HEALTH2.0

TEAM 1

Revolutionizing how you manage your well-being in the palm of your hand

GitHub: https://github.com/aflam745/CS426-Health-App/tree/milestone-1-submission

Project Overview

Team Members:

- **Owen:** Prescription Page UI
- **Luke:** Journal Page UI
- **Mattheus:** Home Page UI
- **Andrew:** Info Page UI/nav bar

Problem:

Patients face challenges managing prescriptions, refills, renewals, and logging health metrics efficiently.

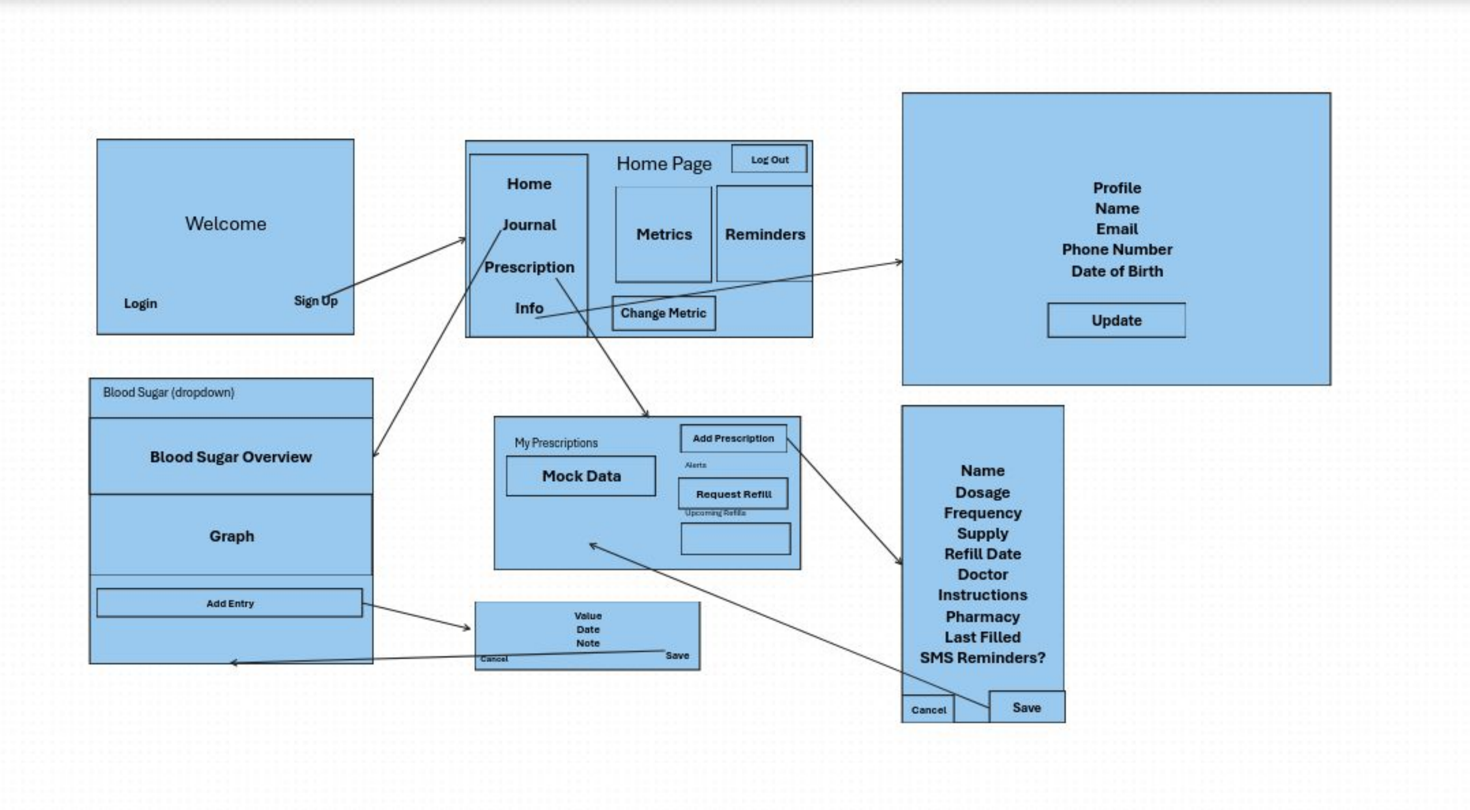
Solution:

A mobile app to track prescriptions, set reminders, manage refills, streamline doctor renewals, and log health data.

