CVWO 2020 Assignment
Final Submission Write-up

Name: Dorcas Tabitha Tan
Matriculation Number: A0190356A

This submission contains two sections: first, my reflections on what I have accomplished, and second, a brief user guide. Other documentation and implementation notes are available in the README file on GitHub.

Reflections

Overall, I am fairly proud of what I have learnt and accomplished through the process of creating this basic web application, which is the first publicly-available application I have written. Designing and implementing each feature from beginning to end helped me to see the real-world application of principles like usability, maintainability, efficiency, and security. Some highlights of my work are: building and consuming a JSON API, allowing for flexible tag management (no need to manually create or delete tags), automatically associating a new to-do with its creator, and making use of sessions to keep users logged in.

Firstly, I am pleased that I managed to find some best practices and implement them when adding features or solving issues, particularly in the early stages of the project. Having flexibility in implementation meant making decisions at every step of the process. Thus, I tried to be forward-looking when choosing one design over another, though the application is small in scale. For instance, I avoided using custom methods in Rails controllers to make sure the code remains easily maintainable, although I did make an exception (to create tags when editing to-dos without fetching from the API multiple times). I also tried to ensure that logic was implemented in the relevant layer.

Secondly, I was able to achieve my goal of gaining a better understanding of the web development stack. With no prior experience in web development, it was eye-opening to see more of the different processes happening behind each action on both client and server. (I only realised this difference later though, so the client code is still in the "server" folder `/app/javascript/components`.) I have learnt much about different steps of the web development process, including designing database schema, adding model validations, writing controllers for both Rails and JSONAPI::Resources, passing data between application layers, managing (the illusion of) state, and designing an intuitive user interface. I believe that this understanding will serve me well in any future web dev work even if the specific libraries or frameworks differ.

Lastly, I am thankful for the chance to develop my skills in an independent way and experience programming "in the wild". Searching for solutions online was not always straightforward, especially when encountering unexpected issues, and I learnt resourcefulness and resilience through the process.

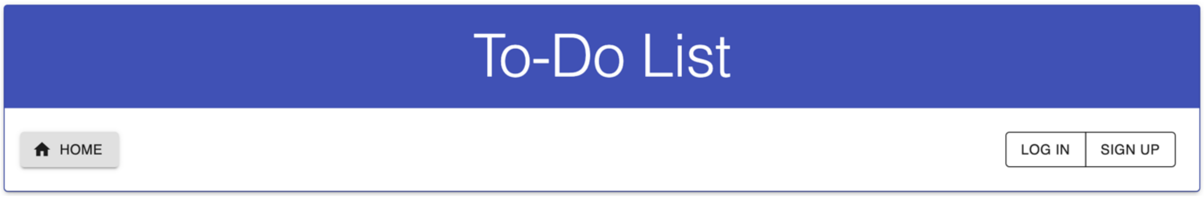
That being said, there is much that I would have liked to improve, especially with regard to user accounts and security[1]. While rushing to complete the functionality in time, I was not able to fully ensure that user data remained secure and to minimise information leakage between accounts. In addition, I think there is room to further utilise the capabilities of the various libraries rather than writing (potentially error-prone) methods. Finally, though creating a clean and intuitive user interface was satisfying, the final code could benefit from some refactoring.

User Guide

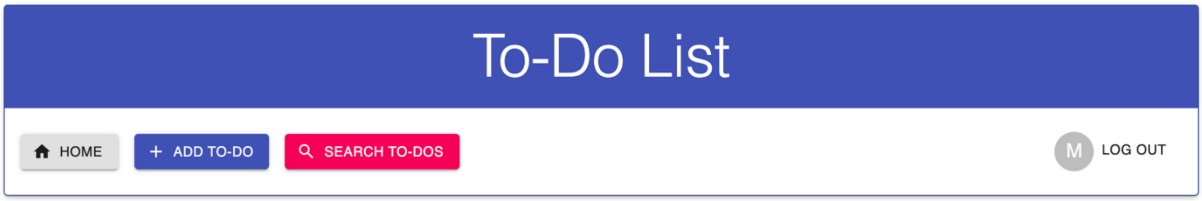
Navigating the app

The *navigation bar* is located at the top of the app. It contains useful links based on whether you are logged in or not.

Not logged in:



Logged in:



Getting started

New users: **Sign Up** for a new account to access the functionality of *To-Do List*. All you have to provide is a **username** and **password**.

- Do choose a unique username. It must not be the same as anyone else's.
- Try not to use the same password that you use for other accounts – this application is not entirely secure.
    - **IMPORTANT**: Please remember your password. It cannot be retrieved.

[1] I had originally intended to convert the front-end code to TypeScript, but decided to implement user accounts instead as I realised that a lack of separate to-do lists for different users would badly affect the application's usefulness.

Existing users: Head over to the **Log In** page to access your account.

NOTE: You *can* access the log in and sign up pages if you are already logged in, but a "Log Out" button will be displayed in the top right corner.

Managing tasks

Viewing To-Dos:

- Your list of tasks is available on the **Home** page after log in.
- Alternatively, you may choose to view them from the **Search** page by selecting "Show All".

Adding a To-Do:

- On the **Add** page, you will see a form where you can specify a title, details, and tag for your task. You *must* include a title and tag for your to-do.
  - By default, the tag is 'General'. You can change it to any value according to your needs, but the tag name must be alphanumeric.
  - You can also use the details field if you need to remember other details about your task. This is optional.
- Click the "Add" button to create the to-do. The button will only be available after the required conditions are met.

Searching for To-Dos:

- On the **Search** page, type in a keyword or word fragment and specify whether you want to search to-dos by title, details, or tag. You can type part of a word (partial matches work). Also, searches are case-insensitive so you don't have to worry about capitalization.
- Click "Show All" to view all of your to-dos in a single glance.

Modifying To-Dos:

- You can modify a to-do by selecting the relevant button(s) in the to-do list. To-dos are modified directly from the page that you are on (**Home** or **Search**), so you can edit more than one to-do at the same time.
- To *edit* a to-do, select the "Edit" button.
  - You will see a form where you can modify the title, details and/or tag of the to-do. Both title and tag must be present, and tags should only contain alphanumeric characters.
- To *delete* a to-do, select the "Delete" button.
  - **IMPORTANT**: Be careful when doing this. Deleted to-dos cannot be retrieved.

End of document. Thank you for reading!