Kristen Krofchik
Springboard Capstone 2 Project Proposal
Pirate Chicks Vintage E-Commerce Website

This project will be an e-commerce website for the small online vintage shop I run with my mom called Pirate Chicks Vintage. It will be a website where customers can browse our vintage inventory by a range of criteria (price, category, date added), add and remove items from their shopping cart, and purchase those items. A logged in user will also have the option of adding items to their "favorites" list or saving their cart items for later. The website will also include an "About Us" page, links to our social media profiles, a contact form, and a page where users can sign up for our email newsletter.

1) This project will be implemented with JavaScript using React for the front-end and Node for the back-end.

2) This project will be a full stack application.

3) This project will be an e-commerce website.

4) The goal of this project is to create an attractive, user-friendly and unique website where our vintage customers can easily find and purchase our products, as well as interact with my mom and I as owners via a newsletter and contact form. Mainly, I want this website to be easy to navigate and use, and to contribute to and elevate our shop's brand.

5) The users who visit our website will be vintage enthusiasts and collectors. Many of these users will be used to purchasing from larger e-commerce websites like eBay and Etsy, so it is important that our website is just as easy to find and use as those larger applications.

6) The data for this website will be pulled from our real world inventory of vintage and antique items. I will start the project with approximately 20-40 items, but the database will eventually be much larger than that. I will create the database of items myself, and it will include data such as: price, quantity, photo(s), estimated date/era, primary color(s), category, height, and width. I will also collect user data for users who choose to register with the website to save their cart and favorites lists.

7) Approach to creating this website:
    a) Some aspects of the database schema will be:
        i) USER

(1) id
(2) username
(3) Password
(4) email
(5) first_name
(6) last_name

ii) USER ADDRESS
(1) id
(2) user_id (FK references USER)
(3) address_line_1
(4) address_line_2
(5) city
(6) postal_code
(7) state
(8) country
(9) telephone

iii) USER PAYMENT
(1) id
(2) user_id (FK references USER)
(3) payment_type
(4) provider
(5) account_number
(6) expire_date

iv) PRODUCT
(1) id
(2) name
(3) description
(4) quantity
(5) primary_color
(6) era
(7) height
(8) width
(9) date_added
(10) price
(11) category_id (FK references CATEGORY)

v) CATEGORY
(1) id
(2) iame
(3) description

vi) CART_ITEM
(1) id
(2) session_id (FK references SESSION)
(3) product_id (FK references PRODUCT)
(4) quantity

vii) SHOPPING_SESSION
    (1) id
    (2) user_id (FK references USER)
    (3) total
viii) ORDER_ITEM
    (1) id
    (2) order_details_id (FK references ORDER_DETAILS)
    (3) product_id (FK references PRODUCT)
    (4) quantity
ix) ORDER_DETAILS
    (1) id
    (2) user_id (FK references USER)
    (3) total
    (4) payment_id (FK references ORDER_PAYMENT_DETAILS)
x) ORDER_PAYMENT_DETAILS
    (1) id
    (2) order_id (FK references ORDER_DETAILS)
    (3) amount
    (4) provider
    (5) status
    (6) date

b) I will be building the API for my website. I may run into issues organizing the data correctly and effectively, as well as setting up the proper communication between the data and the application.
c) I will need to secure a lot of sensitive information including user passwords and especially user financial data.
d) The website will include the typical functionality of an e-commerce site including: searching inventory by various criteria, the ability to add, remove, and save items to a shopping cart, the ability to favorite items, and the ability to purchase items. It will also include functionality for users to communicate with my mom and I as owners via a signup form and a newsletter.
e) The user will be able to browse the website without being logged on. The home page will be a landing page and the items for sale will be organized in a navbar on the left side of the page. The user will have the ability to register, log in, log out, or view their cart in a navbar on the top of the page. The shopping pages will include a list of React cards with limited information about the items. Once clicked, the card will take the user to a detail page for that item. The user can add an item to their cart either on the list page or the detail page. When a user is ready to checkout, the payment will be handled via a form accessed on the shopping cart page. After paying, they will receive a thank you screen with a link back to the homepage.