Project Phase 4 — Final Submission

Introduction (Rewritten & Improved)

During Phase 4 of CollabTrack, our team focused on completing the implementation of all high-priority features while refining key design elements to deliver a smoother and more intuitive user experience. Although the core architecture remained consistent with our original specification, this phase required several targeted updates across both the interface and backend structure. These enhancements were driven by usability considerations and the need for a clean, maintainable codebase as the project transitioned from planned design into full implementation.

A major improvement in this phase was the redesign of the home page into a centralized user dashboard. Previously, important features were scattered across multiple pages, requiring users to navigate through unnecessary views. By consolidating account details, project access, and activity summaries into one dashboard, we significantly streamlined navigation. As part of this update, the "SETTINGS" and "ACCOUNT" links were removed from the global navigation bar; both functions are now integrated directly into the dashboard. This change aligns CollabTrack with modern productivity platforms that emphasize direct access, simplicity, and reduced cognitive load.

On the backend, we strengthened the application's internal structure by introducing modular ES6-based utility files. The most notable addition is the `sqlTools.mjs` module, which centralizes all SQL operations—including SELECT, INSERT, UPDATE, and DELETE queries—into a single, reusable component. By grouping related functions in one module and exposing them through shared imports, we eliminated

duplicated code, improved readability, and made the backend easier to update or extend in future phases.

Throughout development, we also relied on supported npm modules and front-end frameworks to ensure efficient iteration and stable implementations. Collectively, these improvements resulted in a more organized, maintainable, and scalable application architecture—one that positions CollabTrack for future feature expansion without requiring disruptive redesign.

Final Submission — System Overview (1–2 Pages)

System Overview

CollabTrack is a lightweight, full-stack project and task management platform built on a modular Node.js + Express architecture. The system cleanly separates responsibilities across routing logic, dynamic page rendering, database utilities, and user-facing interface components. EJS templates generate dynamic HTML based on user and project data, while SQLite serves as the primary local data store, accessed through custom ES6 modules.

The platform is structured into three major layers:

1. Backend Logic

- Express routes handle user authentication, dashboard rendering, project navigation, task creation, commenting, and activity

updates.

- Controllers coordinate interactions between routes, templates, and the database.

- Modular ES6 utilities standardize database operations and reduce repeated logic.

2. Frontend Presentation Layer

- EJS templates generate dynamic HTML pages for dashboards, project views, task lists, and activity logs.

- Bootstrap and W3.CSS provide responsive layouts, form styles, navigation components, and UI consistency.

3. Database Layer

- A SQLite relational database stores users, projects, tasks, comments, and activity logs.

- Queries are executed through centralized helper functions within `sqlTools.mjs`.

Together, these layers form a cohesive full-stack system capable of supporting individual and team-based project collaboration, detailed task tracking, and activity logging.

Backend Modules and Tools

Express.js

Express forms the foundation of the backend, handling HTTP requests, routing, middleware logic, and controller communication. It supports all primary features, including login flows, dashboard rendering, project views, and CRUD operations for tasks and comments.

dotenv

dotenv loads configuration variables from a `.env` file, preventing sensitive values—such as database file paths or API keys—from being hard-coded in the repository.

sqlite3

The SQLite3 module manages all persistent data storage. It executes queries to create tables, retrieve records, update tasks, authenticate users, and maintain project information.

nodemon

Used during development to automatically restart the server when changes are detected, greatly improving workflow efficiency.

EJS (Embedded JavaScript Templates)

EJS is used to dynamically generate HTML content based on server-side data. It renders personalized dashboards, project spaces, task lists, and comment threads.

Custom ES6 Modules

resources/sqlTools.mjs

This module centralizes all SQL-related logic, including:

- Running SELECT, INSERT, UPDATE, and DELETE queries
- Managing database connections
- Exporting reusable CRUD helper functions
- Reducing duplication across route files

By consolidating database access into one module, the backend gained better structure, readability, and maintainability.

Frontend Structure and Tools

Bootstrap

Bootstrap accelerates UI development through pre-built components such as navigation bars, cards, modals, alerts, and responsive form layouts.

W3.CSS

W3.CSS provides lightweight layout utilities for spacing, alignment, and structural organization across pages. It complements Bootstrap without adding heavy styling overhead.

Conclusion of Overview

The current implementation of CollabTrack successfully integrates all high-priority features while maintaining a clean, modular, and scalable architecture. Express routing, EJS-based presentation, ES6 module organization, and SQLite storage form a stable foundation suitable for ongoing development. This structure supports future enhancements such as notifications, analytics, team collaboration metrics, or cloud-based deployment with minimal architectural changes.