

Toka FSD Home Assignment

Overview

Develop a Real-Time Collaborative Project Management Application

Description

Build a project management application with real-time collaboration features. Users should be able to create, manage, and track projects and tasks, and see updates in real-time. The front-end will be built with Vue 2, the back-end with Express, and MongoDB for the database. Real-time communication will be implemented using WebSockets.

Plase use JavaScript, no TypeScript

Requirements

Front-End

Project and Task Management

- Users can create, view, edit, and delete projects.
- Each project can have multiple tasks. Users can create, view, edit, and delete tasks within a project.
- Changes to projects and tasks should be reflected in real-time across all connected clients.

User Interface

- Clean and intuitive UI.
- Use Vue components for reusability.
- Implement state management using Vuex.
- Implement routing using Vue Router.
- Display real-time updates using WebSockets.
- Do not use CSS frameworks for UI components
 - Only use responsive grid if needed (e.g bootstrap grid)
- Use Less / Sass for styling

Form Validation

- Client-side form validation for creating and editing projects/tasks.

Back-End

RESTful API

- CRUD operations for projects and tasks.
- Proper HTTP status codes for responses.
- Data validation and error handling.

WebSocket Implementation

- Set up WebSocket server using a library like **ws / express-ws** (no Socket.IO).
- Implement real-time communication for project and task updates.
- Ensure that when one user makes changes, all connected users see the updates instantly.

Database

- Design schemas for projects and tasks, considering relationships between them.
- Efficient data querying and storage.

Conciderations

Code Quality

- Clean, readable, and maintainable code.
- Proper use of comments and documentation.
- Consistent coding style.

Best Practices

- Follow best practices in front-end and back-end development.
- Use of modern JavaScript features.

Scalability and Performance

- Thoughtful consideration of scalability (e.g., handling a high number of concurrent WebSocket connections).
- Efficient data handling and querying.

User Experience

- Intuitive and user-friendly interface.
- Responsive design (optional, but a plus).

Submission Guidelines

Repository

- Create a GitHub repository for the project and commit code regularly.
- Include any assumptions made and potential areas of improvement.

Timeline

- The candidate should complete the assignment within one week.

Bonus Points

- Implementation of advanced features like search, filtering, or notifications.
- Responsive design.
- Comprehensive unit and integration tests.

- Real-time collaborative editing (e.g., multiple users can edit a task description simultaneously).
- Create Dockerfile & docker compose