Over the last month, I have been reading a lot about web development and it was extremely overwhelming initially because of the sheer amount of information on the topic and I didn't know how to begin.

But progressively, after taking a few online courses and reading books on the topic I feel a lot more confident about creating web applications.

User Manual:

This is a traditional todo app with CRUD functionality. Clone it onto local system and run it locally or view it on Heroku once its published.

Difficulties faced:

In the past, I have done some web applications with NodeJS using the express framework and have worked with NoSQL databases (Mongo DB). However, in this project I had to relearn a lot of web development techniques I was familiar with as I explore using other methods to do the frontend and backend. The biggest challenge I face was transitioning from the usual frontend development techniques (using HTML/JavaScript to manipulate the DOM directly or using HTML templates like ERB, EJS) to using React to handle the frontend application. The concept of React (lifecycle, states, virtual DOM) was rather foreign to me and even basic tasks like creating a post form was a challenge. In the past, I was able to create forms using HTML easily but with React, I had to also deal with the setState and onChange handlers was a little unfamiliar to me. Another challenge was the issue of resolving the CRSF (cross-site request forgery) token authentication. The guides that I have been using suggested using Rails form action helpers which automatically handles the CRSF authentication but I since I was using React, I had some trouble associating my database to the post route. I was finally able to resolve the issue by sending the CRSF token through my post form in React. There were also problems encountered with creating the React components initially. The concept of creating reusable components which can be rendered to any template was a little tricky for me. Most of the time, my components

simply didn't work or render as I had intended, and I had to look up on viable solutions on Stack overflow and tutorials.

In the backend, many of the concepts were less foreign to me like RESTful routing, database handling (associations, models, migration), packaging components into partials and modules. Other than learning SQL databases, which was a little intimidating initially, it was manageable.

<u>What I have done so far:</u>

I have set up a database and migrations (SQLITE 3) and finished with the basic CRUD of the todo app. I have done some of my styling components like navbar and buttons, but most of the app is not styled yet. I plan to do the styling once I finish my user association and authentication. I will be deploying my app to Heroku soon once I clean up some of the code. I have also gone ahead and done some error handling and added flash messaging for user friendliness. I plan to add more features to allow user-user interactions like messaging and sharing of todos for the final submission.