Objective

This challenge is an application that includes all the below features based on the NodeJS framework NestJS, and one of the common frontend frameworks like NextJS or React.

The aim of this challenge is to create a simple task management system with time tracking to help users monitor their progress and productivity.

Backend Requirements

1. Secure Authentication APIs (register users - login users).
2. A task management system where users can create, edit, and delete tasks (CRUD).
3. Each task should consist of (id, title, status, time spent on the task, and user who created the task).
4. Users should have access to perform CRUD operations on their tasks only.
5. User should search their tasks by title.
6. A time tracking (Timer) system where users can track the time they spend on each task.
7. Tracking tasks should be using a clock-in and clock-out mechanism where the time spent on each task is calculated by the (clock-out timestamp - clock-in timestamp), but feel free to adopt other methodologies.
8. Return time spent on the development for each task.

Use the following technologies:

- NestJS for the backend
- Choose a database of your choice (MongoDB, MySQL, ..etc)

Frontend

1. Login and Registration Pages.
2. After users log in/ registration flow, users should be redirected to the homepage.
3. The homepage should be protected for logged-in users only.
4. Users should view their tasks on the homepage.
5. Users should perform the task CRUD operations from the homepage.
6. The homepage should include a search input at the top, allowing users to search for tasks.
7. Users should track only one task at a time
8. While a task is being tracked, the time counter should be updated in the format of (HH:MM:SS) to reflect the time elapsed.9. A time tracking system that integrates with the backend time tracking system.

Use the following technologies:

- React/NextJS, .
- TailwindCSS or Bootstrap Framework for UI.

Here are some additional tasks that you can consider adding to your project:

- Implement a filter feature so that users can filter tasks by status, priority, or due date.

Deliverables

- The project should be deployed on any hosting service like AWS, GCP, Heroku .. etc, Or Dockerized and runnable with docker-compose.