# Project: Task Management Application with Trello-like Design

## Overview

The goal of this project is to develop a Trello-like Task Management Application using **React** for the front-end and **Express.js** for the back-end. Users can log in, create, manage, and track tasks. The application should support drag-and-drop functionality for task management and real-time updates using **WebSockets**. The project will use **PostgreSQL** as the database.

## Trello-like Task Board Features

- Task Columns: Organise tasks by their statuses by displaying them inside of according columns.
- Task Cards: Each task is displayed as a card containing information.
- Add New Task: Users can create new tasks by filling out the form with the necessary details.
- Filter and Search: Implement options to filter and search tasks by their details.

## Primary Use Cases Ranked by Priority (Highest to Lowest)

1. Register: Implement an option for new users to add a new account inside the system.
2. Login/Logout: Implement **JWT-based** authentication for secure user login and logout.
3. Create Task: Allow users to create tasks with the following details: **title, description, priority level, due date, and status.**
4. View Tasks: Display tasks on a Trello-like board, where each task is represented as a card inside of the following columns: **To Do, In Progress, QA, and Done.**
5. Update Task: Users should be able to update existing tasks by modifying their details.
6. Delete Task: Users can delete tasks they no longer need.
7. Filter and Search: Users can filter and search tasks by details contained on their cards.
8. Drag-and-Drop Functionality: Implement drag-and-drop functionality which allows users to change task statuses by moving them between columns.

9. Real-Time Updates via WebSockets: Ensure that task updates are reflected in real-time using WebSockets.
10. Track Task Progress: Implement a progress tracker to show what percentage of tasks that are completed.

## Bonus use cases

1. Task Notifications: Implement a system that notifies users about task updates or approaching due dates.
2. Calendar View: Add a calendar feature to visualize due dates.
3. Sprints: Enable an option that allows the user to kick off a new sprint which creates a new task board.
4. Premium users: Premium users don't have a limit on how many sprints can be created, while regular users can create a maximum of 3 sprints
5. Reset password: Users receive a link in their email, which redirects them to the app and allows them to reset their password.

## Technical Requirements

- Frontend: Use React for frontend. TypeScript must be used.
- Backend: Use Express.js for backend logic with JWT for authentication. TypeScript must be used.
- Database: PostgreSQL to store all data.
- Real-Time Updates: Implement WebSockets for real-time task updates.
- Deployment: Dockerize the application (optional).
- Nginx: Set up a reverse proxy in front of the application using Nginx (optional).

## Guidelines

- All primary use cases are **mandatory**.
- Bonus use cases have higher priority than optional technical requirements.
- Document your progress inside of the **README** file with implementation details and instructions on how to run the project.
- Create a GitHub repository for progress tracking and add the following accounts as collaborators: **ivan-at-2c, matija-at-2c, milan-at-2c and igor-at-2c**. You may also be asked to include additional collaborators to review your PRs based on their availability.

- All changes must be submitted using **Pull Requests (PRs)** and all collaborators should be added as reviewers. When you create new a PR, you notify collaborators about it via the **Slack channel** you have been added to.
- Write all code using the functional programming paradigm. Do not use classes in JavaScript code.

## General Internship advice

- Watch tutorials to familiarise yourself with new technologies.
- While going through tutorials it is recommended that you begin development of your application. This will help you in avoiding in spending too much time learning new technologies since the completion of the task is your main focus.
- It is recommended to begin your work on the task with the development of the back-end application.
- View mentors who are appointed to you as colleagues and keep in mind that your interactions with the team are important and will be considered during evaluations.
- Feedback on your PRs should be viewed as constructive suggestions that are meant to improve your work rather than criticism.

## Tutorials

React:

- Url: https://codewithmosh.com/
- Email: ivan.mihajlov2703@gmail.com
- Password: 2CSolution.1@
- If prompted for 2FA during login, contact Ivan to obtain the authentication code.

Express.js

- Url: https://www.udemy.com/course/nodejs-master-class
- Email: ivan.mihajlov2703@gmail.com
- If prompted for 2FA during login, contact Ivan to obtain the authentication code.