Rhystic-Storefront

Byron Liu, Diego Olin, Kanishka Jayathilake

Project Description

Users will be able to purchase Magic the Gathering cards at current market rate obtained from Scryfall API. Scryfall is a database of Magic the Gathering cards. It will also allow the users to make an account, view profile, and add and remove items from cart before purchase, as well as tracking their owned cards. Users can also list their owned cards up for trade, allowing easy selling of cards.

Target market

For magic players who are tired of above market rates and losing cards, the Rhystic Storefront is an ecommerce website that makes card buying and organizing simple. Unlike Card Kingdom and TCG Player, the Rhystic Storefront will provide market rate cards and organized card management and shipping, as well as the ability to track cards in your collection. Want to sell your old cards? No problem! The Rhystic Storefront also makes selling your cards a breeze.

Tools

Project Tracker - Github project board - 4/5

VCS repository - Github - 5/5

Database - PostgreSQL - 4/5

IDE - VS Code - 5/5

UI Tools - HandleBars, CSS - 4/5

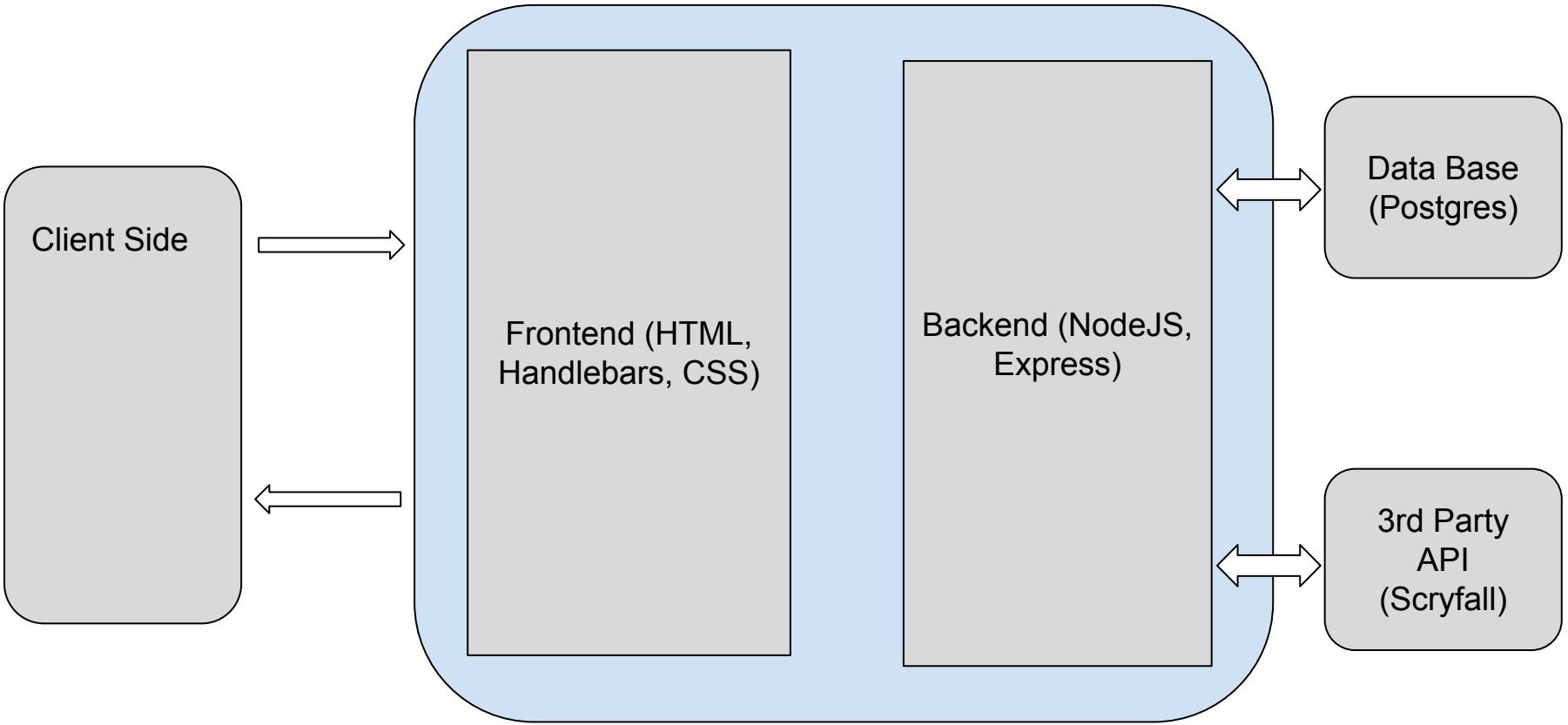
Application Server – NodeJS - 4/5

Deployment environment - Localhost and Render - 4/5

External APIs – Scryfall - 5/5

Testing tool - Chai and Mocha – 3.5/5

Architecture Diagram



Principal Sections of the Project

Byron Liu - Home page, Test cases, Shop page

Diego Olin – Database, Cart page, Inventory/profile Page

Kanishka Jayathilake - Login/signup page, Trade page

Home page,Test cases and Shop page layout

Description -

Home Page: Landing page where users can search directly for cards of navigate to another page via the navbar.
Shop Page: Users can search for cards, see their search results, sort the results, and add cards to their cart.
Test Cases: Testing to ensure that features work (mainly login/register)

Challenges -

Overall, the main challenge that I faced was getting test cases to work properly to be compatible with what had already been written for the website, for which I ran into a nuber of issues that took a substantial amount of time to solve. Outside of that, just managing my time posed some challenge, as the project had to be finished in a relatively tight time frame.

Database, Cart/Inventory pages

Description -

Database: The database stores the relevant info about each user and the cards that are in use. It also keeps track of items in the users cart, inventory, and items put up for trade.
Cart/Inventory Pages: Each page shows the user a list of cards in their cart/inventory and the cards relevant information. On each page is a profile element showing the user their name,username,email, and amount of money each person has. They are able to edit this.

Challenges –

I didn't have too many challenges in regards to the code itself, the main challenges i faced were in regards to time. Few big issues came up with work that limited the time i had available to work on this, leading to a few ideas that either weren't able to be implemented in time or i wasn't able to fully debug in time so were scrapped

Login/Signup page and Trade page

Description :

Login page -The Login/Signup page allows users to securely access their accounts or create a new one using their email and password When you sign up using email and password, those informations are hash with bcrypt and save in database.

Trade page-The trade page allows you to sell your purchased card to other users of the website. It also shows you cards being sold by other users. This page allows you to set your own price for the cards when you sell them.

Challenges :

The main challenge I had was network problems on my computer. Sometimes web ports didn't work properly, and sometimes databases didn't connect and exited as soon as I started Docker. Thankfully, with the help of my teammate, I built parts and checked them on their computer. Other than that, the limited time was a challenge.

Technical Details and Management

- We used a GitHub project board to keep updated and track each individual part of the project. We also used a Discord group to communicate with group members about the project.

- Our programming technique involved dividing certain tasks among team members, with each member responsible for building their part. After a member finished their part or encountered problems, other members would review it. After debugging and review by another member, we merged that branch with the main branch.

- We had weekly meetings every Wednesday after classes where we talked about the current status of the project, the next steps, and the problems we faced during the week. It was really useful for all of us to complete the project successfully.

Future Scope/Enhancements

- Introduce Customized Visual Designs
- Add E-commerce payment systems

Demo Time