Venture Vibes

Andrew Truong, Mae Chen, Emmy Wolf, Noah Rose, and Wenbo Zhang

Description

Empowering personalized travel experiences, our app simplifies vacation planning by crafting tailored itineraries that match individual preferences. We aspire to streamline the overwhelming array of options and create memorable, personalized journeys for every traveler, ensuring each person discovers the perfect activities that resonate with their unique personality and desires. Unlike TripAdvisor, VentureVibes suggest vacation destinations and activities during their alloted vacation period that users can add to their trip iteneraries.

Tools Used

GitHub Projects: 3 stars, useful for organization, but not something we found ourselves constantly referring to more than once a week.

GitHub Repository: 5 stars, mandatory to hold all of our code, documentation, etc...

PostgreSQL: 4 stars, necessary for user login information.

VSCode: 5 stars, efficient and clean coding environment that was readable by everyone

JS and EJS: 4 stars, flexible coding languages that allowed relatively painless full stack development

NodeJS: 4 stars, necessary for downloading packages and creating REST API endpoints, but linking APIs together did become daunting

Azure: 5 stars, simple and easy to use deployment platform that gave us little to no issues.

TripAdvisor and OpenWeatherMap APIs: 2 stars, while these were the best options for what we wanted to achieve, TripAdvisor has misleading documentation and OpenWeatherMap is very limited on a free trial.

Bootstrap: 5 stars, a very easy-to-use formatting and styling service that made our website look clean and professional

