

## CVWO Writeup

### A. Use Cases:

- Users would be prompted to register for an account with their email before being able to create, view, edit and delete task on the frontend website.
- Users would be able to create, view, edit and delete their tasks on the frontend website according to their daily routine after signing in. When creating a task, the user would be prompted for the name of the task, the deadline and optionally the priority and category of the task. After the task is created, the user would be able to update all the details mentioned previously.
- Tasks due on the same day would be displayed separately from planned future tasks on a different page
- Users will also be able to filter/search for tasks based on the name, deadline, priority, and category of the task

### B. Software Used:

- The front will be built using React and Typescript, utilizing various React libraries to perform operations such as form validation etc.
- The backend will be built with Go, with MongoDB being used to store user data and Redis being used to blacklist JWT tokens when a user chooses to sign out before their JWT token expires for better security

### C. Execution Plan:

1. Initialize MongoDB and Redis Database
2. Define user and task models in Go backend application
3. Implement RESTful APIs in Go backend application
4. Implement React frontend application and connect to the backend APIs
5. Upload application to Git
6. Create Docker images for both the React frontend and Go backend applications
7. Deploy both Docker images to AWS EC2