**Finalized PRD (MVP)**

**1) Overview & Goals**

Build a lightweight task-management web app where a **Lead/admin** creates and assigns tasks to **Members**. Tasks move through the lifecycle: **backlogged → defined → in-progress → review → completed**. Deadlines are color-coded (green/yellow/red). Members can only move tasks along the specific transitions you defined. Member accounts are invite-only; Lead/Admin accounts require email two-step verification (2SV) at creation.

**MVP goals**

* Core flows working end-to-end: invite → signup → task assignment → status changes → comments.
* Deadline highlighting + basic notifications.
* Secure account creation for Leads (2SV) and invite-only Members.
* Minimal, easily extendable data model and APIs so you can start coding quickly.

**Success metrics (suggested)**

* Time to create & assign a task (target < 180s).
* % of assigned tasks that reach “completed”.
* Invite acceptance rate.
* Weekly active users (team-level adoption).

**2) Task Lifecycle**

* States: **backlogged → defined → in-progress → review → completed**
* **Transition rules**
  + Lead:
    - can move between any states
    - only Lead can move **review → completed**
    - can reassign tasks
  + Member:
    - can only move **defined → in-progress** and **in-progress → review**
* Completed tasks: **soft delete** (archived but not fully removed).

**3) Task Deadlines & Highlighting**

* **Date only** (no time granularity).
* Display in user’s **local timezone**.
* Colors:
  + **Green:** due date > 48h away
  + **Yellow:** due date within 48h
  + **Red:** past due

**4) Assignees & Attachments**

* Tasks can have **single or multiple assignees** (not mandatory).
* Attachments allowed but **not required**.
* Max size (recommend: 10MB per attachment, limit 3 per task for MVP).

**5) Roles & Accounts**

* **Lead/Admin**:
  + Sign-up requires **one-time 6-digit email verification**.
  + Can create other Leads/Admins under them (like org hierarchy).
  + Can invite Members.
  + Can export task list to CSV.
* **Member**:
  + Account only via Lead invite.
  + Limited state changes + comments.

**6) Notifications**

* **In-app only** for MVP (badge + dashboard alerts).
* No email/push/Slack initially.

**7) Task Grouping**

* Tasks grouped by **project/tags**.
* Filtering by project/tag available on dashboard.

**8) Export**

* Leads/Admins can **export task list in CSV** (all tasks or filtered set).

**9) UI Scope**

* **List-only view** for MVP (no Kanban board).
* Task list should support:
  + Filter by project/tag, state, assignee
  + Sort by due date, creation date
  + Highlighted due-date colors

**10) Data Model Updates**

**User**

* role: lead | member
* parent\_org\_id (for org hierarchy when multiple leads exist)

**Task**

* assignees: array of user\_ids
* attachments: array of file refs
* project\_tag: string (or multiple tags)
* archived\_at (for soft delete)

**11) Acceptance Criteria (locked in)**

* Lead can create a task (title, description, assignee(s), due date, optional project/tag, optional attachments).
* Members can only change task status along allowed transitions.
* Lead only can move review → completed.
* Tasks show **red/yellow/green** deadline color in local timezone.
* Attachments can be added but not required.
* Leads can invite Members and create other Leads within org.
* Leads can reassign tasks at any time.
* Leads can export task list as CSV.
* Completed tasks soft deleted (archived, visible in export).
* All accounts created with **6-digit email verification**.

**12) Implementation Notes –**

**1) Angular Frontend (MVP scope)**

* **Pages/Components**
  + LoginComponent → email + password
  + DashboardComponent → list of tasks, filters (state, project/tag, assignee)
  + TaskListComponent → table view, colored due dates (red/yellow/green)
  + TaskDetailComponent → description, attachments, comments, status-change buttons
  + InviteComponent → Lead invites Members by email
  + ExportComponent → CSV export (Lead only)
* **Services**
  + AuthService (JWT or session token handling)
  + TaskService (CRUD + status transitions)
  + CommentService
  + InviteService

**2) Middleware (Node.js / Express)**

* **Responsibilities**
  + Implements API endpoints (REST/JSON)
  + Validates transitions (business logic: who can move which status)
  + Handles invites, 6-digit OTP email flow, file uploads
  + Generates CSV export for Leads
* **Libraries**
  + express for REST
  + jsonwebtoken for auth
  + bcrypt for password hashing
  + nodemailer or service (SendGrid/Mailgun) for OTP emails
  + multer (for handling attachments uploads)
  + fast-csv or json2csv for export

**3) Database Schema (MySQL data)**

* + Database schema to be designed for MySQL database.