Mocha and Chai: 3 Stars, Good for testing but annoying

Docker: 5 stars, simple and easy to run locally

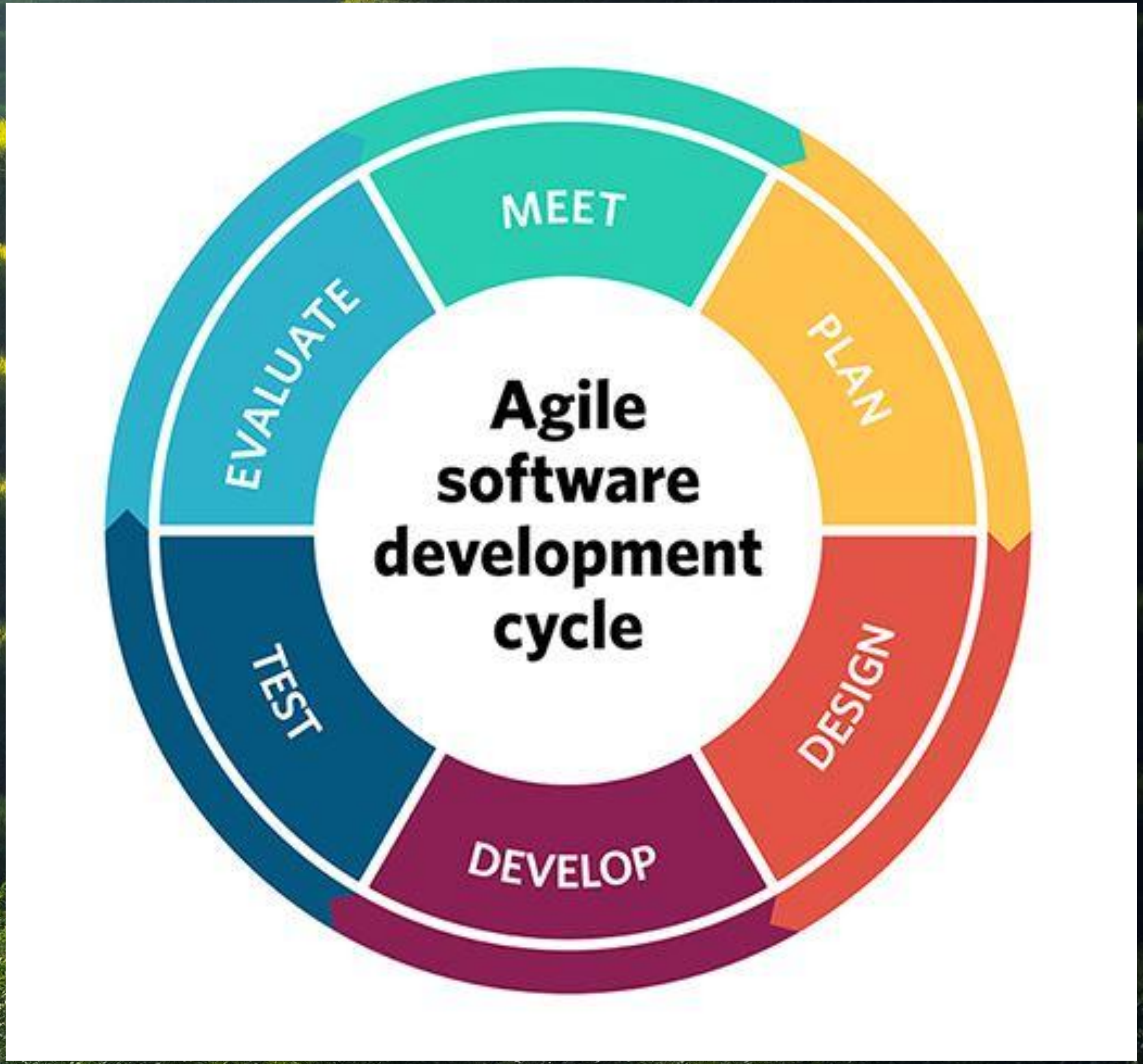
Methodologies – 5 Stars

We used the agile methodology of software development. We first met together and discussed what topic we wanted to pursue and how best to implement our idea.

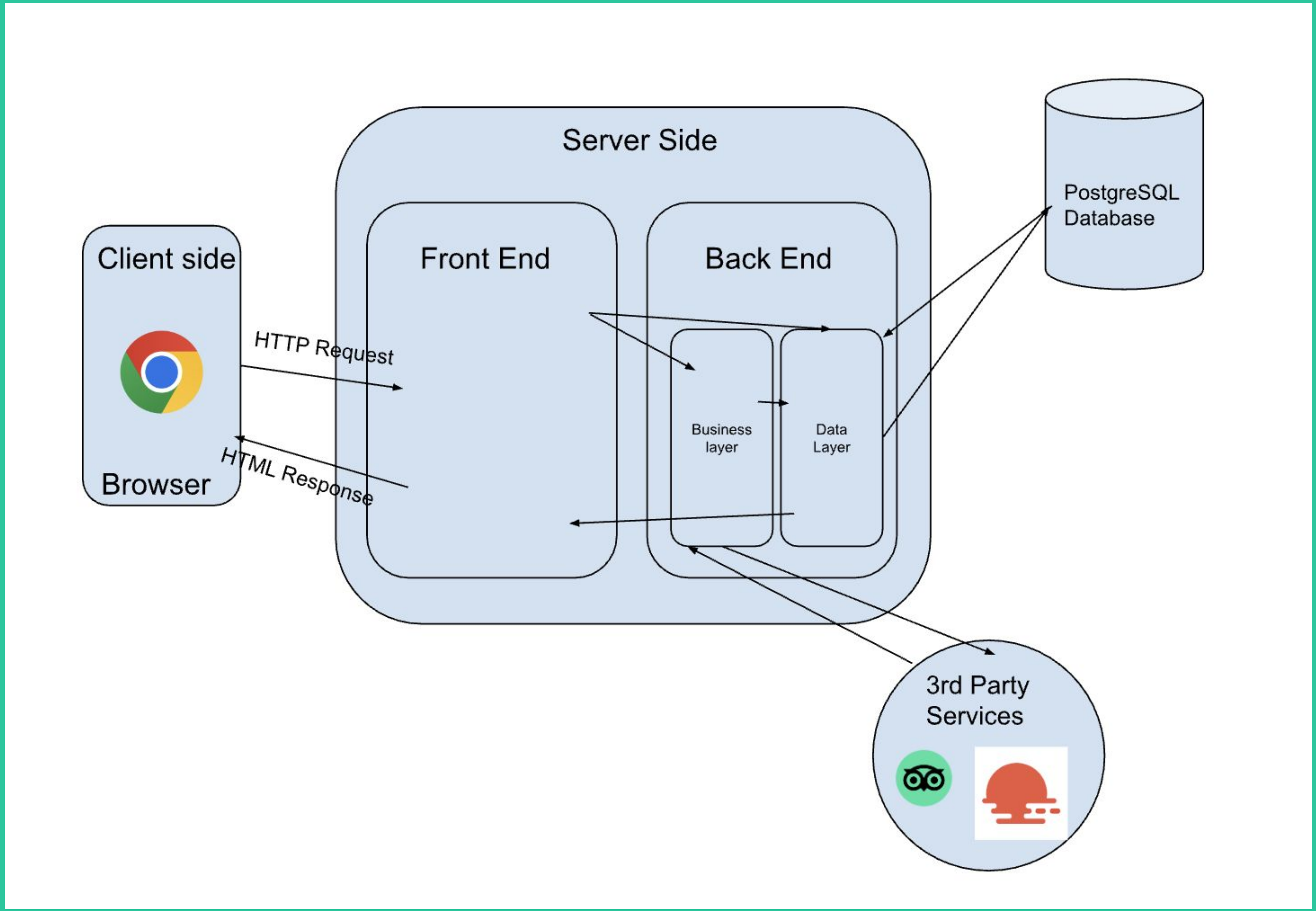
We then layed out all of the files that we would need and planned who would work on which functionality.

