

CVWO Final Assignment

Technology used

ReactJS
Ruby on Rails
Chakra UI (Library)
Postgresql14

Deployment Website

Frontend: <https://63d11b879ad4fb16c6b67cf1--capable-tartufo-82bcbc.netlify.app/>
Backend: <https://cvwo-backend-7cq1.onrender.com/>

I built the frontend using ReactJS, the backend was using Ruby on Rails , and the backend was using Postgresql

The inspiration behind the website was to be a gossip/forum site and I tried to include design elements that would seem more “gossipy?” and “trendy?” hence the choice of selection of words such as tea/fluff/rant

The user would be able to delete posts that they made as well as to make posts with certain tags such as tea/fluff/rant.

There are 2 primary walls, one being the overall user wall which would display the all the posts from which people have commented across the website, and a personal wall on the profile website which would display the individual who has logged in and their posts

I managed to successfully deploy it onto netlify for the frontend and render for the backend I wish to do level 5 account authentication with JWT tokens if i had the time but i only managed to complete it halfway due to time constraints. So instead when a user logs in it will send a fetch request to check if the input username is in the database and if the password matches the password stored in the database. I did not manage JWT and how to go about using them and hence did implement this “self authentication” instead so that the user has some form of account stored albeit it being not very secure

User experience(how I imagine a user would go about)

When the user enters the website, it brings them to the home page where they can view the website and posts but are unable to make a post. There is a sign in button that will redirect them to the sign in page and a prompt for them to sign in

In the sign in page, there will also be a sign up button for users who have yet to sign up.

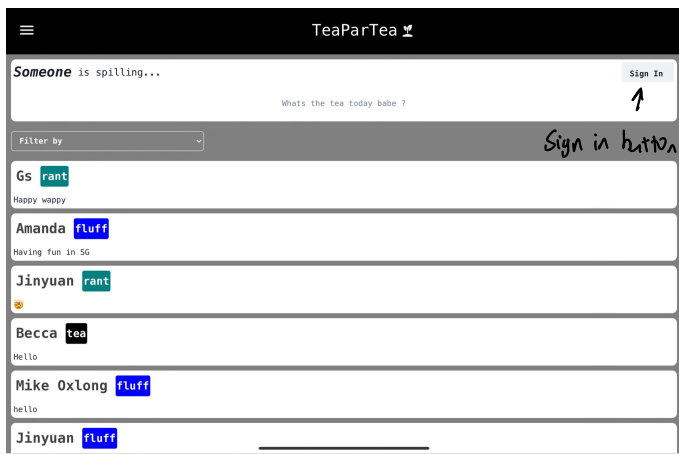
Upon successful login, users will then be able to access the website and view the posts that others have

made. users will now be able to either make a post with the relevant tags or be able to scroll through what others have commented and are thinking

Developer experience

As I went about this project, it definitely was an amazing experience. Previously i had only built static website with basic css, or only could host my websites on localhosts. I was really happy with my result and end product. For me, my greatest challenge was learning the new technology as well as with regards to the backend. Ruby on rails is new terrain for me and thus initially I had to rely heavily on tutorials in order to understand what was going on. As mentioned above, I would have wanted to build a more secure user account authentication but due to lack of time I did not manage to learn about JWT and other session authentications. It was fun figuring out workarounds and experimenting with the data fetched from the backend. There were times where I would get lost and confused and wanted to give up but the idea of having my own blog sounded cool and I decided to continue with it

I think I learnt a lot doing this project, and I actually spent most of my time experimenting with different features and googling how to achieve certain features. Waiting anxiously for my deployment to go live and getting stumped at the ruby controllers definitely thought me much more about web development than I knew. If given time I would definitely wish to work on this project more by adding more features such as friends and personal wall bios. It was quite perspective to implement the website in such a way that I would imagine a user would be able to navigate and use with ease



Main page, user not logged in

