

CVWO 2020 Mid-Assignment Submission Write-up

Tan Ying Jie, Dexter

A0149802A

Introduction & Learning Process

Prior to attempting this assignment, I had little relevant experience in web development and object-oriented programming. As such, I first familiarized myself with Ruby through Codecademy before going on to the tutorials provided by the Ruby on Rails Guides and React.

First Attempt

https://github.com/dextertanyj/ruby_todo_app

After completing the recommended tutorials on Ruby on Rails and React, I started development on what would be my first of two attempts at making the todo application. I started out by drafting a brief overview of what features I would like to have included in my application and decided on the following:

1. User accounts and authentication
2. Basic CRUD functions on todos
3. Each todo should have a title (required), notes, due date and category.
4. Basic CRUD functions on categories
5. Filter by categories.

Since it was my first time developing a web application from scratch, I decided to break the development into 2 sections, backend and frontend and attempt to tackle them separately. I began my research into sessions and user authentication and decided to use the bcrypt gem to implement this function as it assisted in storing salted passwords.

When designing the database for the application, I introduced 3 models: Users, Categories, and Tasks (before I decided to use 'Todos' instead in my second attempt). Both Categories and Tasks had a many to one relationship with Users as each category and todo had to belong to a user. However, I did not introduce a relationship between Tasks and Categories as I wanted to allow the user to leave a todo as uncategorised. (After realizing that a foreign key could be null, I would have done otherwise.)

This implementation would mean that any delete or edit action on a category would need to cascade down to its associated todos manually and thus I had to implement them in the category controller. Additionally, validation of categories had to be done at the form level and thus I employed the use of the rails select helper for displaying category options based on records in the category model.

Throughout this time, I had been using the Rails view helper to test and debug my application without using React yet. As the backend of the application became closer to completion, I changed my focus onto React development and thus met my first major setback. My lack of experience in JavaScript and unfamiliarity with React proved a significant hurdle in trying to integrate the current state of my application with React. After spending a significant amount of time trying to resolve the issues of user authentication and submission of forms, I decided it was better to restart from the ground up with new knowledge gained from this experience.

Second Attempt

In my second attempt at creating the web application, I decided to use the JSON API method. This method proved to be significantly different as routing was done on the client rather than the server and thus required me to read up on reach/react router as well as HTTP request methods. Currently, I have not found a secure method to implement user profiles and authentication and have thus left it out in the current application. A possible candidate is JSONAPI::Authorization, although it has not yet been adapted to JSONAPI::Resources v0.10.

The Road Ahead

In the time leading up to the final submission, I plan to complete the following:

1. Improve UI of the application.
2. Implement due dates to todos.
3. Sorting and search functionality.
4. Debugging.
5. Refactor the application to improve code reusability.
6. Attempt to implement a form of user accounts.
7. Deploy the application to Heroku.