User Manual

This Todo Manager implements a basic to-do list per user.

1. Log in is required for usage. Apart from the log in and signup page, every other page will redirect you back to the log in page as they require a user session to continue. Please create a new user if you have not done so.
2. After log in, you will be redirected to /tasks, containing the main to-do list component. It is separated between tasks and completed.
3. Create a new task by clicking the 'New Task' button and entering an appropriate description/title for the task.
4. Check/uncheck the checkbox associated with each task to complete/uncomplete them. The database will be updated in real-time. In order to prevent unnecessary inconvenience from unintended clicks, the tasks will only be sorted into their appropriate tabs on refresh.
5. Click the 'Delete' button associated to a task to delete it for good. Note: Tasks cannot be recovered!
6. To edit the task/category of the task, or to add a new category to the task, click the 'Edit' button associated to a task. Categories are case-sensitive and each task now only supports one category. Submitting the edit form with a new category will override the existing category of the task.
7. In the main to-do list, the categories of respective tasks will be displayed on the far right of said tasks as a badge. Click the badge of a category to look up all tasks associated with a category.
8. Click the log out button to end your session.

Reflection

This CVWO assignment was a fruitful experience that allowed me to hone my competency in software development, as well as challenged my application design skills.

A major challenge for me was that, although I had a clear image on what I wanted the end-product to be, my competency level was far from adequate at the beginning of the assignment. I signed up for Udemy courses by using my eprep credits awarded by MINDEF, and had to learn ruby, javascript and react from scratch. At this point, it was clear that I will probably not be able to implement many of the features I have drafted, which was laid out in the mid-assignment submission.

Throughout the assignment, I realised that although many aspects of the project might seem clear, there should be purposeful, intended implementation behind each and every one of them. For example, users indicate the completion of a task by checking/unchecking the checkbox associated with the task. At first glance, I underestimated the care needed for me to implement this feature. What feedback, if any, should be given to the user to provide them confirmation that the database has been updated? Should a task that was just completed be immediately removed from the 'Tasks' tab and placed in the 'Completed' tab, or should it only be updated on refresh? How should the request be handled? Should the request be an ordinary http post request, or is there a reason for ajax to be used? In the end, I decided that the simple visual feedback of the checkbox checking/unchecking, and striking through the description of the task displayed on complete, and un-striking-through it on uncomplete would suffice. Flash notifications would be disruptive to the user experience and unnecessary from a user standpoint. Also, to reduce inconvenience from accidental clicks, the tasks will only be re-rendered under their appropriate tabs on refresh. Instead of an ordinary http post request, ajax should be used as there is no need to re-render the entire page after every task completion.

After exposing myself to many of such design issues, it reinforced that thorough thought is required to examine single component incorporated in the solution.

This assignment also exposed many gaps in my knowledge. For example, although I understand the many-many association between tasks and categories, I have yet to come up with a solution for users to dynamically add/delete category fields on the front-end, within the edit task form. Thus, the to-do list currently only supports one category per task, although a skeleton solution has been drafted in the controller on the backend. Although the assignment is over, my interest in developing my competency has only grew greater, and I am eager to further my learning to address these gaps in my knowledge.

All in all, I am grateful for this assignment. I have developed my competency in the various skills required by this assignment, and have been spurred to continue said development after my submission, and I hope to have many more other opportunities and challenges for me to partake in, and apply my learning.