

Name: Wong Li Yuan  
Matriculation number: A0262316L  
Website: (Recommended browser: Google Chrome)

### Final Write-up

Being new in web development I was quite clueless when I first started the project. I started the project by joining a Ruby Course to learn the basic syntax. Then, I looked up the MVC model and was able to generate my models and controllers using scaffolding. I have also set up my Rails router that dispatches the URLs to my controllers' actions.

For the database, I have used PostgreSQL for the future deployment on Heroku. I have drawn an entity relationship diagram to illustrate the relationship between models and created my relational database through primary and foreign keys.

For the account authentication, I made the username a unique entity and used the Bcrypt gem to make the password secure. When I was looking for a way to maintain session data, I came across the Session Store provided by the Rails Framework. After a few attempts, I managed to make the session data persist after each render by creating the Session Controller and setting up the Session Store.

I learned a lot from the CORS mechanism when I was setting the link between my frontend and backend. I have also understood the concept of HTTP response and request headers when handling with special data (session data, picture data) and the different data formats used in API requests (json and FormData).

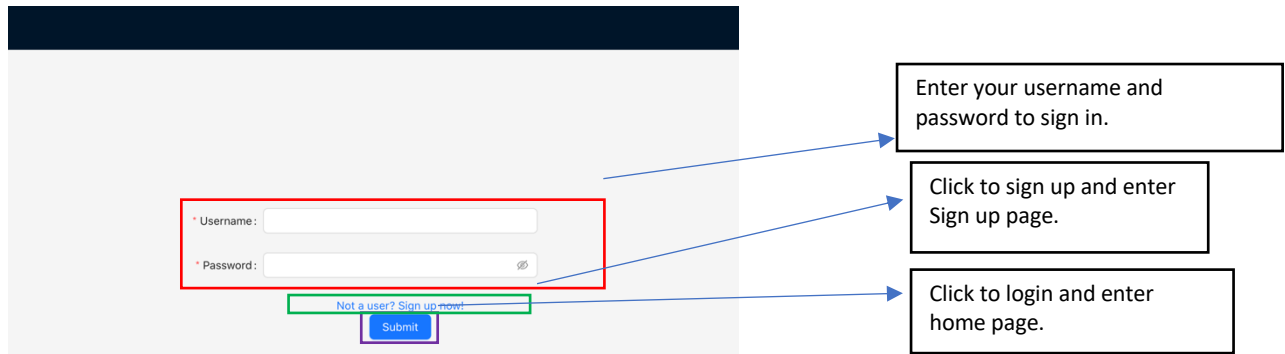
I have utilized the Ant Design Library to add user-friendly features to my website. By splitting my components into separate files, I can introduce code reusability to my program and better structure my code.

To manage states across the React framework, I have used *createAsyncThunk* and *createSlice* in Redux Toolkit. I have created several slices to store the initial states of my entities and the extra reducers to handle the statuses of my async actions i.e., *pending*, *fulfilled* and *rejected*. For every slice, I have created API requests to send/request data to the backend.

Along the way, I have encountered bugs that took me a lot of time to fix. By browsing through online forums and websites, I have successfully fixed them. After reaching the end stages of my project, I realize that while debugging is a painful process, I really learned about a lot of the underlying operations behind boilerplate code. I also started to appreciate reading documentations more as I wanted to add additional features to my program. At last, I have deployed my website and server on Heroku.

I feel thankful for the people who shared their knowledge through tutorials. I definitely feel more confident in creating new projects, but I feel that I have still a lot to learn. Regardless of the final outcome, I am proud that I was able to finish the website.

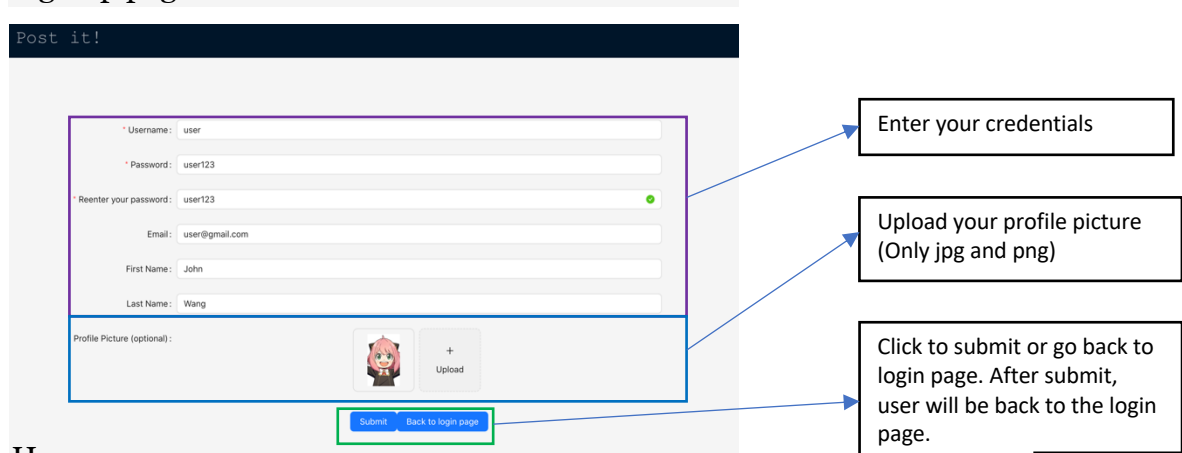
## Login page:



