Task Manager using OOP and File Management

Project Proposal

Problem Description

In many academic and professional environments, individuals work primarily in command-line interfaces, especially in software development and system administration roles. While numerous task management applications exist, most are web-based, over-engineered for simple needs, or require user registration and internet connectivity. This creates a barrier for users who seek a lightweight, offline, and easy-to-use task tracking solution within the terminal.

There is a clear gap for a tool that:

- Works entirely offline.
- Is simple, fast, and focused on essential functionality.
- Stores task data persistently.
- Follows good software engineering practices, especially Object-Oriented Programming (OOP).

Proposed Solution

We propose to develop a **command-line task manager application** using **Node.js** and **OOP principles**. This application will allow users to:

- Add new tasks with relevant details.
- View all existing tasks.
- Update tasks (e.g., mark as completed, change title/description).

- Delete tasks.
- Persist all task data using the local file system (JSON).

Key components of the system:

- **Task class** to encapsulate task data.
- **TaskManager class** to manage the list of tasks and business logic.
- **FileService module** to handle file read/write operations.
- A clean and intuitive **CLI interface** for user interaction.

The entire application will follow OOP design principles (encapsulation, abstraction, modularity) and use JSON files to simulate a database.

Technology Stack

Technology	Purpose
Node.js	Server-side runtime environment for CLI logic
JavaScript (ES6+)	Programming language
Node.js `fs` module	For reading/writing tasks to the filesystem
Git & GitHub	Version control and collaboration
Draw.io / Lucidchart	UML diagramming
Google Docs / Word	Report documentation
Terminal / Command Prompt	Platform to run the CLI app

Project Timeline and Milestones

Timeline	Activities / Deliverables
Day 1	Define scope, create project repo, assign team roles
	Design class structure, write project proposal
	Build `Task` and `TaskManager` classes
	Implement file operations using `fs`, test saving/loading
	Finalize CLI features (add, update, delete)
Day 2	Complete documentation and UML diagrams
Day 3	Record video walkthrough, finalize report and contribution logs
	Submission and final review