

Agile Development Plan: Task Management System

Document Version: 1.0

Date: 15 March 2025

Author: Anshika Gupta

Phase 0: Discovery & Planning (Day 1–3)

Goal:

Understand business needs, finalize scope, and define architecture & setup.

Activities:

| S.N o | Task | Type | Status | Assignee | Estimate |
|----------|---|----------|--------|----------|----------|
| 1 | Deep dive into requirements | Planning | ✓ Done | Anshika | NA |
| 2 | Document assumptions (e.g., enums for roles, holidays logic) <i>Shared a separate requirement file</i> | Planning | ✓ Done | Anshika | NA |
| 3 | Draft wireframes for key screens | Frontend | To do | Anshika | NA |
| 4 | Define JSON contracts for tasks <i>Shared in a json_contract.txt file</i> | Backend | ✓ Done | Anshika | NA |
| 5 | Plan tech stack (Angular 19 + MobX, Node.js/Express, PostgreSQL) | Planning | ✓ Done | Anshika | NA |
| 6 | Create Sprint Planning Board | Planning | ✓ Done | Anshika | NA |

Outputs:

- Wireframes (Landing, Task List, Task Details, Create/Edit)
 - Final JSON Contracts (OpenAPI/Swagger)
 - Assumption Doc
 - Tech Stack Selection
 - Sprint Board Setup (tabular format below)
-

Phase 1: Project Setup (Day 3–5)

 **Goal:** Set up frontend, backend, database, state management, and dev workflows.

Tasks:

| S.No | Task | Type | Status | Assignee | Estimate |
|------|---|----------|---------------|----------|----------|
| 1 | Initialize Node.js project with Express | Backend | ✓ Done | Anshika | NA |
| 2 | Set up Swagger/OpenAPI | Backend | ✓ Done | Anshika | NA |
| 3 | Setup environment configs | Backend | ✓ Done | Anshika | NA |
| 4 | Design schemas for tasks/public_holidays/users/task_history | Database | ✓ Done | Anshika | NA |
| 5 | Add indexes (dueDate, priority, status) | Database | ✓ Done | Anshika | NA |
| 6 | Initialize Angular 19 project | Frontend | ✓ Done | Anshika | NA |
| 7 | Install Carbon Design System | Frontend | ✓ Done | Anshika | NA |
| 8 | Setup MobX stores (taskStore, userStore) | Frontend | ✓ Done | Anshika | NA |
| 9 | Add routing and base layout | Frontend | ✓ Done | Anshika | NA |
| 10 | Git setup (main/dev branches) | DevOps | ✓ Done | Anshika | NA |
| 11 | Docker setup (frontend/backend containers) | DevOps | ✓ Done | Anshika | NA |
| 12 | CI/CD pipeline (circle ci) | DevOps | 🕒 In Progress | Anshika | NA |

Deliverables:

- Fully initialized full-stack repo with README
- Swagger setup with base API spec
- Dockerized dev environment
- Basic DB schema with migration
- Circle ci implemented can explore more alternative for CICD

Sprint 1: Core Task CRUD (Day 4–6)

Goal:

Implement full task lifecycle management (CRUD) and validations.

User Stories:

| S.No | Task | Type | Status | Assignee | Estimate |
|------|--|----------|--------|----------|----------|
| 1 | Create CRUD APIs for tasks | Backend | ✓ Done | Anshika | NA |
| 2 | Implement validation (title, priority, due date not in the past) | Backend | ✓ Done | Anshika | NA |
| 3 | Create auth/role middleware | Backend | ✓ Done | Anshika | NA |
| 4 | Add public holiday validation | Backend | ✓ Done | Anshika | NA |
| 5 | Seed public holidays data | Database | ✓ Done | Anshika | NA |
| 6 | Add enum validations for status & priority | Database | ✓ Done | Anshika | NA |
| 7 | Task List & Task Details UI (Carbon Design) | Frontend | ✓ Done | Anshika | NA |
| 8 | Create/Edit Task Form | Frontend | ✓ Done | Anshika | NA |
| 9 | Integrate with backend using HTTPClient | Frontend | ✓ Done | Anshika | NA |
| 10 | MobX state integration | Frontend | ✓ Done | Anshika | NA |
| 11 | Jest for APIs (Create, Read, Update, Delete) | Testing | ✓ Done | Anshika | NA |
| 12 | Jasmine tests for task components | Testing | ✓ Done | Anshika | NA |
| 13 | Insomnia tests for workflows | Testing | ✓ Done | Anshika | NA |

