MediTrack

Presented by Helly Niteshbhai Diyora & Lili Mei Ye

Northeastern University

Table of Contents.

01

Project Objective

MediTrack helps individuals and families manage daily medications and prescription refills.

- Track medications with dosage & schedule
- Mark medications as taken to avoid missed or duplicate doses
- Log and review prescription refill history
- Designed for caregivers, elderly patients, and those with chronic conditions
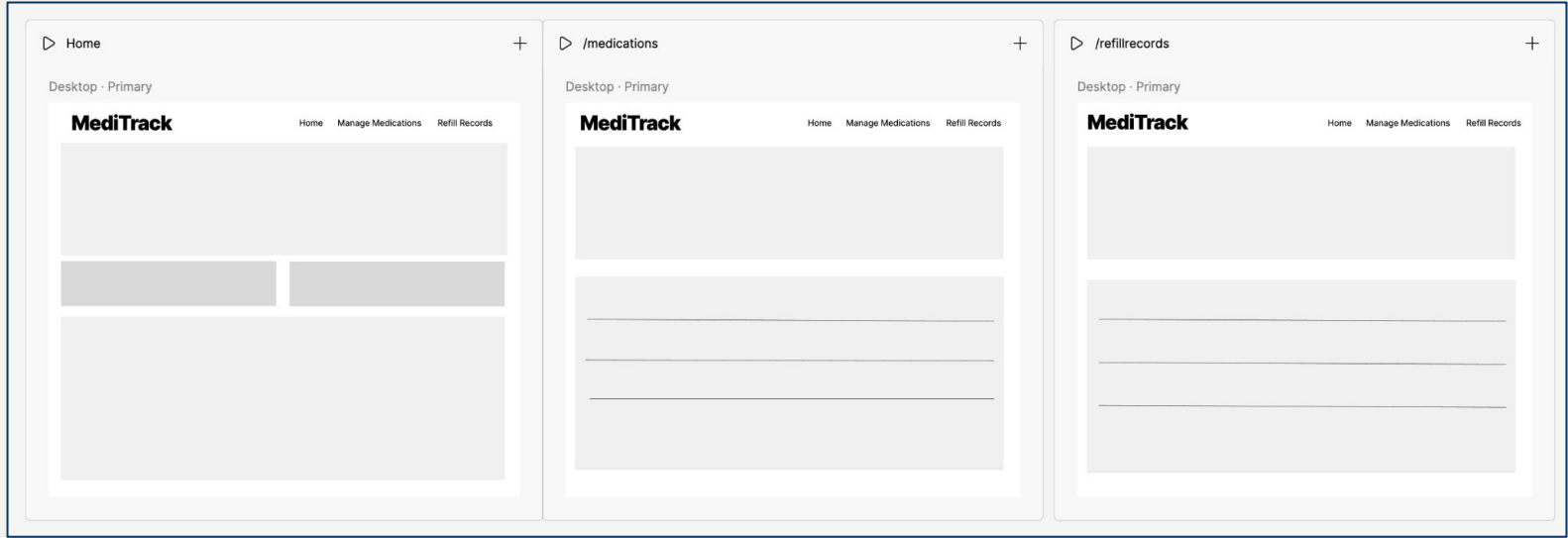
02

