CVWO Mid-Assignment Writeup

Name: Bryan Lim Jing Xiang

Matriculation No: A0233605M

Overview:

As per the requirements for CVWO AY2021/22 winter assignment, this will be a Todo/task management web app, using React for the frontend and Ruby on Rails for the backend. This will be a pretty minimal application designed to be smooth and responsive so as to maximise user experience.

Use Cases:

- Todos
  - Users create/edit/view/delete todos (CRUD operations)
- Tags
  - Users create/edit/view/delete tags (CRUD)
  - Users add/delete tags from todos
  - Users filter todos based on selected tags
- Search functionality for todos
  - Users search for todos based on keywords

Backend: Ruby on Rails vs Go/Golang

This is more of a choice of practicality rather than one of design. Personally, web development is something very new to me. In fact, I have only really gotten into programming in cs1101s during sem 1 this year. Alongside other commitments I had during the winter break, it is simpler much easier and faster to stick to a framework like Rails instead of implementing the backend from the ground up with go/golang

Furthermore, being a fairly simple and minimal app, Rails allow for much faster prototyping and development by sticking to convention rather than configuration. Rails also abstracts away many of the underlying details which we would have to otherwise implement in go/golang.

Execution plan:

- Frontend: React
- Backend: Ruby on Rails

- State management: Hooks and Context Api
- Styling/Aesthetic: Material UI/Bootstrap
- Deployment: Heroku

To minimise the number of requests/queries made to the backend, the search and filtering functionalities will be implemented in the frontend using React instead of Rails. This should also better align with the concept of a Single Page Application.

For state management, I will most likely be using hooks and Context Api. This will be needed for todos and tags, as well as some of the UI elements.

Styling will most likely be done using Material UI, with Bootstrap as the backup plan.

Deployment will be to Heroku. The frontend and backend will also be developed separately using the –api option to create a Rails Api.

Database: PostgreSQL

Database would be PostgreSQL which is also a requirement for deploying to Heroku. For this app, there will be two models/tables: todo and tag. A todo can have many tags and a tag can have many todos, which is a many-to-many relationship. This will be implemented in Rails using the has_and_belongs_to_many Active Record association.

Extra functionalities/future improvements:

- Login authentication and account management
- Pagination when there is too many todos
- UI/UX improvements
- Using cron for scheduled backups of database, as well as compression and archival of todos completed long ago
- Deadlines and reminders
- Calendar or simply integration with other apps such as Google Calendar
- Implement testing
- Possibly scale it into a Progressive Web Application
- Other optional requirements for this assignment
    - Redux
    - Docker
    - AWS