Weather Men

Ian Fernandez, Noah Vilas, Will Dravenstott, Matthew No, Gustavo Sanchez-Sanchez, Jay Bentley

What is WeatherMen?

- WeatherMen is the future of weather forecasting and presentation to the public!
- This is an application which displays weather information about specified cities. The user is able to favorite cities along with searching for others around the world!
- However, this is no ordinary weather application, here we offer benefits to users such as clothing and travel recommendations. This is obviously determined by the weather and is all up to user choice!

Project Tracker (Github Project Board) - 4.5/5

The Project Board allowed our group to efficiently plan and organize tasks. It also allowed us to designate tasks for each member to work on independently and cooperatively.

VCS Repository (Github) - 5/5

This allowed all group members to work on their own set of code without interference. It also provided planning tools with a project board. The branching allowed members to seamlessly merge code into one master branch with all working code.

Database (PostgreSQL) - 5/5

This allowed for easier storage and access to information. It allowed us to keep a memory of users and cities so that users could personalize their profile that would be present when they logged in next.

IDE (VSCode) - 5/5

VSCode works seamlessly with Github allowing us to see which branch we were currently working in and to deal with merge conflicts within the software. Also, it is just simple and works well for writing and running code.

UI Tools (HTML, EJS, JS, CSS) - 4/5

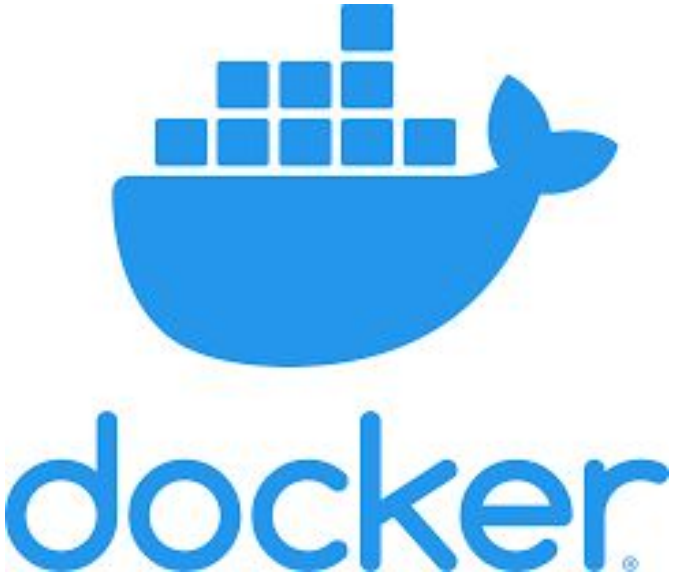
This was an effective way for us to create our website, however, we struggled with knowledge regarding api calls through ejs and other types of calls. CSS was difficult to learn but helped make our page a better experience for users.

Application Server (Node.JS) - 3.5/5

We struggled with our server through docker as some errors required special inputs. The Node syntax can be obtuse for first-timers but is powerful for building our site with database calls and rendering our pages.

Deployment Environment (CU Boulder/Localhost) - 1/5 and 5/5

The CU Boulder OIT server deployment did not work as expected during Lab 11 but we will attempt to use it to deploy. We will also give instructions for users to run the site locally using Docker and Localhost which work wonderfully for letting a user use and experience the application we built.



External API (WeatherAPI) - 5/5

Were able to get current forecasts and weather for any city around the world, with an easy to use api call and JSON response. We were able to search by city, postal code, latitude/ longitude, IP address and more.