Technology Used

- **Frontend**: HTML5, CSS3, Vanilla JavaScript (client-side rendering)
- **Backend**: Node.js + Express v5
- **Database**: MongoDB (native driver, no Mongoose)
- **Tools**: ESLint + Prettier, dotenv
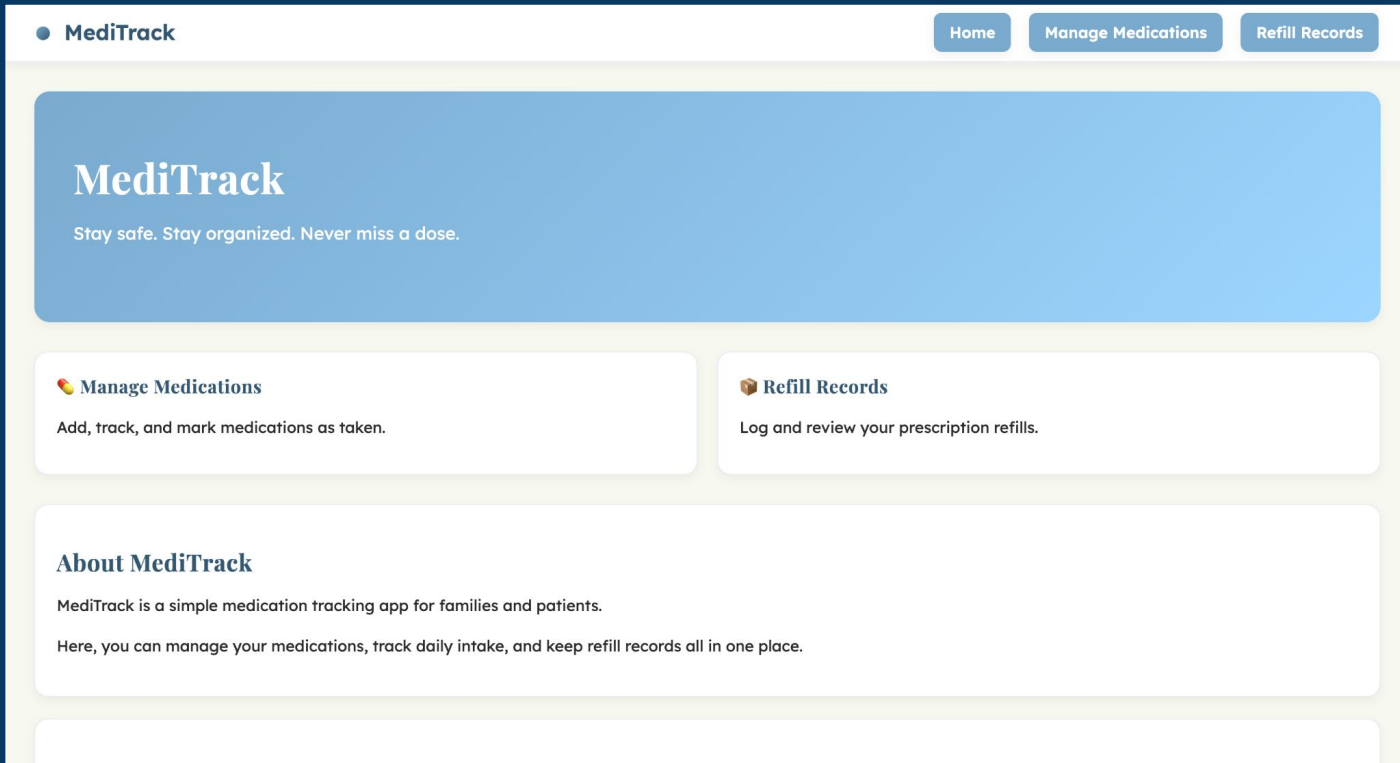- **Architecture**: REST API with modular ES Modules

03

User Personas

- **Maria (Caregiver)** — Manages her elderly mother's medications. Needs to organize prescriptions, check refill status, and confirm medications are taken.
- **Jason (Elderly Patient)** — Takes several daily medications. Needs a simple list, refill tracking, and a way to mark medications as taken.
- **Jessica (Chronic Condition)** — Self-manages her routine. Needs reliable records of prescriptions, refills, and daily intake.

04

Design Mockups

05

MediTrack

MediTrack

Stay safe. Stay organized. Never miss a dose.

💊 Manage Medications

Add, track, and mark medications as taken.

📦 Refill Records

Log and review your prescription refills.

About MediTrack

MediTrack is a simple medication tracking app for families and patients.

Here, you can manage your medications, track daily intake, and keep refill records all in one place.

Live Demo -

06

Proudest moment

- Building a full-stack CRUD app with two MongoDB collections and clean separation between frontend and backend
- Achieving modular code organization — each page has its own HTML, CSS, and JS files
- Implementing inline editing on the Refill Records page — the form switches between Add and Edit mode seamlessly
- Handling 1,000+ records with pagination that works smoothly

Challenge

- Coordinating between team members and ensuring consistent code style, shared navbar, and matching CSS design tokens across independently developed pages
- Seed data field mismatches - the seed script used different field names than the frontend, causing display bugs that required debugging
- No template engine - building all UI with vanilla JavaScript createElement and innerHTML required careful DOM management

Thank you!

Do you have any questions?

diyora.h@northeastern.edu & meiye.l@northeastern.edu