The login page features a dark blue header with the text "Post it!". Below the header, there is a login form. The form includes a red-bordered box containing the "Username:" and "Password:" input fields. A green-bordered box contains the link "Not a user? Sign up now!". A blue-bordered box contains the "Submit" button. Annotations with arrows point to these elements:

- Enter your username and password to sign in.
- Click to sign up and enter Sign up page.
- Click to login and enter home page.

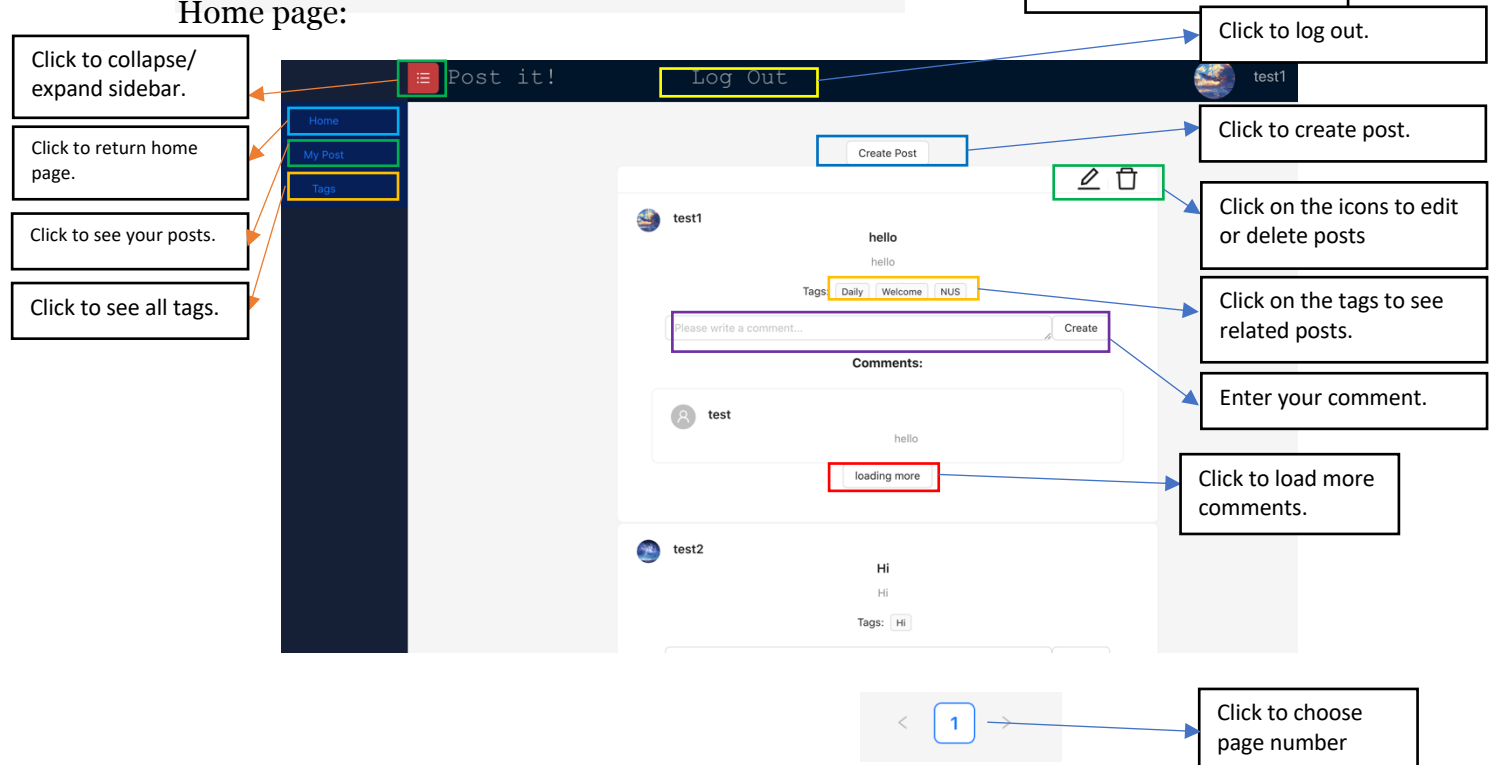
## Sign up page:



The sign up page features a dark blue header with the text "Post it!". Below the header, there is a sign up form. The form includes a purple-bordered box containing the "Username:", "Password:", "Reenter your password:", "Email:", "First Name:", and "Last Name:" input fields. A blue-bordered box contains the "Profile Picture (optional):" section with a profile picture icon and an "Upload" button. A green-bordered box contains the "Submit" button and a link "Back to login page". Annotations with arrows point to these elements:

- Enter your credentials
- Upload your profile picture (Only jpg and png)
- Click to submit or go back to login page. After submit, user will be back to the login page.

