

## **User Manual**

Run bundle install. Run command rails db:seed. Run bin/dev to open local host. Then sign in using account user1, password:user1 and start exploring the forums and perform CRUD on forums.

1. User has to login before proceeding with the features in the website
2. User able to sign up a new account
3. The login status will be stored in web local storage so that user will not need to login every single time when wish to use the website
4. User able to create, read, update and delete (CRUD) forums.
5. User able to comments on a particular forum.
6. User(Admin) able to use the categorize system to do some changes on type of categories.
7. User able to click on interested categorize to see related forums

## **Write up**

I have learned Ruby on Rails and React separately as pure backend and pure frontend languages. During the learning, I have set up few Rails projects with provided frontend features. It helps me to learn the core concepts of the software engineering skills and solidified my idea to come out with a food ordering app including a discussion forum. Then I proceed to learn React and set up a few project examples using React and learn some fundamental concept from React. I have also learnt fundamental concept of React involving Hook – useEffect, useContext and some other frequently use Hook concept. Then I realized that I am lacked of basic HTML and CSS knowledge and so I spent a few days to learn the fundamental of HTML and CSS which gives me a better image on how to use React as frontend.

When I prepared myself with enough basic knowledge, I proceed to set up a project with Ruby on Rails as backend and React as frontend. The difficulty is so terrifying and complex. The way of setting up the projects was much more difficult than purely using Rails to set up backend and frontend together. However, I found it interesting that using rails to communicate with react by setting rails as API. This project has given me a clearer image of how real-life applications work. And so, I decided to continue my learning journey in CVWO. The amount of workload was so much higher from my expectations.

The first big difference that I have encounter is to do routing. The set up guide helps me a lot in understanding routing and GET/POST request. It is so fascinating to find out that doing routing in frontend, we are able to display different jsx file in order to display correct page. The concept is not just routing, and also involve API. Since we are doing a dynamic website, we have to receive information from our backend. I spend some time to understand the process of sending a GET request to backend, then backend will send a JSON data to our frontend, and we have to decrypt the data to use in our jsx file.

I spent a lot of time to understand how does sending of request to server works. I spent my time to understand GET request, as well as POST request in order to communicate with our backend code. A basic CRUD is much more difficult in this project compared to only using Rails as backend and frontend. So I start practicing it by creating a forum page with basic CRUD features. I also practice my database knowledge here by creating a one-to-many relation between forum and comments. This nested relation have further strengthened my knowledge in using GET and POST request handling different situation. I have also getting a clearer image on what does JSON send to React and how do we handle the JSON information.

After getting a clearer image on GET and POST request, I continue to learn authentication system. I did some research on how authentication system works and how caching works. I was so clueless and helpless in order to understand the whole structure until I spent another few days to research on relationship between JSON, Caching, Web Local Storage and authentication system. Then I create a user model with basic CRUD. When moving to authentication system, it is far more difficult than expected. I spent a lot days to debug and set up. After I have successfully set up an authentication, I realized I am able to improve this basic authentication by using React Hook – useContext. Then, in most of the routing, I will be able to share the information I keep in one context file and reuse the feature that is required in every page. I keep my login information in this context file and send in a data into web local storage so that the web knows that we have already login. During routing, react will check the web local storage on whether is there any login information. As soon as it receive a login information from local storage, useContext will send to my context file, telling my react that user have already login/ haven't login, the it will send to other page to control whether is the user able to visit the page. This implementation will allow user don't have to login as soon as they refresh the page. I am going to implement admin method to restrict some of the features so that some features are only accessible by admin, but due to time constraints, I will only implement this features in future.

After authentication system, I continue with tagging system on forums. I build up a relation between forums and categories so that user able to categorize each forum when they create the forum. I also spent some time to debug when I was trying to show out categorized forums chosen by user. But at this stage, it is far more intuitive than my previous stage since I am getting a clearer picture on this project.

Finally, I proceed to UI of the project. I realized it is too complex and time-consuming for me to write out every single HTML CSS elements by myself and required a lot of time in doing a design planning. And so, I spend a couple days to research in MUI, which is REACT library. I found out that it is quite intuitive to use the library since I have spent some time in learning fundamental of HTML and CSS. But still, designing a user-friendly UI is very time-consuming and at least I come out with a basic UI which is quite clean after a few days work.

Due to time constraints, I am not able to come out with a 100% food-ordering system written in my proposal. The difficulty of separating backend and frontend is underestimated. There is a lot of features that I wish I am able to improve and implement in my project. Throughout this project, I have learned a lot of knowledge and I wish to continue this journey to learn more. I am confident that I am able to implement my food-ordering system as I continue this journey with a longer period of time. I am also wish to deploy my website and hopefully it will be done as soon as possible.