Deliverables:

- Functional CRUD for tasks

- UI for task creation, update, listing, and detail view
 - Role-based delete API for admin
 - Tests
-

Sprint 2: Filtering & Rules (Day 6–9)

Goal:

Implement filters, public holiday/weekend restrictions, and RBAC.

User Stories:

| S.No | Task | Type | Status | Assignee | Estimate |
|------|--|----------|--------|----------|----------|
| 1 | Add filtering endpoints (query by priority, dueDate) | Backend | ✓ Done | Anshika | NA |
| 2 | Restrict task creation based on public holidays/weekends | Backend | ✓ Done | Anshika | NA |
| 3 | RBAC for deletion (admin only) | Backend | ✓ Done | Anshika | NA |
| 4 | Add filter controls (priority, due date) | Frontend | ✓ Done | Anshika | NA |
| 5 | UI error handling for invalid task creation | Frontend | ✓ Done | Anshika | NA |
| 6 | Admin-only delete button | Frontend | ✓ Done | Anshika | NA |
| 7 | Tests | Testing | ✓ Done | Anshika | NA |

Deliverables:

- Functional filters in UI

- Public holiday/weekend logic
- Admin-only delete option
- Complete test coverage

Sprint 3: Undo, Sharing & Role UI (Day 10)

Goal:

Implement undo, enhanced role-based UI create login and register user.

User Stories:

| S.N o | Task | Type | Status | Assignee | Estimate |
|----------|---|----------|--------|----------|----------|
| 1 | Store task history for undo | Backend | ✓ Done | Anshika | NA |
| 3 | Add role metadata in user profile | Backend | ✓ Done | Anshika | NA |
| 4 | Undo button (use MobX state snapshot) | Frontend | ✓ Done | Anshika | NA |
| 6 | Show/hide elements based on role | Frontend | ✓ Done | Anshika | NA |
| 7 | Implement login with JWT auth | Backend | ✓ Done | Anshika | NA |
| 8 | Implement Register user | Backend | ✓ Done | Anshika | NA |
| 9 | Responsive UI for login | Frontend | ✓ Done | Anshika | NA |
| 10 | Responsive UI for Register | Frontend | ✓ Done | Anshika | NA |
| 11 | cron job to delete old task history records | Backend | To Do | Anshika | NA |

Deliverables:










- Undo functionality
- Polished role-based UI
- Responsive login and register screens

Sprint 4: Finalization & Deployment (Day 10)

Goal:

Finalize project, polish UI, refactor code, and deploy.

Tasks:

| S.No | Task | Type | Status | Assignee | Estimate |
|------|--|------------------|---|----------|----------|
| 1 | Code refactoring | Backend/Frontend |  In Progress | Anshika | NA |
| 2 | Linting | Backend/Frontend |  Done | | |
| 3 | Add security headers and input sanitation | Backend |  In Progress | Anshika | NA |
| 4 | Final UI/UX polish (validation errors, empty states) | Frontend |  In Progress | Anshika | NA |
| 5 | Write Setup & Deployment Guide | Documentation |  Done | Anshika | NA |
| 6 | Write API Docs (Swagger) | Documentation |  In Progress | Anshika | NA |
| 7 | Write Undo Logic documentation | Documentation |  Done | Anshika | NA |
| 8 | Document Design Decisions | Documentation |  Done | Anshika | NA |
| 9 | Set up GitHub Actions for CI/CD deployment | DevOps |  Done | Anshika | NA |

Deliverables:

- Deployed app (Dockerized or cloud hosted)

- Complete project documentation
- Final test coverage report
- README with instructions

Color-Coded Gantt Chart by Sprint

