Chase Learn JS (231):
https://www.w3schools.com/js/default.asp
Used W3Schools to learn general syntax, good practices, and DOM control. Looking into possibly using strict mode or modules to increase security in our product. There is still a lot of information on javascript including specific API documentation and jQuery. Took the quiz provided and scored 21/25 only missing questions about browser and window API. Will continue to learn more as we progress, but now have a functional level of javascript.

Chase Host Docker (240):
https://www.serverwatch.com/virtualization/amazon-ecs-vs-ec2/
From previous experience, I knew that AWS provides an elastic container registry, however, I wanted to learn the different options. I decided to use the linked website to learn the difference, but it also told us how to host them either way. For what we needed, it appears that we need to start an EC2 instance, download docker, and start our image using our flask app.

Chase Learn Python (243):
https://www.w3schools.com/python/default.asp
Only needed to brush up on syntax, so this didn't take too long. We will be writing our flask app in python, so we needed to know how to start. Python is an expansive language that will depend on us learning the libraries as we use them. Scored a 22/25 on the provided quiz. Have functional use of python.

Chase Learn MySQL (245):
https://www.w3schools.com/mysql/default.asp
Though my focus is on the front end, to promote cross-training and reduce dependency within the group, I brushed up on a language I previously used. Utilized the above web reference, and notes from my previous database class to recall use cases. I have also completed the beginners SQL udemy class. Scored 25/25 on the provided quiz.

Chase Learn DB Normalization (249):
https://www.sqlshack.com/what-is-database-normalization-in-sql-server/
Brushed up on normalization techniques and data-key dependencies to be able to look at the ERD diagram and help the team decide if it is good.

Chase Explore API Options / Documentation (255/259):
https://developer.ebay.com/
Read through a bunch of options for APIs, but found that eBay had separate scopes for browsing and ordering, which made me feel more comfortable. Also, it was one of the companies that allowed for account activation with an education email. Finally, it has a lot of documentation and support.

Chase Learn GIT (266):
https://learngitbranching.js.org/
Played this git puzzle game.

Chase Docker Research (236):
https://docs.docker.com
Read through the docs. Don’t anticipate a need for CI/CD, so docker image files seem simple enough for a single container application.

Chase Learn DB Hosting (253):
Briefly explored options on AWS, but Nick did most of the setup and explained it to me. That's about as much as I need to know currently since the DB is already hosted and we can now modify it using MySQL.

Chase State Dive (235):
https://www.freecodecamp.org/news/state-in-javascript-explained-by-cooking-a-simple-meal-2baf10a787ee/
https://www.freecodecamp.org/news/how-to-manage-state-in-your-react-apps/
Looked into managing state using react. State at a JS level seems to just be variable representations of values that are stored. In a react level, it appears that state is a function that is destructured into a value and a function that will change that value. We can then use these functions and values in our components. More specific research will need to be done when looking into react redux.