Chrystal Quek Wan Qi

A0206350U

Final Writeup for CVWO

ACCOMPLISHMENTS

As I had no experience with Ruby on Rails and React, this project proved to be challenging. I started by learning basic concepts of Ruby, which was not too difficult as it was a beginner-friendly language. There were many great online resources for learning Ruby and Ruby on Rails and my primary learning source was the Odin Project. After that, I learnt Ruby on Rails and the Models, Controllers, Views model. What I found most challenging was the concept of routes, where API calls from the React Frontend had to correspond exactly with predefined routes in Ruby on Rails. After building a foundation in Ruby on Rails, I started learning React and was introduced to concepts like components, state and rendering. React was exciting to learn as there were visuals that accompanied the modifications I made to the code.

Upon learning Ruby on Rails and React separately, the difficulty came in linking them together. There were multiple ways to go about doing this. I started off using Webpacker but there were difficulties as Rails itself has default 'views' but React was supposed to render the pages. I decided to make a major switch at the end of December to separate my frontend and backend into two separate repositories. I felt that the different ports would make it easier to differentiate the frontend and backend but it also raised issues with CORS during hosting.

Another major issue I had was providing user authentication in the web application. Originally, I did not intend to have user accounts but after much consideration, I realized that it would not make sense if any user could edit a public Todo list. User authentication was tricky as some sources encouraged the use of JWT tokens while others promoted sessions. I started using session based data but quickly ran into problems where a simple refresh of a page would wipe out all the data. After research, I change to JWT tokens stored in localStorage instead and that required changes to both the frontend and backend.

After completing the project locally, I hosted my web application with the backend on Heroku and the frontend on Netlify. I faced a CORS policy issue for a few days despite including ‘rack-cors’ in the gem file and making modifications to require files. It was not until I executed the command ‘gem install rack-cors’ that solved all the problems. Previously, I had even moved my frontend from Heroku to Netlify as I thought that it could solve the problem.

My greatest takeaway, as cliché as it sounds, was learning how to persevere despite the difficulties of the project. Each step was fraught with problems which if I was lucky enough, required simple fixes, but mostly needed much research and testing. My first plunge into web development has been extremely memorable and fruitful.

LIMITATIONS

One major limitation was that I used CSS for styling although Bootstrap is conventionally used. Hence, my web application is not device optimized [currently optimized for a MacBook Pro 13inch display]. I was a little uncertain about using Bootstrap as it was my first time styling and I felt that it was more important to have control over the CSS styles I used initially. For future projects, I would employ Bootstrap as it is more efficient.

Some of the code felt repetitive and it would have been better if the code was repackaged for easy reuse.

CONCLUSION

Overall, I have learnt valuable frontend and backend skills for web applications that can be used for future projects. Although it took me a long time to learn web development, I feel that I have really gained a lot from this experience and I hope to utilize these skills for real-world applications. There are many areas of improvement, but I hope that I have shown my interest and desire to improve on my skills.

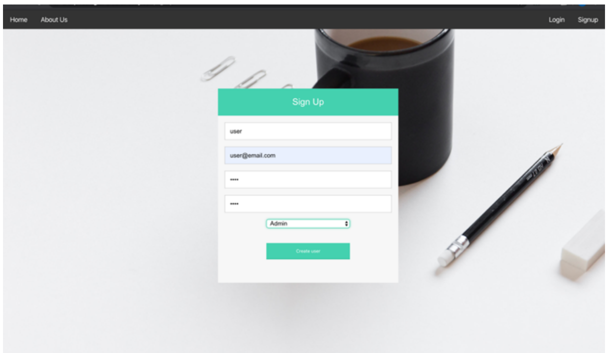
This semester, I will be learning TypeScript and Redux as part of an independent project module CP3108.

Basic User Manual

Navigate to https://sharp-roentgen-6f32fa.netlify.com/. If you do not have an account, click Sign Up. Otherwise, click login.

* The application may be a little slow if you have just started to use the application.

