Trackify: Personal Task & Habit Tracker

Objectives

The goal of this project was to build a full-stack web application that helps users manage their daily tasks and habits more effectively. The app allows users to add tasks with deadlines, create daily habits, receive automated reminders, and keep track of their overall progress.

Concept

I wanted to create something that supports productivity and consistency. With Trackify, users can:

- Create an account and log in securely
- Add, edit, and delete tasks and habits
- Mark tasks and habits as completed
- Get email reminders 1 hour before due times
- Toggle reminder settings anytime

The app is focused on helping users build good routines and never miss important deadlines.

Implementation

I began by building a basic dashboard skeleton and connected it to backend routes. In the first iteration, I focused on functionality — getting registration, login, task and habit creation, and completion to work dynamically. Once the core features were functional, I moved on to refining the design, adding interactivity on the frontend, and implementing extras like email reminders and user preference toggles.

Design Process

I started with a rough sketch of what I wanted the dashboard to look like. Since I wanted something simple and not overwhelming, I kept the layout clean — focusing on clarity and responsiveness.

I used static HTML and CSS for UI components, keeping animations and styles subtle but functional. The design allows users to quickly log in, add tasks/habits, and view their progress at a glance.

Backend Development

The backend was built with Express and MongoDB. I used **Mongoose** to manage schemas for users, tasks, and habits. I also implemented secure user authentication using **bcryptjs** to hash passwords.

One interesting challenge was the email reminder system. I used **node-cron** to schedule jobs and **nodemailer** to send emails. I also made sure completed tasks/habits don't trigger reminders again.

Testing & Debugging

I tested the app manually by creating multiple user accounts, adding tasks/habits with different times, and checking how reminders worked across edge cases. I also tested login/logout flows, email toggling, and account deletion. Any issues I found — like reminders being sent even after completion — were debugged and fixed carefully.

Documentation & GitHub

All code was pushed to a public GitHub repo. I added comments throughout the codebase, included setup instructions in the README, and wrote out test cases clearly. Screenshots and a short video demo were also included.

Lessons Learned

This project reminded me how effective it is to get a working version up first and then iterate. Planning features and breaking them into smaller blocks saved a lot of debugging time. I also learned how to better manage backend logic for reminders and user preferences.

Working with node-cron and nodemailer was a new experience, and now I feel much more confident managing real-world scheduling and email handling.

Reflections & Future Improvements

The final product successfully met the goal of a working productivity app that solves a real need. However, there are several potential improvements that could enhance the application such as:

- Adding user-friendly error messages and loading spinners
- Creating better UI animations
- Integrating Google Calendar for syncing tasks and habit schedules.
- Adding support for custom habit frequencies