Whenever we made progress on our individual design/development, we would push our changes into the main branch and it would have to be approved to make sure there were no merge conflicts and that we were all on the same page.

Once it was constructed, we started testing before going into the details and then would evaluate what would need to be changed from there



Architecture Diagram



Challenges

Overall Challenges

- Version Control:

We imagined our code be modular that each of the task/issues can be solved separately, but turns out to be that the implementation of some of tasks are not distinctly separated. Since we all fetch the same repository at the same time and work on them simultaniously, occasionally we find that in order to make my codes run I need to make changes on other codes. And that ends up sometimes we covers each other’s work, either partially or completely to make the local implementation functional.

- Api Test limit:

During Testing, we noticed that we have not only a monthly allowance of usage, but also a daily limit, especially when we make multiple request per search, the count accumulates quickly.

This became a real problem when we trying to test how everything works together, we registered and attempted to use alternative API keys for tripadvisor when we run out of daily allowance during testing and this has caused more problems while doing so.

Overall Challenges

- Integration:
  Due to our largely diversified coding background, every merge request and integration attempt turns into a large Q&A session in chat, and the fact that we fetched everything at same time, the difference in implementation of each other’s code in the version control doesn’t always work together. This creates all sorts of funny issues that weren’t exist in each individual code, but appear when we put things together. The increased communication and quick response from everyone of the group SMS chat as we encounter these problems did help drastically.

Individual Challenges – Noah

- API Linking:

  Attempting to take the response of one API and sending it as the request to another proved to be a more challenging feat than anticipated. Working with async/await functions and making sure the data existed before calling the next API with axios became complex and overwhelming very quickly.

Individual Challenges – Wenbo

- Testing
  - Despite the difficulties with trial and errors, This is a good experience in understanding the causes of program misbehavior and learning different ways to approach such issues.

- Github
  - The thing I feel benefitted the most out of this project is the large amount of practical usage of github version control feature and learning how a team could utilize it more effectively. From the beginning that I struggle with updating my local files to I could quickly compare multiple branches and address potential issues.

Individual Challenges – Mae

- Delete/Add Activities:
  - I became very comfortable with SQL queries in javascript and EJS
  - I became more experienced with GET, POST, and UPDATE API routes
- Unit Testing:
  - Mocha and Chai has a time out of 2 seconds, since our GET discoverData API took a long time I learned to how to change the time out amount
  - Became better at understanding error codes

Individual Challenges – Emmy

- Dropdown:
  - I faced a major challenge in creating the dropdown options for the radius and weather forms in discover.ejs. None of the options would show up, so after multiple tries I ended up switching over to a <select> tag instead and disregarding the <dropdown> class all together.
- Logo:
  - I created a team logo for us to use but ran into trouble when trying to implement it into our project. This was because the image was not public and was not available as a link, and even when I tried to input the image directly into the views file, it would still not be routed correctly.
- Calendar:
  - I implemented a calendar feature to the start and end dates for the user to select instead of having to type it in. I ran into a lot of trouble with this feature and found that it would completely change the formatting and only work for either to start or end date. I got the stylesheets to all work correctly and the script tags in the right place along with the correct java functions to make them work.

Individual Challenges – Andrew

- Database setup
  - One of the things that came up repetitively during group discussion during early on of the project was the vision of the final product, and what kind of structures we need to set up on the back end. This back and forth has lead to alternative solutions multiple adjustments and to our user database.
- Github
  - I feel like I benefited from struggling with github when it came down to learning how the git commands work. This includes making my own branch, learning how to push my changes to the main branch, and how to merge my branch with the main branch.

Demo

http://recitation-11-team-02.eastus.cloudapp.azure.com:3000/login