

Product Requirements Document (PRD)

Project: Task Management Web Application (Take-Home Assignment)

Objective: Deliver a clean, usable, and production-ready task management application demonstrating strong engineering fundamentals, UX thinking, and disciplined execution.

1. Goals and Success Criteria

- 1 Support full CRUD operations on tasks
- 2 Provide clear task filtering by status
- 3 Maintain a simple and intuitive user experience
- 4 Demonstrate clean architecture and modular components
- 5 Be deployed publicly with a live URL
- 6 Use incremental Git commits reflecting development phases

2. Technology Stack

Frontend: React 18, Vite, TypeScript

Styling: CSS or CSS Modules

State Management: React useState and useEffect

Deployment: Vercel

3. Application Architecture

- 1 UI components handle rendering and interactions only
- 2 Mock API layer simulates backend operations asynchronously
- 3 Centralized TypeScript types ensure consistency

4. Folder Structure

src/api, src/components, src/types, src/styles, App.tsx, main.tsx

5. Data Model

- 1 id: unique identifier
- 2 title: task title
- 3 completed: task status
- 4 createdAt: ISO timestamp

6. Functional Requirements

- 1 Create tasks

- 2 View task list
- 3 Edit task titles
- 4 Toggle task completion
- 5 Delete tasks with confirmation
- 6 Filter tasks by status

7. User Experience Requirements

- 1 Disable empty submissions
- 2 Display empty states
- 3 Provide immediate feedback

8. Mock Backend

Mock API functions should return Promises and optionally simulate latency.

9. Git Commit Strategy

- 1 Project initialization
- 2 Data model and mock API
- 3 UI components
- 4 CRUD functionality
- 5 UX polish
- 6 Deployment and documentation

10. Deployment

Deploy using Vercel and include the live URL in the README.

11. Assumptions and Trade-offs

Backend services and authentication are intentionally excluded to maintain scope focus.