WebMart

Team Incognito

Team Members: Connie Pak, Michael He, Ellen Yim,
Chan Young Jeong, Harrison Cooper, Nate Natividad

WebMart is an online retail service where users can sell or buy items that their friends have posted. WebMart uniquely has an interactive social aspect, where users can stay up-to-date with their friend's purchases and share their own status.

Final Codebase: https://github.com/ellenyim8/WebMart
Description of source code:
Our source code is composed mainly of HTML (done in Handlebars templating engine), Javascript, and CSS. Our backend, consisting of all of app.js and its imported modules, utilized NodeJs, Express, and MongoDB. We utilized MongoDB models (inside of models folder) to specify the type of data we store, as well as “handler” functions for routing specific pages (inside of handlers folder). The views folder consists of the handlebars/HTML files that contain the actual web page we user/generate for each url. The public folder consists of public assets, mostly images and a css stylesheet. For testing, we have __tests__ folder which contains Jest testing files and the nodejs-selenium-sample folder that has Selenium tests.

How to run our project

- Downlaod project off github, also install NodeJS (https://nodejs.org/en/download)
- Ensure that Key.env is downloaded from github with the following credentials inside:
  - DB_USERNAME=test1
  - DB_PASSWORD=test1
- In terminal/command prompt
  1. npm install
  2. npm start
  3. Click on link or go to localhost to open webpage
- You can register a new account or use any existing accounts
  - Useable existing account: email: mhe034, pw: 123

Automated Tests

Our testing is divided into 2 categories: front-end and back-end.

- For the front-end we use Selenium with a chrome webdriver. This generates a browser which automatically simulates a user performing tasks on the website. The selenium tests can be run with the following commands in the terminal:
  1. cd nodejs-selenium-sample
  2. npm install selenium-webdriver
  3. npm start
  4. node test.js (note: may need to split the terminal first)

- For the back-end we use automated Jest tests. These tests will be run every time a change is made to the code and saved (using vscode jest extension), or with the following command:
    1. npm test
- The tests can be found in the github repository or here (also includes description of them)

Implemented Features and Cuts

We have implemented the major functionalities and features for our project, such as login/registration page, exploring all items that are selling, user to buy/sell an item, requesting/accepting friend requests, editing user profile, and listing items that are being sold.

We did add some additional features such as the payment page, and added a bit more features to smooth out the ones previously mentioned. For example, for listing items we made it so that the items being sold by friends are listed on the home/exploring page, and when a user goes to their profile they can see what items they are buying. If they go to the item listing page, they will be able to see what items they are selling and what items have already been sold by them. The core functionality is the same in ‘listing an item”, we just put more detail into it.

We had to take out the auctioning part, with bidding and putting initial bid prices and time limits on the items that we mentioned in our initial proposal. We focused more on the website being for retail purposes in general and emphasizing the social media aspect of it with the friends functionalities. We cut the auction part because it was more time consuming and it was a bit more difficult to plan out compared to what we ended up doing. This essentially saved us a few weeks of work that would've been needed to make this function successful.

Improvement

Our team used a Kanban board to make a list of all possible tasks that needed to be done, and each member split them up. Sometimes we would work in pairs on a certain task, and sometimes we would work individually. Even when one member took up a task, if there was any issue other members would join in to try and help.

In general though, the distribution of tasks were as follows:
Michael: Setup Backend, Handler Functions, User Schema/Class, MongoDB, Login Page, Landing Page, Home Page, Home Page Item Dashboard, Backend Requests, Item Page, Payment page, Explore Page, General UI, Jest testing

Harrison: Item Module, Tags, Testing,

Connie: List Items Page, Items Schema Model, UI for website (Landing Page, Friend's List, Home Page)

Ellen: List Items Page, Items Schema Model, UI for website (Landing Page, Friend's List, Home Page)

Chan Young: Registration Page, Create Items, Friend Requests/friend list, List Items/Items database, Edit profile

Nate: Profile Page, Explore Items Page, Edit Profile, Friend's List Page

The parts that ended up taking a majority of our times and workload were:
Michael: The majority of my time was spent setting up backend and creating/editing handlers (encapsulated class that handles get requests to got specific pages, done to apply single responsibility principal and organize what resources need to be sent to what page). This was because of the amount of MongoDB data each page needed and specific queries to do so. This was my original expectation. However I feel that I should have spent more time working on better schema design (which made some queries difficult) and incorporating more jest testing cases.

Harrison: I spent most of my time in the beginning of the project just adding the item module and improving page UI, which took longer than it should have because this was my first time with web development. Afterwards, I shifted my focus to testing and implemented the Selenium tests from scratch. I expected to contribute more to the development side but automatic testing was something we really needed at that point so it became my job.

Connie: I spent a majority of my time working on UI design for the pages, especially landing page as that was the first one I worked on. I had never done web design before so a lot of my time went into researching bootstrap and how design can be done on the website, but once I got used to it it got better. I also spent a lot of time working on how to get the items to show up on a list on one of the pages, as it was difficult trying to access the items that were created and figuring out how to display them.

Ellen: I spent most of the time working on UI for the pages, and I tried getting our item database to work. The item part took a long time for me because I was also learning JavaScript as well as working with MongoDB for the first time. Working on design for the landing page and some of the other pages that I collaborated with Connie on them. I expected to contribute more on the backend but since the first few weeks I took up was learning how to use the tools for our project. Overall I gained a good understanding on how to use Bootstrap which is much easier than building our own stylesheet.

Chan Young: Registration, Create Items, Friend Request, Edit profile

I tried to implement most of features that functions need. For example, registration error checking when user try to use exists e-mail or username. Also, user can upload items or edit profile with local image. I spent most of my time to understand and use nodejs, express and mongoDB such as form,post,get and multer. I expected to spend more time on refactoring the codes but I was pressed for time to implement function without errors.

Nate: I spent a lot of my time working on the front end for the profile page and the edit profile page and also some work to get the functionality for the edit profile. For the backend, I ran into some trouble when it came to getting the functionality for edit profile as it was my first time working with nodejs, express, and mongoDB so a lot of my time was spent trying to understand how to use them and I ended up having to ask for assistance with the backend of edit profile page from Chan. For the front end, I ran into the same issues of just not having experience with html and css that I was spending most of my time reading and trying to understand exactly what I was doing. However, I was able to get a good enough understanding to create the simple layout for the profile page. I was expecting to contribute more to the backend of our project, however I spent most of this quarter learning and understanding so most of my contribution came from design of our app such as setting up the profile page, friends page, and the edit profile page and the documents that were required to submit throughout the quarter.

We spent too little time on UI in the end, it could have been better than what we ended up with if we had more time and more knowledge on the front end tools to improve it. We also ended up spending a lot of time on the list items portion and explore items portion as it was difficult to figure out how to access the data from our database and have it appear on the website. This took a while to work on, but other than that there wasn't anything specific that we spent too much time on. We generally finished each task within a week of assignment and it was ready to be presented during our group meetings. If not, that's when more group members joined in to help with the task.

Feedback to Staff

Feedback to Instructor:

Lecture slides should provide more examples on something similar to what is covered in quiz, specifically quiz 4-6. Deadlines should be more clear, as there was a lot of confusion with this in the beginning few weeks where no one knew what was due when. Also, when some of our reports were assigned there was no deadline given so that was confusing.

Feedback to TA:

During lab sessions, TA was too quiet when showing testing tools and also the demo was very short, as it lasted less than 5 minutes. In the first few labs, the TA also just sat there so none of us knew what we were supposed to be doing, so we waited for around 20 minutes before starting work on our own.