

Final Submission Write-Up

These two months have been very fruitful as I've picked up a lot of new skills and gained insights into software development. I realised that many technologies and concepts could be self-taught using a vast amount of information (StackOverflow and guides). Something that I have severely underestimated was the difficulty of deployment. After completing the application's coding aspect, I thought I was mostly done. I felt that deployment would take only two days maximum as I didn't understand much about infrastructure. However, the more I researched Amazon web services(AWS), I realised that I knew just the tip of the iceberg. There were many days where there was no progress, and I felt like giving up many times. The sheer amount of information overwhelmed me, especially on the Amazon web services. However, I still managed to push through and continue. Initially, I saw Elastic Beanstalk and online forums said that it was much easier than using an EC2 instance. However, I decided to take up the challenge since I can control a lot more things using EC2 even though the learning curve is much higher. I've learnt many of the DevOps related technologies such as reverse proxy using Nginx and security groups and ports. This was indeed an eye-opening experience as I know that front-end and backend development is just one aspect, and I have to put equal focus on DevOps as well.

What I think could have been improved on:

At the start of this project, I wanted to do some level of project management myself. I intended to use Trello to create Task cards and set deadlines for each mini task. However, I only did it for the first few tasks, and as the weeks progressed, I lost the discipline to update the task on Trello constantly. As a result, I didn't complete all the features I hoped to implement, such as automated testing and sign in/sign up. Having project management skills is extremely important as it allows developers to prioritise the most critical and urgent tasks. This ensures that the essential features are presentable even if there is a lack of time.

What I want to accomplish in the future:

I wanted to write automated testing for the front-end using Jest. However, I prioritised DevOps and decided to focus on docker and AWS with my remaining time since they were part of the grading criteria. However, if I had more time, I would use Jest to perform automated front-end testing. For the backend, I would add in some tests as well. Other than that, I only managed to set up authentication for development and did not managed to deploy it to production. What happened was that the 3rd party library(Auth0) that I was using required me to use https in order for the API authentication to work. This meant that I had to create an SSL certificate and reconfigure my EC2 instance to serve the website on https. However, I didn't have enough time to learn how to do that. This was a mistake on my part as I did not know that I would need to do that .

Final thoughts :

Although the app is not perfect, I'm still proud of what I've accomplished these two months. When I finally got the app deployed and could access it from a public ipv4 link, it was an extremely satisfying feeling, especially when I've spent countless hours just trying to get the app deployed on AWS. Learning all these new technologies has been enjoyable. Thank you! I

enjoyed the process these two months, and I hope to have the opportunity to create impactful applications for CVWO.

Level 4: App deployed on:

<http://ec2-13-214-133-50.ap-southeast-1.compute.amazonaws.com/>

Github version which app was deployed on: 19c40c032d77e6c3a20b110dc4353fc4b0407af0

As I only managed to set up authentication on development mode, this feature is not available in production mode. The latest commit in my repository contains the newest features, however the app is deployed with an older commit (commit id above)

Level 5 deliverables:

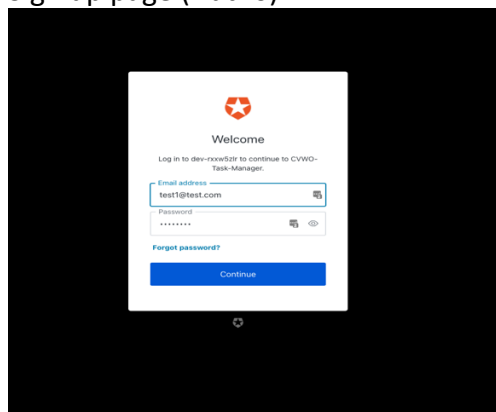
1. Typescript
2. Redux: Used to store all the tasks and also the token for jwt authentication and also loading states for authentication is stored using redux as well.
3. Docker: Used docker and docker compose to easily run application in development mode. Simply run docker-compose up in root directory to get the whole application (both backend and frontend) running.
4. AWS: Deployed both backend and frontend using AWS EC2 instances (ubuntu) and used Nginx to serve the frontend application and also act as a reverse proxy server to direct request to the Go backend.
5. Process Management solution for go backend using systemd to automatically start the backend service and restart it if it crashes. (Can be found in github backend.service file)

User Manual: (development):

Email: test1@test.com

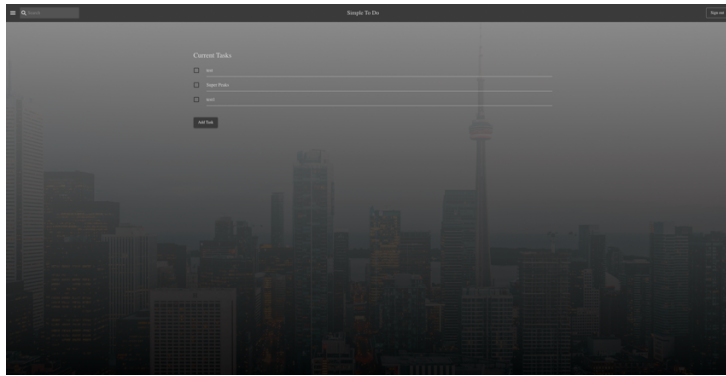
Password: Test1234

Sign up page (Auth0) :



Main Page:

- Only the tasks that are not completed will be seen on this page.
- Users can mark a task as completed
- User can click on the task to enter the editing screen
- User can click on the add Task button to see the Add Task modal



Add task modal:

- User can fill in the title, description and priority of the new task
- When the Add Task button is clicked, the menu will disappear and the new task will be shown on the main page

Edit task modal:

- When the user clicks into a task, the Modal will pop out
- The form will be filled with the current task details
- The user can delete the task on this modal
- Once the user is done editing the task, the user can click confirm changes and he/she will be redirected back

Other pages:

Completed Page: Only completed tasks will be shown on this page

Low Priority Page: Tasks that are marked low priority will be shown on this page

Medium Priority Page: Tasks that are marked medium priority will be shown on this page

High Priority Page: Tasks that are marked high priority will be shown on this page