Project Overview

Task Management System

This task management system will enable users to create, manage, and collaborate on tasks effectively. It will include essential features for user authentication, task management, collaboration, and user profiles, ensuring a comprehensive solution for users. Importantly, the application will operate without requiring a server API, with all data being stored locally using LocalStorage or IndexedDB.

Objectives

To create a Vite React application that serves as a platform for managing tasks.
To provide a user-friendly interface and functionalities that promote task organization and collaboration.

Task Breakdown

Setup and Initialization:

Initialize the project using Vite for fast development.
Implement TailwindCSS for streamlined and responsive styling.
Configure TanStack Router for routing to various application sections:

- Dashboard
- Task Management
- User Profiles
- User Authentication

User Authentication:

Implement a registration and login system for users.

Dashboard:

Create a dashboard that summarizes key task metrics:

- Display total tasks
- Show completed tasksa
- Highlight tasks that are due soon

Task Management Features:

Task Creation and Modification:

Develop forms to create and edit tasks with:

- Title
- Description
- Due date

- Priority level
- Tags

Task List View:

Present tasks in either a table or card format, with filters for status (all, active, completed) and priority.

Task Details:

Enable task selection for viewing details, editing, or deleting.

Subtasks:

Allow users to add and manage subtasks for better organization.

Collaboration Features:

Users can assign tasks to themselves or others.
Introduce a comment section for task discussions.
Implement notifications for task assignments and new comments.

User Profiles:

Establish user profiles displaying:
- Ongoing tasks
- Completed tasks
- Activity history

Allow users to edit profile information and settings.

Admin Area (Optional):

Consider adding an admin dashboard to oversee user statistics and task analytics.

Integration Options:

Explore APIs for calendar integration to manage due dates and reminders.

Design Guidelines:

Focus on a minimalistic design approach:
- Effective use of white space for task focus.
- Avoid clutter; emphasize essential information.
- Maintain a clear font hierarchy for improved readability.

Deliverables:

A fully functional task management application with robust user authentication features.
Complete source code with documentation detailing the architecture and tools used in development.
Prepopulated tasks within the application for demonstration purposes.

Additional Notes:

No Server API Required: The application will function entirely client-side without a backend server API.

Data Storage: All data will be managed locally using LocalStorage or IndexedDB, ensuring data persistence across sessions.