## Home page:



The home page features a dark blue header with the text "Post it!". Below the header, there is a sidebar on the left and a main content area. The sidebar contains a red-bordered box with a hamburger menu icon, a blue-bordered box with "Home", a green-bordered box with "My Post", and a yellow-bordered box with "Tags". Annotations with arrows point to these elements:

- Click to collapse/expand sidebar.
- Click to return home page.
- Click to see your posts.
- Click to see all tags.

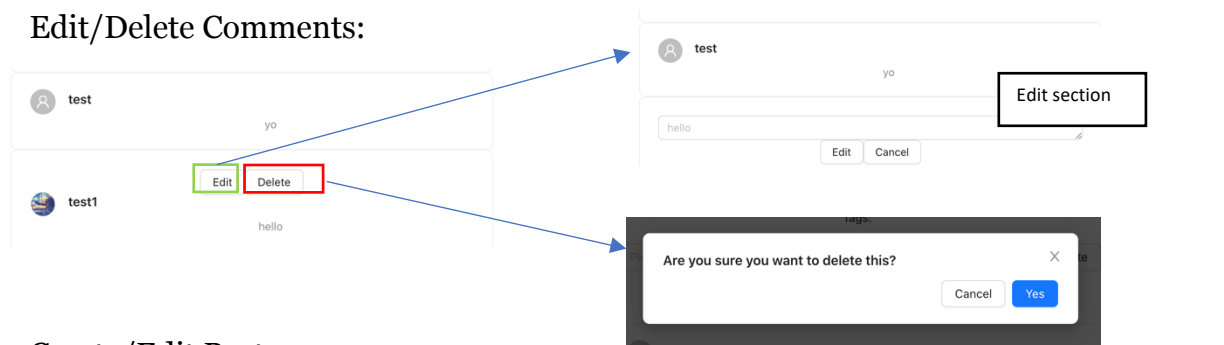
The main content area includes a "Log Out" button in the header, a "Create Post" button, a post by "test1" with a "hello" message, tags "Daily", "Welcome", and "NUS", a comment input field, a "Create" button, and a "loading more" button. Annotations with arrows point to these elements:

- Click to log out.
- Click to create post.
- Click on the icons to edit or delete posts
- Click on the tags to see related posts.
- Enter your comment.
- Click to load more comments.

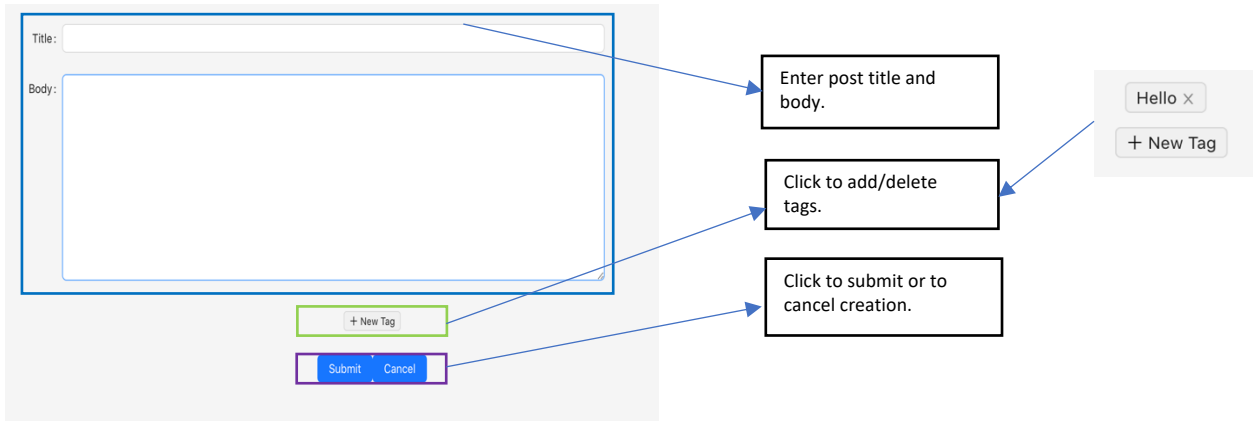
At the bottom, there is a pagination bar with a blue-bordered box containing the number "1". An annotation with an arrow points to this element:

- Click to choose page number

## Edit/Delete Comments:



## Create/Edit Post page:



## Tags page:

