Project Overview

You will be building a "Personal Task Management System" that helps users organize their daily tasks and projects. This application will demonstrate your understanding of full-stack development using Next.js 15, PostgreSQL, Drizzle ORM, and other modern web technologies.

Technical Requirements

Backend Development

1. Database Design and Implementation
   - Set up a PostgreSQL database
   - Implement database schema using Drizzle ORM
   - Create necessary tables for users, tasks, projects, and categories

2. API Development (Next.js App Router)
   - Implement RESTful APIs for:
     - User authentication (sign up, login, logout)
     - CRUD operations for tasks and projects
     - Task categorization and filtering
   - Implement proper error handling and validation
   - Use middleware for authentication and request validation

Frontend Development

1. User Interface
   - Create a responsive dashboard layout
   - Implement a sidebar navigation system
   - Design forms for task and project creation/editing
   - Add a calendar view for task scheduling

2. State Management and Data Fetching
   - Implement global state management using Zustand
   - Use React Query for API data fetching and caching
   - Handle loading and error states appropriately
   - Implement optimistic updates for better user experience

Required Features:

1. Landing Page
   - Should have good UI/UX
   - Should have at least 5 sections: Header, Hero, Features, CTA and Footer
   - Should clearly outline the offerings of the product

2. Authentication System
   - User registration and login
   - Protected routes and API endpoints

3. Task Management
   - Create, read, update, and delete tasks
   - Assign priority levels and due dates
   - Categorize tasks into projects

4. Dashboard
   - Display task statistics and upcoming deadlines
   - Show task completion progress
   - Implement task filtering and search
   - Add a calendar widget which shows tasks inside a calendar view

Technical Stack Requirements:

- Next.js 15 with App Router
- PostgreSQL
- Drizzle ORM
- React Query (TanStack Query)
- Zustand
- TypeScript
- Tailwind CSS (recommended for styling)

Evaluation Criteria

We will assess your submission based on:

1. Code organization and architecture
2. UI/UX - Overall Design
2. Proper use of TypeScript and type safety
3. Implementation of modern React patterns and hooks

4. Database schema design and query optimization
5. API design and implementation
6. State management approach
7. Error handling and input validation

Bonus Points

- Implementing dark mode support
- Creating a mobile-responsive design