CS 362 In-Class Exercise: Project Beta Testing

Your name: Justice Peyton (Team 5)

Project that you are testing: Team 6 (project_team_6)
https://github.com/Software-Engineering-II-Project-Group-6/Group-6-Project

PART-1: Organization and Purpose

Q1) Does the repository provide a README explaining the purpose of the software? If yes, based on reading that documentation, do you understand all the interesting features provided by the software? Do you have any advice to improve that documentation?

Their README provides an explanation of the software's purpose and features– NourishQuest is a web application that gamifies dieting and nutrition planning. The application also has multiplayer through leaderboards.

The main README is nice and short, making it digestible (quite) and approachable.
I admire the use of GitHub's wiki//documentation feature and it's personally made me want to try utilizing it for our own project.

I think perhaps elaborating more on what exactly the application does with bullet marks of major features with small descriptions would also help– the general README at the very front of the repository is quite short, but it makes the application seem very generic.

PART-2: Installation and Setup

Q2) Is the documentation to install or setup the software available? (Note that for web application, it would be a URL to access the website and instructions to host the website on a server). When following the instructions, do you face any difficulties while installing the software (accessing the URL for a website)? If yes, please explicitly state what issues you encountered, so that the project team can fix them.
NOTE: If you are testing a web application, then you do not need to set up a web server and try hosting the web application. Just go through the documentation to find out if it clearly explains the steps to host the website.

While this is a web application, no URL was given for the site, so I presumed that I'd need to host it locally. There is documentation for installing and setting up the software and it is rather simple and straightforward to do. I managed to set it up without difficulty, but there were a few errors from NPM complaining about unsupported and outdated packages.

I think it should be explicitly listed what the minimum versions for NPM and Node are for the site (for simplicity, this could just be the versions the team has been using). It would also help for those who are inexperienced to explicitly state that they need to change directories into "/NourishQuest/" and then run NPM commands– purely for those who aren't experienced with NPM.

Additionally, I recommend using "flip" to host your server– it'd be hosted at "http://flip1.engr.oregonstate.edu:3000" if you use it there on port 3000.

PART-2: Functional and Non-Functional Testing

Q3) Select a use case for the application-under-test and use your creativity to test the application in different possible ways. For example, if you are testing a login functionality, then test the sign up feature, sign in, adding invalid credentials, special characters, etc. Please provide the details of the use case you tested on the software by describing exactly what all you did and in what order? Make sure you are making notes while doing this. If you find any issues (e.g., something that was confusing, incorrect, or not working at all), please provide as many details as you can to replicate the issues so that the team can fix them.

Use Case 2

I decided to go over Use Case 2 as specified by “project_proposal.md”. This use case goes over signing up on their website.

Step 1 – user clicks sign up button

I attempted to forcibly visit other sections of the site while not logged in just to try to see if I could attempt to sign up in odd places– but the site appears to always redirect the client to “/login” when the client isn’t logged in. Regardless, all works fine and I am redirected to the sign up form when I press the “sign up” button.

Step 2 – user fills out sign up form, hits submit

Just outright submitting without filling anything out causes an internal server error– perhaps it would be good to cleanly handle the case where the client attempts to submit dud data.

I attempted to enter a decimal value for “Age”, and it cleanly handled it and told the client to enter an integer.

However, I was able to enter negative values for “Age”, “Height”, and “Weight”. I am also able to enter nonsensically large numbers for all of these entries.

Step 3 – form input is validated client-side

Nothing actionable here. I would like to warn that validation on the client side might be easily manipulated by savvy clients.

Step 4 – if valid, form input is sent to server

Step 5 – form input is validated and formatted server side

Step 6 – form input is inserted into the database by the server

This appears to be functioning. The newly created user is present in the Leaderboard, can be logged into, and has a profile with the values of “Age”, “Height”, and “Weight” displaying there.

Step 7 – user is logged in to the newly created account

This also works as intended.

However, I am able to go to the “login” page forcibly by changing the URL or by going back to previous pages– meaning that I have a login page present while also being logged in with the “log out” button present on the top right.