Code Refactoring Report

- Added Integration tests
- Added more unit tests

//— Ethan Herpich

After the second sprint, there were two major areas in the front end of the code that were identified as requiring refactoring. Firstly, the UI of the site required a few small tweaks to promote consistency between pages in the application and to improve the user expeirence. Secondly, there were many bugs that were occuring, such as css not loading, having duplicate navbars, and typescript build errors.

Code refactoring in the front end consisted of a few cosmetic and structural changes. Of the cosmetic changes:
- The navbar was made larger, and given box shadow to increase visual separation from each page
- The background image was changed to a hero banner on the landing page
- A category section was created on the landing page
- The search button was removed, and instead the user could activate their search by pressing enter after inputting the keywords into the search bar

In an effort to increase code and visual uniformity across the application, the search by category functionality and UI was made consistent by:
- Reusing similar UI features
- Reusing the same page switching mechanism (re-routing through next.js)
- Reusing the same page component to display results

There was also an investigation into why the website was not formatting properly and built with 'npm run dev'. It was discovered that not all formatting when running the development server but instead, would be applied after building the application properly.

To better demonstrate the functionality of the application six more products were also hardcoded into the database.

Finally, much of the code was refactored to address Typescript build errors and compilation errors. This enabled us to then successfully dockerise the application and use CI appropriately.

—//