UnlimiTrade

Spring 2025

Asher Siegel

Ethan Doyle

Yasir Qasemi

Max Lane

Connor Short

Ivan Gorodinski

UnlimiTrade can be described as a website that:

- Allows users to create accounts via username and password
- Allows for users to post items that other users are able to trade for
    - Example: user 1 has a wallet posted. User 2 requests a trade for the wallet with item that they have posted ushc as a book.
- No sort of currency is apart of any transaction, it is simply just trading one item for another
- Allows for users to give reviews on the user they traded with which adds a sense of security for future traders

Project Description

Git and GitHub

- Satisfied the version control portion for our project
- Used for tracking code changes, collaborating as a team, handling merge conflicts, and managing individual branches
- Github Project Board: Where we assign and track tasks
- 5/5 stars due to its user-friendliness and wide variety of features

Methodologies

- Peer Code Reviews: We would sometimes have multiple people assigned to a story and they would review each others code
- Iterative Development: We broke the main parts down into smaller parts to make it easier to tackle and progress our iterations

Tools Used

Expected Tools

Project Tracker
- We didn't specifically use a project tracker because we were able to hold ourselves and each other accountable for our work. We also utilized the GitHub story board which somewhat acted as a tracker 3/5

UI Tools:
- We heavily utilized handlebars for simplicity and CSS to enhance user experience. 5/5

Database:
- We used a PostgreSQL environment in VSCode locally, with no additional services. 4/5

Application Server:
- We used NodeJS asour application server and Docker to containerize the environment. 5/5

Expected Tools

Deployment environment:

- We used render to host and deploy our web application, allowing us to easily manage our server, database, and environment variables with automatic deployment from our GitHub repository

Framework:

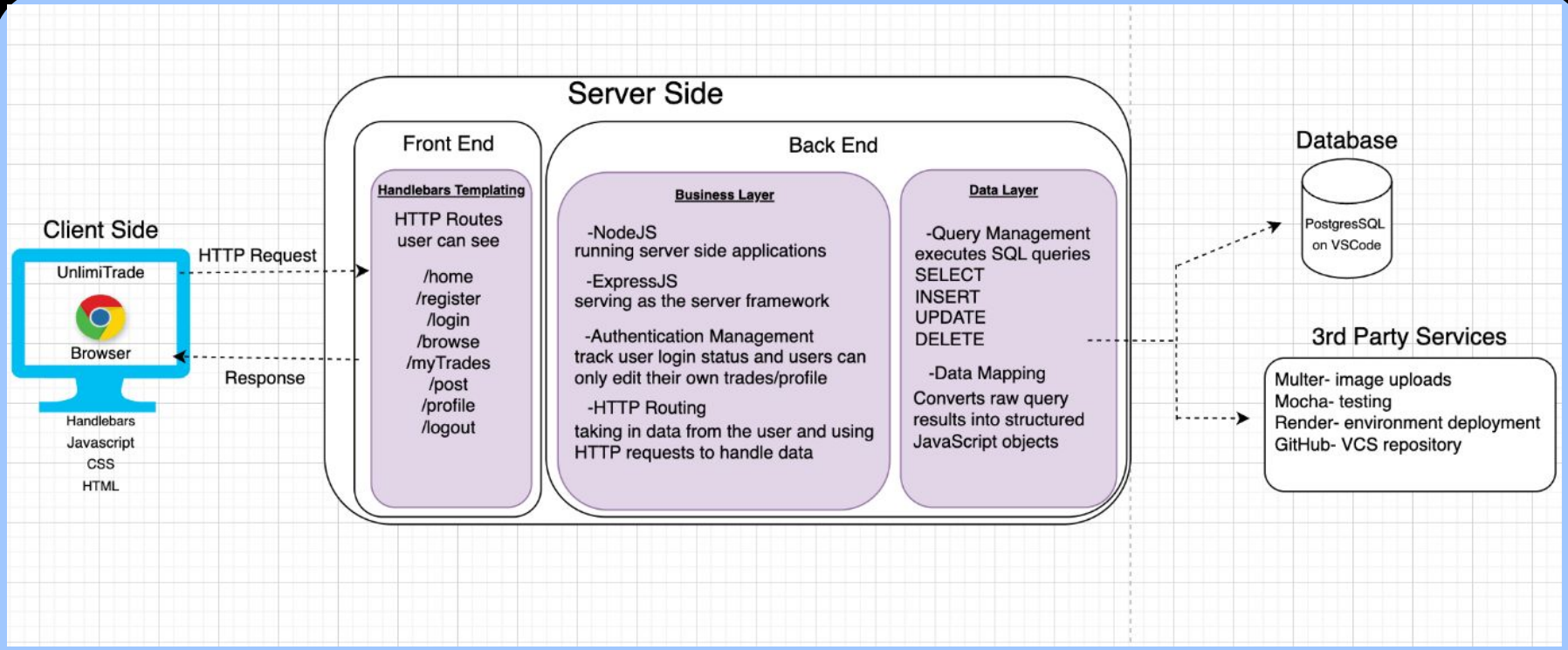
- We utilized expressJS to build our server-side logic and define routes for handling user requests

Testing Tools:

- We used Mocha to test our programs because it was very easy to use and efficient. Mocha was also very customizable and had great support for async/await tests using callbacks

IDE:

- We didn't use an IDE, we just used a code editor, VS Code for the entire project



Architecture Diagram

Challenges

Here are some of the challenges we faced and what we did to overcome them:

Actually starting the project:

- We had a few issues with getting all of the pages running since we used part of lab 7 as a ‘skeleton’. We jumped the gun and got too far before making sure we could get it up locally with docker

Being able to see other people’s items:

- Actually being able to see public items from other users was super difficult. We made sure to work through this with patience and a lot of testing with the cloud server

Time management:

- This doesn’t directly include coding, but includes the group side of the project. It was tough to find times for everyone to meet at the same times to practice and push for sprints, so we would sometimes split up the group to accommodate for everyone to participate

Making each user perspective unique:

- The trade history and my trades pages are different depending on if you requested the user or if you just accepted it

Future Enhancements

If we were to continue this project for a longer period of time, we would want to:

- Create a more complex searching process for items. This would include more filters and more sorting, to better enhance the user friendliness and experience
- Create a chatbot. This was talk from first first day when we were creating the storyboard on GitHub
- Dedicate more time for the user interface. We focused on getting the functionality of the application down rather than prioritizing the UI and CSS. We started with one CSS layout for each page at the beginning of the project and didn't touch it too much through the process of the development of the application
- Integrate google maps when a user makes a trade so you have the option to meet up in person to make a trade, instead of shipping it directly to the other user
- Improve compatibility across different screen sizes and devices
- Implement API such as supabase to handle the photo uploads

UnlimiTrade Demo