Task Management System Final Report
CP4485
John-Michael Woodrow
John Cumby
Mohammad Aftab

For the final project in CP4485 we created a Task Management System with options to view, add, edit, delete tasks, teams and members which are stored in a database. In our initial proposal we stated that the project would be planned using the following technologies and features.

Proposed Technologies:

- React for interactive UI components.
- Node.js and Express.js as the backend framework.
- MongoDB for data storage.
- JavaScript, HTML, and CSS for development.
- Visual Studio Code and GitHub for code management.

We stayed mostly on track but through the process we decided a full fledged react application wasn't worth the time and effort when we could easily deliver the UI components with plain vanilla javascript. Although we didn't use react for the UI, we kept the same principles when developing the project throughout (modularity/organization of file structure).

Proposed Features:

- User Authentication: Register and login.
- Task Operations: Adding, editing, deletion, assignment, and prioritization.
- Task Views: Kanban board and list view.
- Notifications for deadlines and task assignments.
- Reporting: Analysis of task statuses.

We stayed mostly on track with the basic functions of the project, you can complete full CRUD operations using the application and seems to work for every scenario, unfortunately we settled on a simpler UI rather than a kanban type board and the notifications as well as user authentication were ruled redundant due to time and workload constraints.

For the backend we established the connections between the database and the application using the mongoose library which simplifies interactions with our mongodb database. The routes for handling data from the database were organized in separate files for better structure and maintainability. The express server was also established here with the express framework, a flexible Node.js web application framework that simplified the creation of our API endpoints. For the front end we decided to develop functions that typically handled a single task based upon user interaction with the application. For the UI elements we fed data from the database into html elements and styled them with CSS.

We ran into multiple small hiccups along the way with the code, there was a definite learning curve when it came to using mongodb so little things slipped through the cracks such as the wrong information in the connection string and common errors with html elements and other

code. Utilization of alerts and console logs helped a lot along the way but ultimately left the code a little messy and hard to clean up without missing some things.

Throughout this project we feel as though we have gained extensive knowledge with javascript, node, express, mongodb as well as general planning, structure and common practices in the development field. This project has contributed immensely to making us better developers all around through the adaptability and focus on core web development principals. Seeing a finished product that works as intended is what we should strive for as developers and we are greatly satisfied with what we have made with limited knowledge of the specific subjects going into the project.