

CVWO Mid-Assignment Write-Up 2022

Particulars

Name: Rayner Toh Jing Xiang

Matriculation Number: A0248846M

Link to GitHub repositories:

<https://github.com/raynertjx/forum-frontend> (frontend)

<https://github.com/raynertjx/forum-backend> (backend)

Use Cases

- User authentication
 - Users can create an account with a username and password to create threads or comments saved to their accounts.
 - Data validation to ensure that usernames created are unique and passwords are of the minimum length (6 characters)
- Thread/comment CRUD
 - Users can view threads and comments on the forums without creating an account.
 - Users must be authenticated to create, update, or delete threads and comments.
- Thread/comment organisation
 - Users can create threads in one out of five categories.
 - Users can add comments to any individual thread.
 - Threads and comments are displayed in descending order by date, with the latest at the top.
 - Threads and comments will overflow onto a new page once they exceed a certain threshold.
 - Users can post comments that reply to other users' comments.
- User experience
 - Dark mode option is made available for users, which they can toggle on/off.
 - Light/dark mode chosen using prefers-colour-scheme.
 - Users should stay logged in after page refresh unless they log out.
 - A “directory” path is included to let users know which part of the forum they are nested in.

Backend Choice (Ruby on Rails)

I have planned to utilise a Rails backend for this assignment. The front end would send requests to the server and receive updated data displayed for the user to interact with. I chose Rails due to the Model-View-Controller (MVC) nature of the framework, which allowed me to develop the CRUD functionality for this forum app.

Execution Plan

Type	Task	Done
Authentication	Sign up/log in pages.	✓
	Data validation of usernames and passwords.	✓
	Keep the user logged in using JWT tokens and HTTP-Only cookies.	✓
	Log out button for the user to terminate their session.	✓
App	Nested directory path for navigation within the app.	
	Redux state management.	✓
Threads	CRUD for threads.	✓
	Latest threads in the respective forum categories can be seen on the forum homepage.	✓
	See how many comments are in a thread from the subforum homepage.	✓
Comments	CRUD for comments.	
	Ability to reply to other users' comments.	
UI/UX	Compatibility and testing with different browsers: Internet Explorer, Firefox, Chrome, Safari.	
	Dark mode toggle.	
	Fix accessibility issues.	
Code Quality	Ensure that parameters or props are strongly typed using TypeScript.	✓
	Component abstraction and reusability.	
	Organise TSX files.	
	Proper linting and styling of code.	
Deployment	Deploy frontend and backend.	

Problems Faced

- Unfamiliarity with Rails, as this is my first time using it as a backend to communicate with the frontend.
 - **Proposed solution:** Consult Rails online documentation and various Stack Overflow posts to set up the app and troubleshoot issues (e.g. no CORS header).
- Styling components became messy with unorganised CSS files.
 - **Proposed solution:** Use Tailwind CSS to style the various components and reuse various styled TSX components to build the app.
- Certain frameworks needed additional configuration due to TypeScript (e.g. Redux).
 - **Proposed solution:** Consult TypeScript and Redux online documentation and add custom hooks to ensure strong typing across various methods and components.
- Unfamiliarity with storing JWT tokens in the HTTP-Only cookies, as most online tutorials and guides involved using LocalStorage, which is prone to XSS attacks.
 - **Proposed solution:** Consult rails online documentation and use CookieStore to sign cookies with JWT tokens for user authentication.
- Files are unorganised, and the project directory is challenging to navigate.
 - **Proposed solution:** Abstract reusable components and organise them into respective folders.
- Lag between fetching of data and displaying of data on the front end.
 - **Proposed solution:** Incorporate promises and async/await into my workflow and add loading components (loading spinner, etc.) if there is lag during data fetching. Redux can also store various data to prevent the unnecessary sending of API requests.
- Lack of navigation in the front end.
 - **Proposed solution:** Use React Router to create links between different app pages and create a directory path on top of the main container to let the user know where he is nested in the forum.
- Certain styling may be incompatible in certain browsers.
 - **Proposed solution:** Use Tailwind CSS to ensure browser compatibility across the various browsers.İ