

CVWO Final Submission

Name: Tan Yi Guan

Matriculation Number: A0217680A

GitHub Repo: <https://github.com/tenebrius1/CVWO-assignment>

Live Website: todo-list-yg.herokuapp.com

Accomplishments

Although I have worked on similar "FullStack" Web development before, I think that one of the biggest accomplishment for this assignment was using 2 frameworks I had close to zero knowledge about (namely Ruby on rails and React.JS). While it was definitely difficult to learn new frameworks and develop a MVP for submission in such a short period of time, I think that it was a very rewarding experience for me to learn more about the concepts of rails and React which I have been wanting to learn for a long time.

Some of the things I accomplished includes:

1. Fulfilling the requirements of the Assignment up to Level 4 (Deploying the app).
2. Acheived a decent code review score of **B** for maintainability of the project on CodeCLimate!
3. Porting over from JavaScript to TypeScript for React.JS components for static typing. (Kind of regretted not using TypeScript from the start as the switching process took quite awhile...)
4. Having a decent understanding of how states and prop changes works in React to create a more seamless UX by not reloading the page every time the user changes something.
5. Created an authentication system using Rail's built-in `bcrypt` library.

Overall, I think that I have learnt a lot within the considerably short time frame I had for this project and it was an insightful dive into concepts such CRUD and RESTful APIs. I thoroughly enjoyed developing this project and hope to be able to learn and interact with like minded individuals through the CVWO program!

Short User Manual

Logging in and Registering new account:

1. When a user first visits the website, they will be greeted with the Sign In page where they are prompted to Log In with their user name and password.
2. If the user does not have an account yet, he can create a new account by clicking on the "Sign up" link on the top right of the NavBar. The user will be prompted to enter their username and password and will be automatically logged in once an account is successfully created.
3. Logging In allows a new session to be created, which allows the user to stay logged in even after he leaves the webapp.
4. Users can verify the correctness of their password by clicking on the eye icon which will then convert their password to plain text.

Logging Out:

1. If a user wishes to log out of their account after they have signed in, they can do so simply by clicking on the "Log Out" link in the NavBar on the index page.
2. This action would destroy the current session the user is in and the user would be redirected back to the login page.

Note that all features mentioned from this point onwards would require users to be logged in first.

ToDoList Page:

1. Once the user has been logged in, they will be redirected to thier ToDoList page where they are able to view a table with their Todo items split by whether it has been completed or not.
2. The user can add a new task by clicking on the textbox below the "Add Todos" header and filling in a description of what needs to be done. The new Todo item would then be created once the user hits on their keyboard or when they click on the add button.
3. A search bar has also been provided for the user where they can easily search for specific keywords in their todo lists. Once the user starts typing any query into the search bar, the lists will be dynamically updated to filter out todos that (partially) match the keywords.
4. The user can easily toggle the completed states of their todo items by clicking on the checkbox under the "Done?" column. This will dynamically update the table and shift the

todo item to the correct table accordingly.

5. Todo items can also be easily edited by clicking on the task name. A textbox will appear and the user is free to update the name of the task. The task name will then be automatically saved once the user clicks away from the box.
6. The user can delete a todo item simply by clicking on the trashbin icon and the todo item will be dynamically removed from the table.

Possible Improvements:

Given the tight timeline and the fact that I had to learn most of the required frameworks from scratch, I was unfortunately unable to implement all the features that I wanted.

If given more time I would have worked more on these functionalities:

- Allowing the user to modify their account details
- Creating categories so that users can easily sort and manage their todo items.
- Creating a deadline feature
- Have a onboarding feature to guide first time users