Sign Up / Login



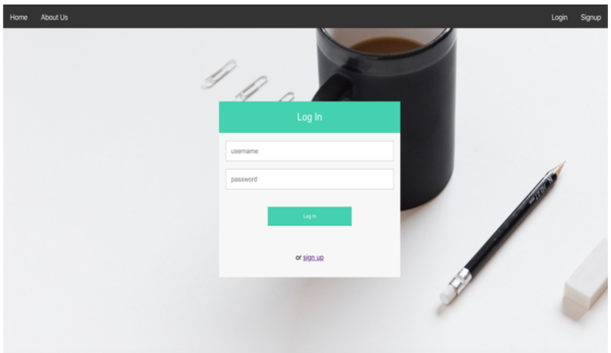
user

user@email.com

••••

••••

Admin

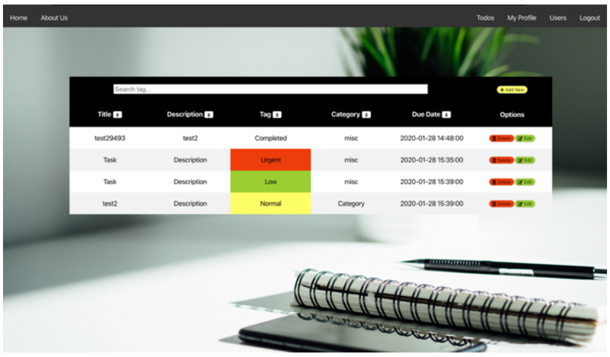


Your username and email have to be unique and the password has to be at least 4 characters long.

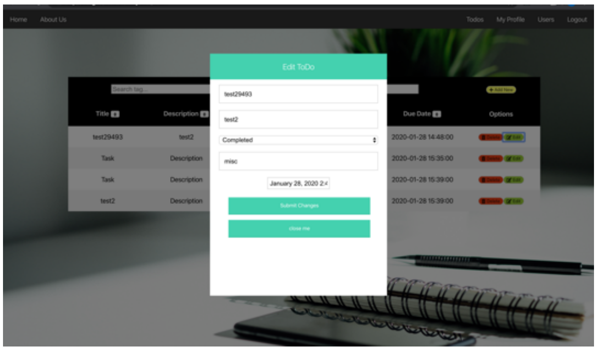
Choose either Admin or User role in the dropdown list. An admin role has more options as he can edit, delete and create other users and admins. To experience the full functionalities of the web application, select the admin role (default).

Upon sign up (and login), you will be directed to the Todos page.

Todos



Title	Description	Tag	Category	Due Date	Options
test29493	test2	Completed	misc	2020-01-28 14:48:00	Delete Edit
Task	Description	Urgent	misc	2020-01-28 15:35:00	Delete Edit
Task	Description	Low	misc	2020-01-28 15:39:00	Delete Edit
test2	Description	Normal	Category	2020-01-28 15:39:00	Delete Edit



test29493

test2

Completed

misc

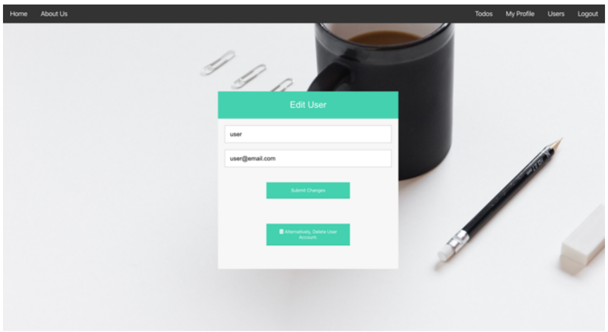
January 28, 2020 2:4

You can create, update and delete your Todos on this page.

When you create and update your Todos, ensure that your Title and Category field is not blank. Select a tag: Urgent, Normal, Low, Completed which will be colour-coded in the Todos table accordingly.

The due date is selected by default to be the current time but use the datepicker.

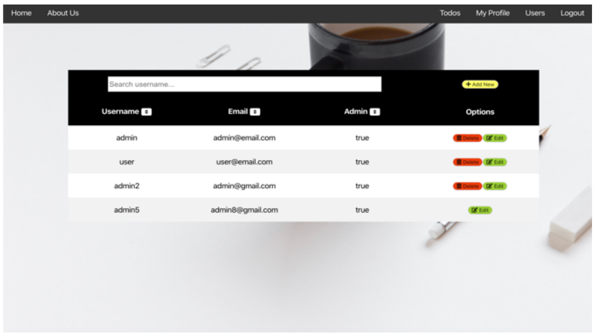
Edit Profile



user

user@email.com

Users [only for admin roles]



Username	Email	Admin	Options
admin	admin@email.com	true	Delete Edit
user	user@email.com	true	Delete Edit
admin2	admin@gmail.com	true	Delete Edit
admin5	admin8@gmail.com	true	Edit

Edit your username and email under the tab “My Profile”. Create new users, update existing users or delete users under tab “Users”.