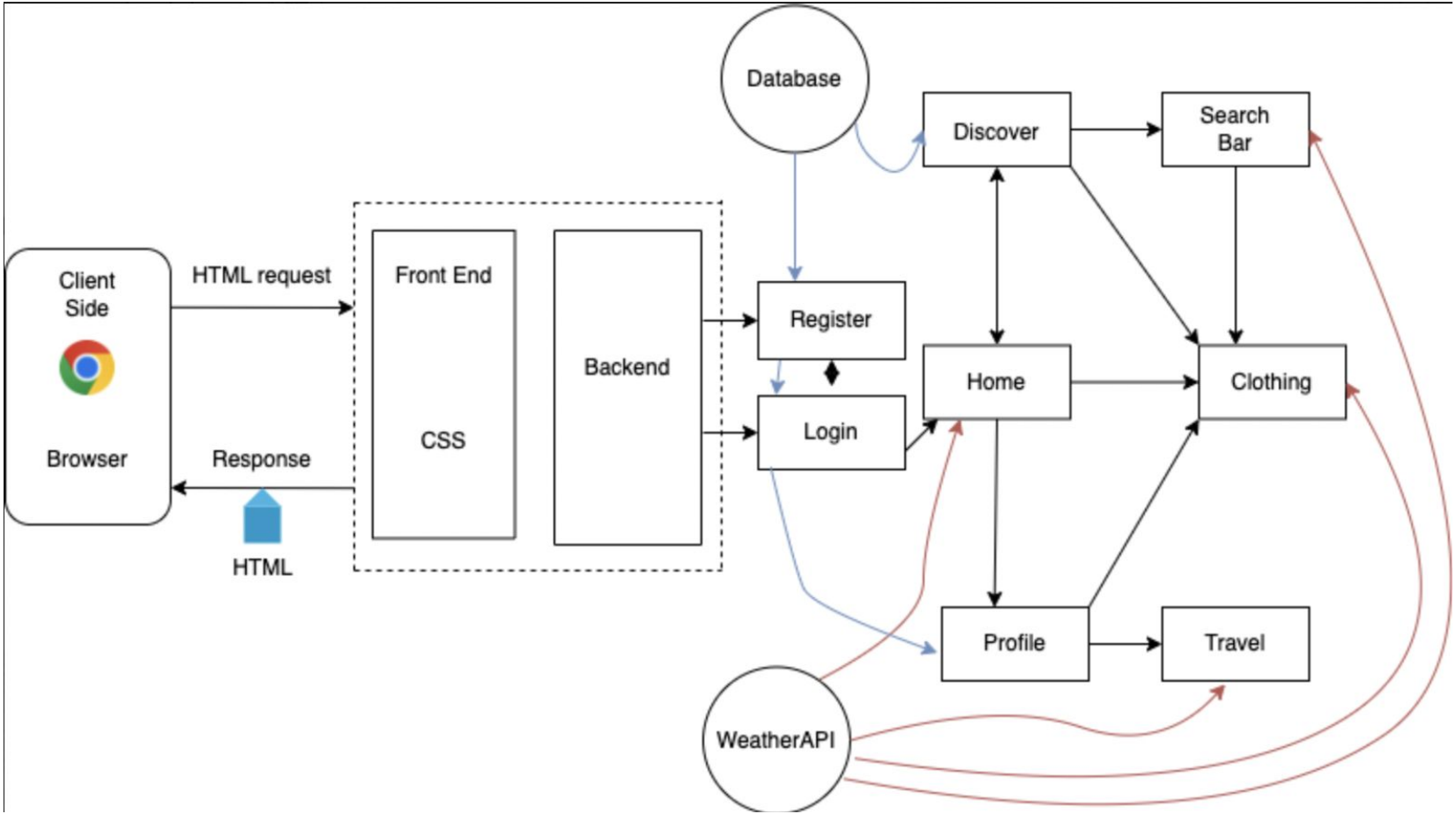
Auto Documenter (Github Releases) - 5/5

The Auto Documenter took all our group’s commits and auto compiled them into auto-generating release notes. It allowed us to designated a version release of the application and if it was a pre-release build. It was super simple to use and very descriptive in terms of features introduced and the group member’s contributions.

Methodologies

- Iterative
  - We used Iterative Development as it made it easier to start from scratch. We were able to throw code at the wall to see what stuck and kept trying things until we decided what we liked. We also took our prior code and built on top of it to continue progress.
- Agile
  - We used Agile to help plan our development progress. We met once a week with the TA in a scrum meeting and used 1 week sprints to track our current progress and plan future progress.
- Pair Programming
  - During group meetings, team members would work on a API call or a Node function together to help each other determine what was going wrong and what we liked for our application.
- Peer Code Reviews
  - We reviewed each other's code commits before merging with the main branch to help maintain the stability of the code base. This helped to minimize bugs we had to fix at in the final sprint.

Architecture Diagram



Challenges

- Getting the API calls for a unique search
  - This was very challenging throughout the project and took a week or two to complete, with the help of our team we soon overcame this challenge and completed the api calls.
- Staying organized
  - We used a project board on github to track what was in progress and what needed to be completed, this along with effective division of tasks helped us to complete the project and overcome the difficulties of staying organized.
- Effective communication
  - We used Discord to communicate. The group struggled to communicate with each other at the beginning of the project but we got into a better pattern of communication as it progressed.
- Time management
  - As students in many classes, each of us have many classes to juggle so each of us took turns taking on larger portions of work when we had more time to contribute
- Sickness struck many of us
  - We gave team members time to recover and made sure we adjusted the project schedules so that other members could continue to work without other aspects of the project working

Experience our Product!