**📋 Task: Build a Task-Management Dashboard**

**🏆 Overview**

You’ll build a small, full-stack Task Management Dashboard that allows users to create, view, edit, and organize tasks. This exercise will test your skills in:

* Front-end development with Next.js & Tailwind CSS
* Back-end API design with Express & MongoDB
* State management & data fetching
* UI/UX considerations (responsive design, modals, forms)
* Add a Kanban boards
* (Bonus) Advanced features like theming

**🚀 Requirements**

**1. Front-End (Next.js + Tailwind)**

* **Pages**
  + **All Tasks**: List all tasks in a responsive card grid.
  + **Create Task**: Modal form (or dedicated page) to add a new task.
  + **Task Details**: Show full details; allow in-place editing.
* **Styling**
  + Use **Tailwind CSS** for a clean, responsive design.
  + Cards should adapt to mobile/tablet/desktop breakpoints.
  + Modal overlay for “Create” and “Edit” forms.
* **Data Fetching**
  + You can choose **getServerSideProps**, **getStaticProps + client fetch**, or **SWR**/React Query for dynamic data.
* **Client-side form validation**
  + Ensure required fields (e.g. title, due date) are filled before submission.

**2. Back-End (Express + MongoDB)**

* **API Routes**
  + GET /api/tasks → list all tasks
  + GET /api/tasks/:id → get one task
  + POST /api/tasks → create task
  + PUT /api/tasks/:id → update task
  + DELETE /api/tasks/:id → delete task
* **Data Model**
* {
* title: String,
* description: String,
* status: String, // e.g. "To Do" | "In Progress" | "Done"
* dueDate: Date,
* customFields: Object, // for bonus requirement #1
* createdAt: Date,
* updatedAt: Date
* }
* **Database**
  + Use **MongoDB** (local or cloud).
  + Apply basic error-handling & status codes.

**➕ Extras (Nice-to-have)**

1. **Status Filter**
   * Dropdown or tabs to filter by status: “To Do,” “In Progress,” “Done.”
2. **Kanban View**
   * Display tasks in columns by status.
   * **Allow user to define new columns** (“custom statuses”) on the fly.
3. **Custom Color Theme**
   * Let users pick or switch between color palettes (light/dark, or custom).

**📂 Submission**

* Provide a GitHub repo link with two folders:
  + /frontend (Next.js app)
  + /backend (Express API)
* Include a README with:
  + Setup instructions
  + How you structured your code
  + Which extras you implemented
* Please submit your completed work by **End of Day (EOD) Wednesday, Jun 4, 2025**.

Feel free to use any additional libraries (e.g. Zustand/Redux, Mongoose, Chakra UI, React DnD for Kanban), but be prepared to explain your choices.

Good luck! We look forward to seeing your implementation.