CVWO Final Submission Writeup (A0255768L)

After more than one month of work, I am glad that my application is finally done. Ruby on Rails and React are very interesting and I learned a lot at the past month.

I started designing the models of the data. The main models being used in this app are post, comment, tag to user. The most important thing I learned here was about handling many-to-many relations. I was amazed by how using an intermediary model can solve this problem and I implemented this to associate posts and tags.

After modeling the data, I moved on to implementing the controller and routes at the backend. I figured that implementing user authentication and authorization in Rails early on would make things more organized. As a result, the backend was completed very smoothly. Later, I also discovered the use of serializers in Rails and it saved me a lot of time.

After most of the APIs were ready, I finally moved on to building the frontend of the application. The most challenging part of building the frontend (or perhaps the most challenging part of this project) was getting Redux to work. Redux is quite challenging to set up. But fortunately, after reading its documentations, I eventually understood the concepts and I applied it to my application. Things went smoothly from there onward.

Before building this app, I already had some experience with Vue and Django. These two are like my mother tongue, my gateway to web development. Hence, to learn another framework like Ruby on Rails and React is like learning a foreign language to me. Although web frameworks are often similar, each web framework has its own design, and its own opinion when it comes to solving problems.

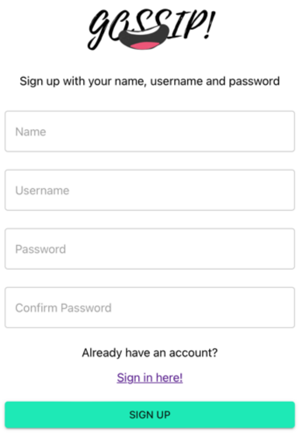
Initially, I thought that having prior experience with other frameworks will give me an advantage. However, I was held back by my own prior knowledge in other frameworks because of the presumptions I have. For example, database migration works slightly differently in Django and Rails, and I spent a lot of time to figure out how things work in Rails because I kept expecting it to work the same way as how it does on Django. React is not actually a framework, it is a library, it does not have everything built-in like how Vue does. It takes a lot of time to learn how to implement a lot of the features I got on Vue for granted on React.

I guess the biggest takeaway of this project is: to learn something new, we must first unlearn other things in some way. Having presumptions on how things work can only stop us from making progress. This does not only apply in programming and web development, but also in everything else in general. If we are not willing to look at things differently, we will be stuck with no progress.

User manual

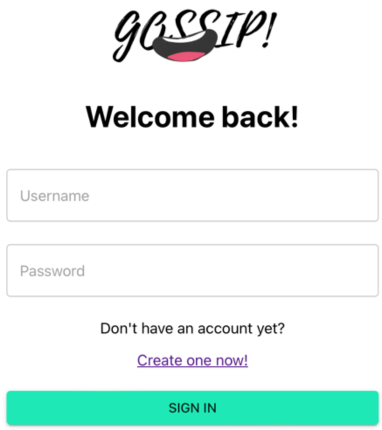
Site is hosted on: https://gossip-nus.netlify.app/
The following user is created for your testing purposes:
username: cvwo_reviewer
password: cvwo_reviewer_password

Sign Up and Login



Sign up with your name, username and password

Already have an account?

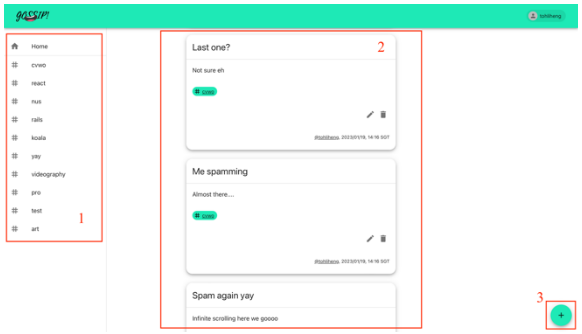


Welcome back!

Don't have an account yet?

You can create an account with your name, username and password. You can then login with your username and password.

Browsing, Creating and Editing Posts



Last one?

Not sure eh

@tohliheng, 2023/01/19, 14:16 SGT

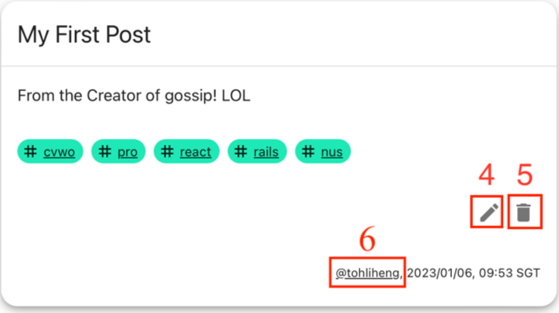
Me spamming

Almost there....

