SWDV 691 Capstone

Professor Gradecki

MVP Assignment

October 24th 2021

MVP Assignment

Establish what the project is:

The project will be an E-commerce Website where I can sell products, I want to get rid of. The concept of the website is essentially a virtual thrift store. The site will be a place where I can sell any number of products. It will provide the admin with analytics to view the percentage change or revenue the total revenue and how many products were sold. Users can add items to cart, log out or check out.

What problem(s) will your web application solve?:

Today in America the percentage of income spent on non-essential items is at an all-time high. In fact, according to research conducted by Ladder. Americans spend $1,497 a month just on non-essential items. This totals up to $18,000 a year just on material items. The application will allow the user to sell products they may want to get rid of. According to the New York Times Magazine, 1 out of every 10 Americans rents offsite storage, which is the fastest growing sector of commercial real estate Additionally the U.S Department of Energy states that 25% of Americans have 2 car garages but do not have any room to park inside of them, while 32% percent of people can only fit one car into them. According to NPR, the average size of an American home has tripled in size over the last 50 years. You add all of these statistics together and you realize people have more space and they have filled up that space with items that take up space but provide no value in return. We have an excess of things; I am sure anyone who reads this paragraph might be looking around their desk and having the same feeling. What better way to get rid of this feeling than to sell off items that someone no longer uses? My website will see items to accomplish just that.

Who are the user personas that would derive value from your web application?

My application would do two things, one it would help a user sell thrift products, items in the home to clear up space and derive monetary value from that item in return. Secondly, it would connect another buyer to a product at an exceptionally discounted rate. This gives the buyer a product they might have been interested in for a cheaper price. The platform delivers in non-zero transactions where both sides can leave happy.

What value or benefits are derived from solving this problem? (i.e., there is a cost to building your application, do the benefits justify the costs?)

The costs of the application currently are hosting the website, plus marketing costs to promote the website and increase market notoriety. There is also opportunity costs associated with the product as there is with any product, the potential to be working on something more lucrative, however I would like to sell items out of my home as I am moving, and this would significantly reduce the costs to me. Also to users who could potentially use the website to sell items, this could allow them to reorganize their homes and give relief.

How would your anticipated users solve their problem with your web application?

Users would be able to sell products they do not need in their homes anymore. This frees up valuable space that could be given to much more important activities or use the money to spend time with family and friends.

How do you imagine they would interact with your application?

The result would be a binary outcome where users are selling products or users are buying products.

MVP SECTION:

What does the user interface need to support? / What minimal set of features will provide a workable solution to your users?

The user-interface needs to have the following components:

-A cart that can add or remove items from the cart.

-The admin account needs login authentication.

-The admin needs to see data revolving around the purchases on the site. Percentage change of increase or decrease for monthly totals. Monthly revenue totals and what products are selling.

-Sign in or login for users

-Clear pictures of products.

-Interface that is easy to use, easy to find pages etc.

Admin user flow:

Admin signs into their account. They can look at the analytics page, decipher what products are selling, what products need to be changed that have not sold. What products have been looked at. This will allow them to quickly gauge the business's status.

User flow:

Signs into their account, checks their cart if they left something there, the information is still there. Can look across the website for more product information. Can quickly remove or add items to cart. A user sees a product they like, adds to cart, checks out, payment information is given, a response is given back if payment was successful.

A *high-level* architecture picture (e.g., what is/are the service layer(s), what is/are the database(s), and what do all of them do in terms of your solution)

For my framework I will be using React JS.

The database that will be used will be MongoDB.

I will be creating a REST API where the information will be pulled from. I will be using JSON Web Token which validates the token on behalf of the REST API.

For payment authentication I will be using Stripe with node JS.

One of the key benefits of React is its ability to create single page applications. I will be creating pages that transfer objects through a built in function called "useContext" this will allow me to pass in data through anywhere in the application. I will also be implementing "useState" as well to check the state of variables I have created.

A high-level description of the data you will need to manage for the application to run (e.g., how will you organize your data into persistent objects, what are the major fields and relationships will they have

When the system registers new users. I will have the following table:

ID: Int primary key (unique)

Created at: Date Timestamp

Username: varchar

Email: string, foreign key(unique)

Admin: bool

Updated At: Date Timestamp

Password: string