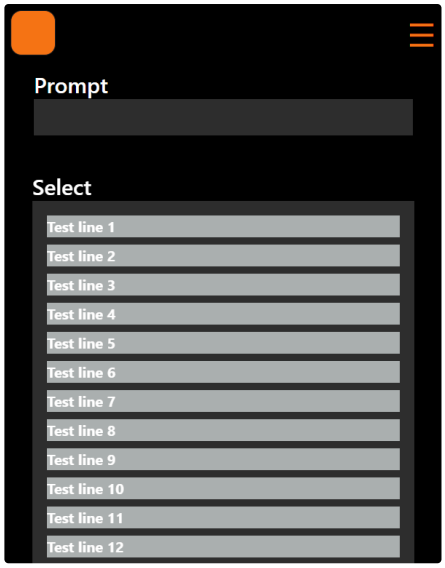
Expectations

1. Responsive test. Frequent checks must be made to ensure that pages remain responsive across different screen sizes.

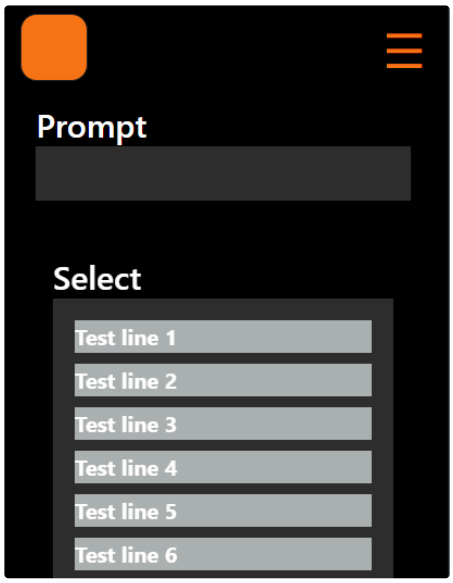
Desktop viewport range: **~800 -1000 pixels**



Tablet viewport range: **~500 -790 pixels**



Mobile viewport range: **~330 -490 pixels**



2. Interactive test. Every interactable component must support touch, and mouse events.

3. Browser test. Ensure that all pages and features are supported for the three major web-browsers repeat steps 1.) and 2.) for each browser.

Chrome

Firefox

Edge

Front end testing methods

For front end testing use it is sufficient to use the developer tools provided by each major browsers being tested. See Front end testing methods for frontend testing expectations.

Example: Chrome dev tool

Chrome DevTools | Chrome for Developers

1. Responsive test

   I.) Find the following navigation bar in the top left corner 

Enter the viewport size range in the input boxes in pixel units.

2. Interactive test

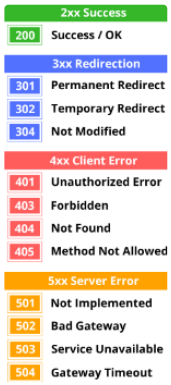
   I.) Find the following navigation bar in the top right corner 

Click the  icon to enable and disable touch events. When the icon is disabled, mouse events will be enabled and your mouse cursor will be visible.

Back End Testing Expectations

Expectations

1. Error codes and asserts. It is expected that asserts and HTTP error codes are placed throughout backend code to ensure ease of debugging.



2. Route and request test. Routes should be tested for all HTTP request types placed onto the route.

Backend testing methods

For back end expectations see Backend testing methods

1. Error codes and asserts

   I.) Error codes and assert statements should be self explanatory and implemented in the code itself.

2. Routes and request test

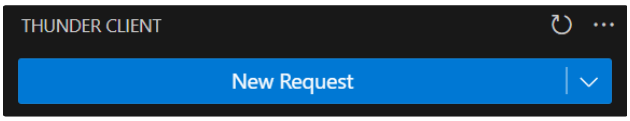
   Example: Thunder Client

   The following tool will be used for testing, and tool support is on VS code.

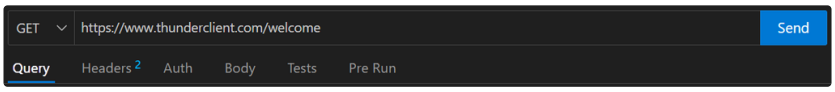
   Thunder Client - Rest API Client Extension for VS Code

After installing thunder client via VSCode extensions, navigate to the icon on VSCode left sidebar.

I.) On the top left, click New Request



II.) Select the HTTP method type, enter the URL. Click “Body“ and enter test cases for the body of the request object.



III.) The following are example tests