CVWO Mid-Assignment Write-up

Name: ‘Izzah Bte Abdul Jalil
Matriculation Number: A0238063H

**Subject to changes*

Basic Use Cases

<u>User creates an account</u>

Description:

1. User clicks on “Sign Up” button
2. User enters a unique username into the “Username” field
3. User enters a password into the “Password” field
4. User enters the same password into the “Confirm Password” field
5. User clicks on “Sign Up” button
6. User account is created on successful sign up

<u>User logs in to account</u>

Description:

1. User clicks on “Log In” button
2. User enters username into the “Username” field
3. User enters password into the “Password” field
4. User clicks on “Log In” button
5. User is directed to home page on successful login

<u>User adds task to to-do list</u>

Description:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on “Create New Task”
3. User enters task description into the “Description” field
4. Optional steps:
    a. User enters task deadline into the “Deadline” field
    b. User enters time due into the “Time Due” field
    c. User clicks on dropdown menu in the “Priority” field and selects a priority level for the task
    d. User enters tags for the task in the “Tags” field
5. User clicks on “Create Task” button
6. Task is added to the task list on successful submission

<u>User views details of individual task</u>

Description:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on “View” button of the task
3. User is redirected to a page showing all the details of the task

User edits details of task

Description:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on the “Edit” button of the task
3. User updates the given fields accordingly
4. User clicks on the “Update” button

Alternative Flow:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on “View” button of the task
3. User is redirected to a page showing all the details of the task
4. User clicks on “Edit” button of the task
5. User updates the given fields accordingly
6. User clicks on the “Update” button

User deletes task

Description:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on the “Delete” button of the task
3. User is shown a confirmation message for deletion of task
4. User clicks on the “Yes” button
5. Task is deleted from the task list

Alternative Flow:

1. User clicks on “Task List” on the sidebar, which navigates to the home page
2. User clicks on “View” button of the task
3. User is redirected to a page showing all the details of the task
4. User clicks on the “Delete” button of the task
5. User is shown a confirmation message for deletion of task
6. User clicks on the “Yes” button
7. User is redirected to the home page
8. Task is deleted from the task list

User searches for task

Description:

1. User enters the description of the task into the search bar on the side bar
2. The task list will display a list of tasks containing the description entered in the search bar

User filters task list based on tags

Description:

1. User clicks on the “Filters” button on the side bar, which will display a dropdown list of all existing tags

2. User selects a tag
3. Task list is updated to only display tasks containing the selected tag

Current Progress

- Implemented a Rails backend with PostgreSQL database and a React front-end
- Currently working with JavaScript in React, but plan to convert to TypeScript
- Created basic CRUD functionality for the task model

Plan

Software-related

- PostgreSQL database
- Rails backend
- Bootstrap for styling purposes
- Migrate JavaScript React app to TypeScript after implementing all features
- Integrate Redux with React to handle state (if time permits)
- Containerize the app with Docker
- Deploy app with Heroku

Features to be implemented

- Add tag model
- Add user model
- Add user authentication and login/sign up page
- Use custom styling to override Bootstrap styles
- Implement search functionality
- Implement tagging system for tasks
- Implement sorting functionality → sort tasks by deadline or priority
- Implement filtering system to filter tasks by selected tags
- Add confirmation for deletion of tasks
- Add a tab to display all tasks due today