CVWO Mid-Assignment Submission

Siddarth Nandanahosur Suresh
A0206124W

Github Repository: https://github.com/siddarth2824/CVWO-Assignment

Basic Use Cases

- ***Task Components***
  A todo task will have an item name, description about the task, due date for when the task needs to be completed by. Each task can be associated with many tags.

- ***Home Page***
  The home page will require users to sign up with their email and a password. Users who have already signed up, will log in with their registered email and password. Upon logging in, users will be able to enter to view his/her tasks.

- ***Index Page***
  The index page of the app will allow users to view all their individual todo tasks. The tasks will be differentiated according to whether it is an ongoing task or a completed task. Users will be able to mark a task as complete which will then push the task to the completed section. The index page will only show the item name of the todo task. Users can click on the item to display the description of the todo task. Each task will have multiple buttons to provide additional functionalities for users. These buttons include viewing of tasks, editing of tasks, marking a task as complete and the ability to delete a certain task.

- ***Creating New Todo Tasks***
  Users can create new todo tasks provided they provide a title for the task. Users will be able to do so by clicking a button in the index page.

- ***Viewing Todo Tasks***
  There will be a button for users to view more details about the task. Details include: task description, when the task is due and what tags are associated with each respective todo task. Additional buttons, including editing the task, deleting the task and marking a task as complete, will also be available.

- ***Editing Todo Task***
  Users will have the ability to edit their tasks upon creation of tasks. All attributes of the task will be able to be edited by the user including removing and adding of tags. Users will be able to associate the tasks with new tags.

- ***Deleting of Todo Tasks***
  There will be a button in the index page for users to delete the task. This function will also be available when users view a task.

- ***Adding a Tag to Associate with a Todo Task***
  When a user creates a task, they will have an option to create a tag to associate with a specific task. One task can have many tags and one tag can be associated

with many different tasks. Hence when the user views a todo task, they will be able to see the tags associated with the respective todo tasks. Users will be able to click on the tag to view all the other todo tasks associated with a specific tag.

Execution Plan

- ***Phase 1 (Basic Implementation)***
  - Create a todo application which will allow users to perform CRUD operations for a todo task item.**[Completed]**
  - Each item will have:
    - Todo Task Item Name: a simple string to summarize the task **[Completed]**
    - Task Description: text containing further details about the task **[Completed]**
    - Completed: a Boolean to represent if a task has been completed or not **[Completed]**
    - Timestamp: Date and time created and modified **[Completed]**
  - Each task will have tags associated with it. **[Completed]**
  - Users will be able to view all tasks related to a specific task. **(Having issues to integrate this as I still trying to figure out how to add a custom method to process the tags created by current_user for Devise Gem)**
  - Generate MVCs for my main components
    - Tags **[Completed]**
    - Taggings **[Completed]**
    - User **[Completed]**
    - Tasks **[Completed]**
- ***Phase 2 (Advanced Implementation)***
  - The application will be able to authenticate users and display only the user's tasks. **[Completed using Devise Gem]**
  - Tasks will have a due date to allow users to view when the task is due by. **[Completed]**
  - Users will be able to view ongoing and completed tasks separately. **[Completed]**
  - Allow users to sort tasks in order of urgency.
  - Create a search bar to allow users to search for tasks with respect to todo task title, due date and tags.
  - Create frontend with React.js
  - CSS styling
- ***Phase 3 (Optional Tasks)***
  - Plan to host the application on Herouku (Very important)
  - Incorporate Typescript (Very important)
  - Implement Redux to manage the states for the React components for the application's frontend (Very important)

Problems Faced

- Currently the biggest issue I am facing is the inability to merge the frontend aspect of the application with the backend aspect. I have done research on how I can combine frontend (React.js) and backend (Ruby on Rails), and I have found out that there are many different ways which that can be done. However, I am not sure on which is the best way as I am close to completing the backend aspect of my application. Hence, I am having a conflicting idea on whether I should create a separate frontend and use API requests for the frontend to communicate with the backend or I should just add the react components and change the view part of the MVC in Ruby on Rails.

- Since I am new to frontend and backend development, I have encountered many issues while trying to create this application. It is not easy to solve these issues because I am not well acquainted with the platforms used to create the frontend and backend aspects. Even when I try to find the solution online, there are tons of different suggestions and I am not sure which is the best one. So, there are a lot of instances where I have to experiment with various suggestions and then see what is going on to truly understand how the platform works. Doing this takes a lot of time and hence I am not able to progress as fast as I hoped I could. However, this process is enabling me to truly understand how the platforms work and also understand the fundamentals behind frontend and backend development.

- The other problem I faced was whether to write a certain implementation myself or use a gem that is available. One such case is user authentication. I was trying to write the implementation for this from scratch but soon realized that there are a lot of aspects to take care of especially in terms of security. As I myself am unfamiliar with this, I went to research and found out that there was a gem called Devise which could be used. Hence, I went ahead to use that. During this process I had trouble making a decision as to whether I should use a gem or write the implementation myself. However, I then found out that using Devise did not work well with the implementation of tagging which I wrote. I was unable to display only the user’s tasks when I clicked the tag. It would show all tasks created in the database. Hence I am trying to figure out how to overcome this issue as of now.