lupiya
borrow.invest

Welcome to the Lupiya Coding Challenge!

Overview

To complete this challenge, you will need to write a simple web application.

The purpose of this challenge is to assess your skills and approach to composing a simple web app and CRUD capabilities given a set of screens. We'll also assess the generated HTML, CSS, and JS output.

This challenge should take about 1-2 hours.

Overview

This is a pretty straight-forward challenge. You'll build a task management app that allows users to create, view, update, and delete tasks. The application should consist of both a frontend and a backend, using either Angular or React for the frontend, and Node.js with MongoDB for the backend.

Frontend Requirements (Choose either Angular or React):

1. Create a user interface where users can:
    a. Add a new task with a title, description, and due date.
    b. View a list of all tasks with their titles and due dates.
    c. Click on a task to view its details and edit the title, description, or due date.
    d. Delete a task.
2. If using Angular: Use Angular to create components for task creation, task list, task details, and task editing.
    a. Implement routing to navigate between different views (e.g., a list of tasks and a task detail/edit page).
3. If using React: Use React to create components for task creation, task list, task details, and task editing.
    a. Manage state using React hooks or state management libraries (e.g., Redux).
4. Ensure a responsive and user-friendly design.

Backend Requirements:

1. Create a RESTful API using Node.js and Express.js to handle CRUD operations for tasks.

2. Define a MongoDB schema for tasks with fields for title, description, due date, and a unique identifier.
3. Implement endpoints for:
   a. Creating a new task
   b. Retrieving a list of tasks
   c. Retrieving a single task by ID
   d. Updating a task by ID
   e. Deleting a task by ID
4. Use appropriate error handling and validation for requests.

Additional Considerations (Optional):

1. Implement user authentication to allow each user to have an account and view only their tasks.
2. Add pagination or infinite scrolling for the task list if there are many tasks.
3. Include unit tests for both the frontend and backend components.
4. Implement middleware for logging API requests and errors.

GitHub Repository Instructions:

1. Create a new public GitHub repository by forking the Lupiya Coding Challenge repository.
2. Include a README.md file with the following information:
   - Instructions on how to set up and run the application locally.
   - Details about the technologies used.
   - Any design decisions you made.
   - Any additional features or improvements you've implemented.
   - Mention whether you chose Angular or React for the frontend.

Submission Guidelines:

- Once done with your changes, send the address of your fork to the following reviewers:
  - penjani.mkandawire@lupiya.com