Section 015 Team 7

OptimizedHealth

Vignesh Chandrasekhar, Evie Lee, Leah Dillard, Finbar Forward, Jamal Giornazi

Quick Overview

Our OptimizedHealth application is an all-in-one stop for users to check up on their overall health and evaluate their current progress, as well as allow users to access nutritional information. Users can register and gain access to their profile page as well as nutrition, sleep and fitness trackers and calculators. With this, they unlock a wide variety of nutritional and fitness information- and a way to log their sleep patterns- all tailored to meet their personal goals.

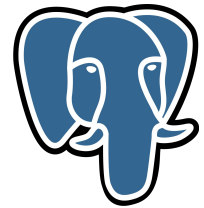
Languages

Incorporating HTML, JS, and CSS allowed us to utilize both frontend style and backend functionality

Tools/Frameworks

Zoom: Very effective in terms of communication and keeping up to date on the project, but had minor connection issues throughout the semester.

Jira



PostgresQL

jQuery: For writing better and cleaner javascript

Localhost: We used this deployment environment because it was very effective in aiding the loading/ viewing of our application

VS Code

GitHub

Jira: Project tracker that we used to keep track of sprints/ deadlines

VSCode: Primary IDE

Node.js/
Node.js with
Passport
Authentication

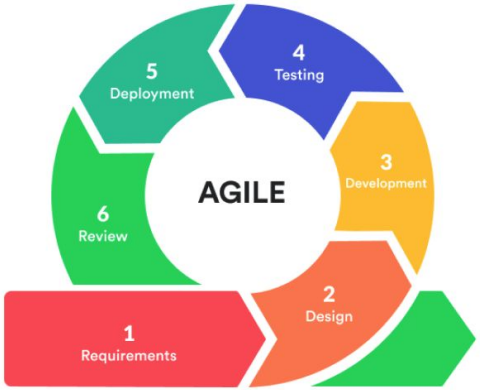
Node.js: Framework for javascript; we also used Node.js with Passport Authentication for user login and registration authentication.

PostgresQL: Primary database

GitHub: VCS Repository we used to save and record our progress throughout the semester.

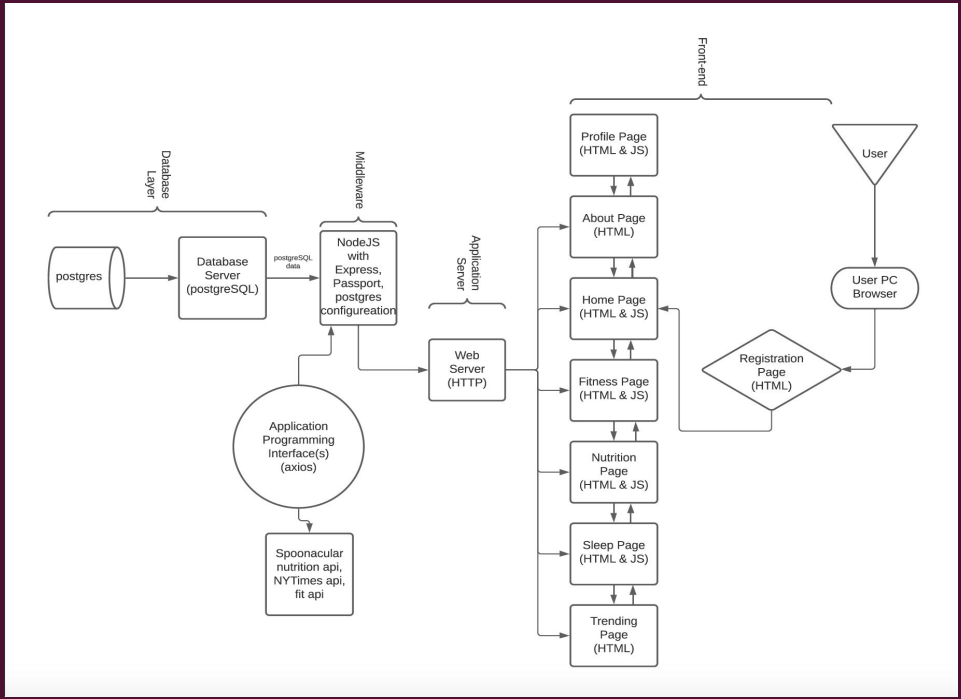
API: Spoonacular nutrition API(2X), New York Times Articles API, Fit API

Methodology: Agile

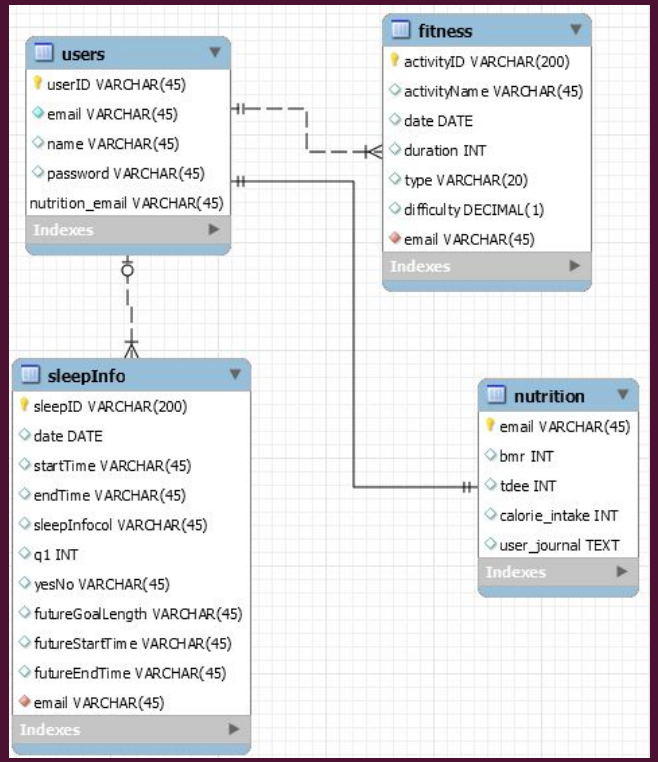


- Developed sprints and user stories
- Create wireframes to aid in design
- Assigned story points to each issue/feature that needed to be implemented
- Conducted peer code reviews
- Conducted Zoom meetings for updates and planning
- Developed user acceptance tests
- Tested deployment on localhost
- Reviewed current progress and planned accordingly

Architecture Diagram



Entity Relationship Diagram (ERD)



Challenges

- **Learning how to connect the postgres database to our node server**
  - Overcame by referencing labs and going to office hours in addition to watching Youtube videos
  - This did not affect our project plans, but overcoming it delayed production of relevant pages
- **Time management (completing all of our relevant ejs pages along with pairing them to the server and/or database)**
  - The many pages on our site allowed each team member to work on at least one page/feature
  - However, we were not able to complete all of the features we would have liked to include(such as uploading user profile image, and automated feedback from system)
- **Using GitHub to pull and merge from and to other branches**
  - Every team member had their own branch, but combining our work led to merge conflicts. We overcame this by researching how to resolve the issue online.
  - Did not affect project plans
- **Styling our site to conform to some scheme**
  - Had trouble with finding a solid layout, but it worked itself out.
  - Did not affect project plans
- **Communicating outside of our weekly Zoom meetings**
  - Overcame this by texting in our iMessage group chat, but was still hard to gauge where everyone was at in their project roles
  - Did not affect project plans
- **Deploying to Heroku (**ran out of time and some unresolved errors)

Demo + Q&A