README for shop.it

shop.it was going to originally be called “craigslist clone”, but I figured that was too boring and would give users the “oh he just copied it poorly” thought. I then chose to call it shop.it because it is something unique and it had a nice sound to it. I built it without following a guide (you will probably be able to tell if you look at my code, not saying it’s bad, I’m just sure a more experienced dev will be able to tell). I utilized Figma, which is an excellent UI design software, and a free logo maker for the logo to give this shop.it idea an actual look. After all, you can’t start making something if you haven’t made a prototype or sketch of it to follow along to.

I used REACT for the front end and mainly used the useState and useEffect hooks. There was plenty of JSX but that should go without saying.I used context API to store the logged in users and the respective user’s cart contents. I created basic components and a basic website layout along with various “pages”. The navigation was handled by react router and I used a bit of state passing when navigating to other pages”. I did not use a CSS framework. My idea was to practice my CSS skills a bit.

The backend was done with express and it communicates with the front end via API endpoints. This server was a basic-ish one that I managed to put together using my ToDo-MERN (can be found on my github) backend as a boilerplate. It communicates with a MongoDB database and 2 separate collections within that database, items and users. Items is where the items that are for sale are kept. Each item has various keys including a key for the URL to the respective item image. The URL links to the image file in an AWS S3 bucket I set up to hold the actual images of the items for sale. I used the AWS-SDK package from NPM to allow uploading images directly from the app, allowing users to add items to the store intuitively, like they would on… well… craigslist. The users collection is where the information about registered users is held. Each user has their own record that contains information about them and an array of objects that contain the itemIDs to the items in their cart and the quantity of said item. When a user logs in, the client sends the inputted email and password to the server (in the body of a POST request) which then does a query to MongoDB to find a record that has that matching email address. If it finds it, it returns the user’s info and then compares the inputted password to the password for that user in the record grabbed from MongoDB. If the password is correct, the user info is set as the state of the UserContext, which then goes on to set the CartContext with the contents of the cart array in the users record. While this isn’t the most secure method of authenticating, it is a method I developed and managed to figure out by myself. After all, I wanted to challenge myself and showcase what I can do without following youtube tutorials or guides. In all honesty, the majority of stackoverflow surfing I did was for CSS rules.

My database used for this app, if you couldn’t tell by now, was MongoDB. For image storage, I used AWS S3. MongoDB would hold the URL to the publicly accessible image stored in AWS S3, and getting that URL ended up being a matter of following a certain structural convention, which could be done from within the server itself. MongoDB stored everything else.

Final words:

I read a book “Think and Grow Rich” and the one thing among many it kept bringing to my attention was that rich people (in health, wealth, relationships, knowledge, etc.) had a habit of being dedicated and persistent. Persistency and dedication must be how one essentially “levels-up” right?

I started my dev journey (for the 3rd time) about a year ago, with the difference between the other two attempts being me being persistent and dedicated to learning. No matter how much I didn’t feel like it, I did it. Even if it was a little bit. Fast forward to today, I am truly amazed that I’m at a point where I can describe in detail a completed Full Stack project. That I created. I still get a kick out of reading this document because of how I *understand* what I did and am reading. I feel confident talking about it. Among the many reasons this project was undertaken, the most significant one must be to prove to myself that persistence, dedication, and doing a bit everyday even if it is a little bit, can achieve a lot.

Shop.it may not look like much to a more experienced dev, but I am very proud of myself and how far I have come in building it.

Thank you for reading.