@tohliheng, 2023/01/19, 14:16 SGT

Spam again yay

Infinite scrolling here we goooo



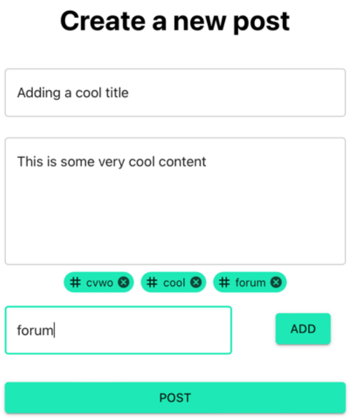
My First Post

From the Creator of gossip! LOL

@tohliheng, 2023/01/06, 09:53 SGT

1. The side bar shows a home button and top 20 most used tags. You can browse posts based on a given tag.
2. The middle part of this page is sorted in descending order of created dates.
3. The lower right button allows you to create a new post
4. For the posts you created, you can edit them.
5. For the posts you created, you can delete them.
6. You can click on the author of a post to view their profile, which lists out all the post they posted.

Infinite scrolling is implemented in pages that shows posts and you can browse more posts just by scrolling down.

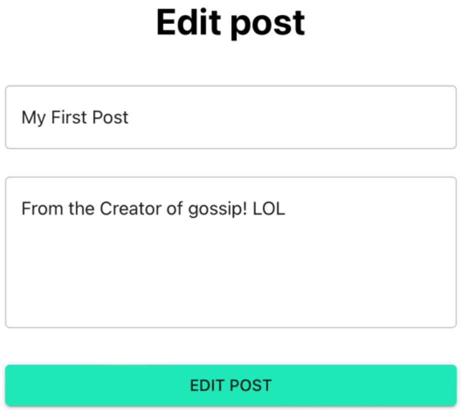


Adding a cool title

This is some very cool content

cvwo # cool # forum

forum



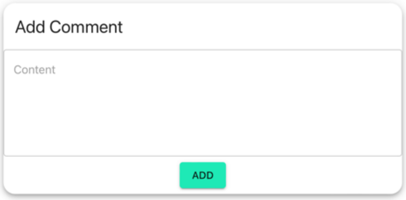
My First Post

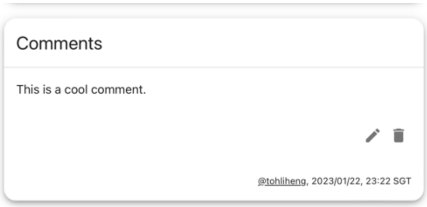
From the Creator of gossip! LOL

When creating a post, a title is required and must be at least 8 characters long. Content and tags are optional. Tags must be 2-20 characters long, with lower case alphabets, numbers, underscores, and no spaces. Each post can have at most 5 tags.

Once you created a post, you can edit its title and content, but not tags. Posts that were edited will have an "edited" label next to its posted date/time (edited date/time is not shown).

Commenting



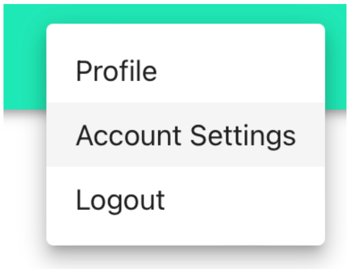


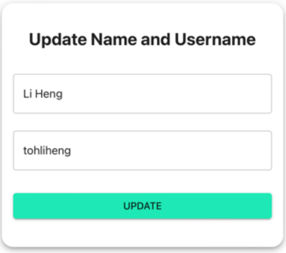
This is a cool comment.

@tohliheng, 2023/01/22, 23:22 SGT

You can leave comments on a post. The comments you left can be edited and deleted, just like posts. Comments are also sorted in descending order of created date/time.

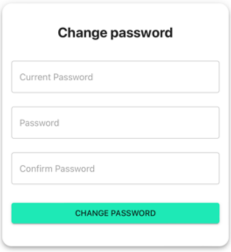
Account Settings

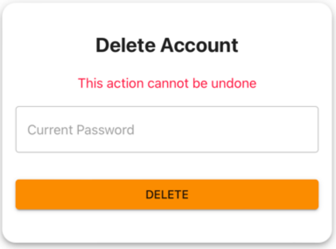




Li Heng

tohliheng





You can access account settings by clicking on your user avatar on the upper right and select "Account Settings". You can then change your name, username, password and even delete your account.