

Project Summary Report: TeamTasker

TeamTasker – A Full Stack Role-Based Task Management System

Live Demo: <https://teamtasker.raguladhithya.in>

Submitted by: Ragul Adhithya
Internship Project – Lark Labs AI
Date: 12 June 2025

Project Overview:

TeamTasker is a full-fledged task and project management system designed to meet the demands of real-world enterprise environments. It comes with **user authentication, role-based access control**, and **streamlined project and task assignment workflows**. You'll also find powerful **analytics** and a **clean, modern user interface**. The application acts like a production-grade SaaS dashboard, enabling teams to collaborate effectively through structured task flows, engage in **real-time commenting**, and visually monitor project progress.

What I Built:

The application consists of a secure backend, an interactive frontend, and a PostgreSQL + Redis-based data layer.

Frontend (React + Vite + Tailwind + Shadcn):

- Responsive, role-aware UI with dashboard, side navigation, and card-based analytics
- Full CRUD interface for users, projects, and tasks with permissions
- Task creation form with dynamic dropdowns for users and projects
- Toast and confirmation feedback using Shadcn toast and alert-dialog
- View all tasks table with deadline indicators and View/Edit/Delete buttons
- Single Task Page with real-time commenting and role-aware comment deletion








Backend (Node.js + Express + TypeScript + Sequelize):

- JWT-based token authentication with secure route protection
- Role-based permissions (admin, manager, developer, tester)
- Routes to create, edit, delete users, projects, and tasks
- Middleware to restrict managers to only create tasks in assigned projects
- Real-time comment handling with Socket.IO and Redis
- Sequelize models with associations between users, tasks, and projects

Deployment:

- Frontend deployed via Cloudflare Pages
- Backend deployed on Railway (Node + Express server)
- PostgreSQL database hosted on Railway
- Redis for real-time comment messaging hosted separately on Railway

Key Features:

-  JWT authentication with secure token storage and context state management
-  Role-based dashboards (Admin, Manager, Developer, Tester)
-  Project-specific task creation logic tied to assigned managers
-  Comment section on each task with live updates via Socket.IO
-  Analytics on users, projects, tasks shown on dashboard
-  Deadline indicators with color-coded labels (due soon, overdue)
-  Smooth feedback using toast + modal confirmation dialogs (shadcn)

Challenges Faced & How I Solved Them:

a) Role-Based UI Rendering:

Challenge: Building dynamic sidebars, dashboards, and route access based on user roles.

Solution: Used role arrays and centralized filtering logic inside the navigation component and routes.

b) Project-Scoped Task Restrictions:

Challenge: Managers must only assign tasks to projects they're managing.

Solution: Added a many-to-many association between projects and managers using Sequelize, and enforced this rule in middleware.

c) Real-Time Comments & Redis Integration:

Challenge: Ensuring multiple users can comment on the same task and see updates in real time.

Solution: Used Socket.IO with Redis to manage real-time channels. Frontend listens for incoming messages and renders them live.

d) Deployment Errors & CORS Management:

Challenge: Connecting frontend (Cloudflare) with backend (Railway) caused CORS issues.

Solution: Added CORS middleware in backend with explicit origin allowlist including local dev and Cloudflare domain.

Final Outcome:

TeamTasker is now a live, production-ready full-stack application. It features modern architecture, clean code structure, and practical usability for managing project workflows in teams.

This project gave me hands-on experience in building a professional-grade web application from the ground up — covering everything from schema design to REST APIs, front-end state management, authentication, and real-time data pipelines.