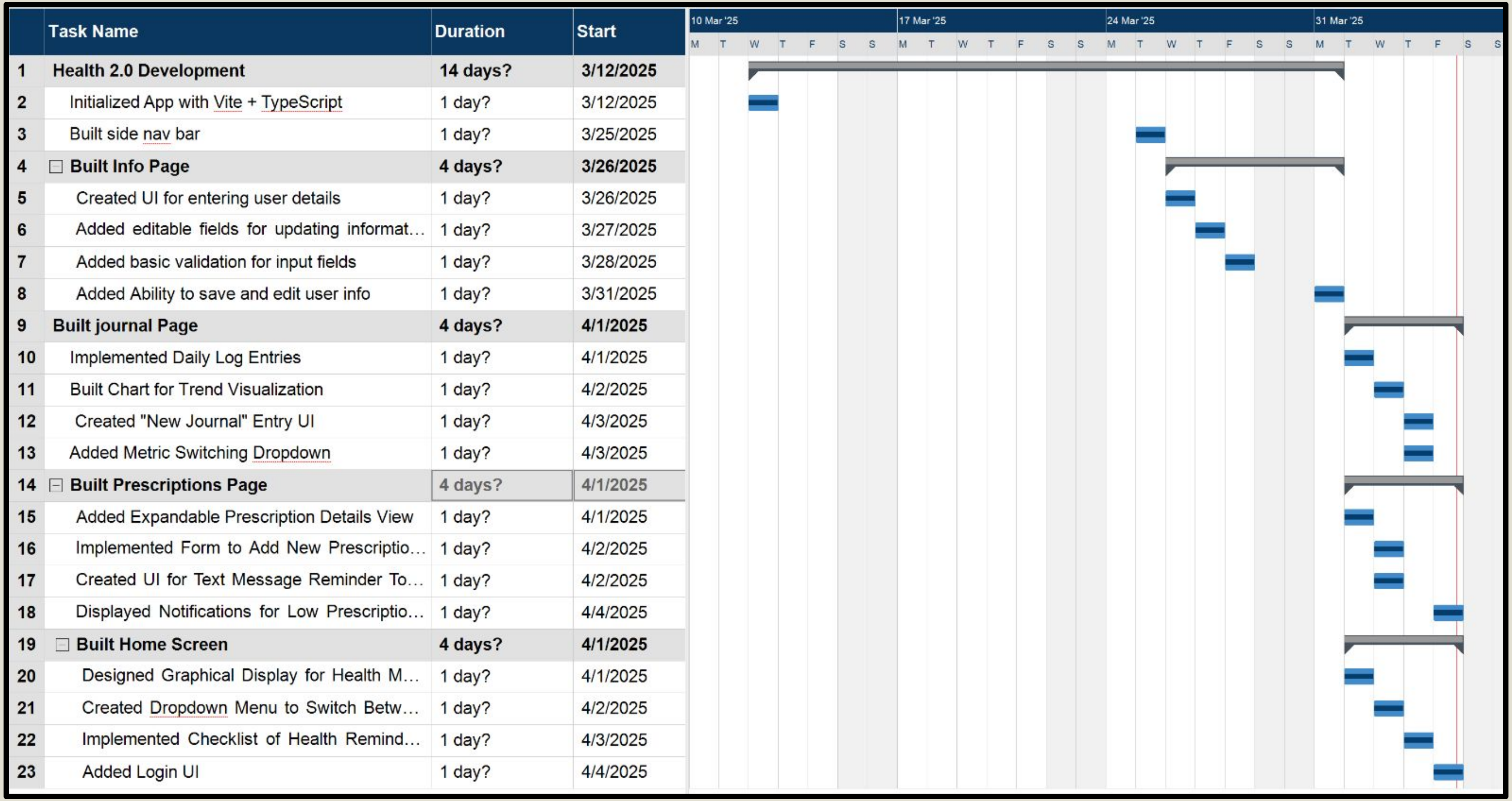
Target Audience:

- Chronic condition patients
- Elderly & caregivers
- Health-conscious individuals

Software Architecture Overview



Historical Development Timeline



	Task Name	Duration	Start
1	**Health 2.0 Development**	**14 days?**	**3/12/2025**
2	Initialized App with Vite + TypeScript	1 day?	3/12/2025
3	Built side nav bar	1 day?	3/25/2025
4	□ **Built Info Page**	**4 days?**	**3/26/2025**
5	Created UI for entering user details	1 day?	3/26/2025
6	Added editable fields for updating informat...	1 day?	3/27/2025
7	Added basic validation for input fields	1 day?	3/28/2025
8	Added Ability to save and edit user info	1 day?	3/31/2025
9	**Built journal Page**	**4 days?**	**4/1/2025**
10	Implemented Daily Log Entries	1 day?	4/1/2025
11	Built Chart for Trend Visualization	1 day?	4/2/2025
12	Created "New Journal" Entry UI	1 day?	4/3/2025
13	Added Metric Switching Dropdown	1 day?	4/3/2025
14	□ **Built Prescriptions Page**	**4 days?**	**4/1/2025**
15	Added Expandable Prescription Details View	1 day?	4/1/2025
16	Implemented Form to Add New Prescriptio...	1 day?	4/2/2025
17	Created UI for Text Message Reminder To...	1 day?	4/2/2025
18	Displayed Notifications for Low Prescriptio...	1 day?	4/4/2025
19	□ **Built Home Screen**	**4 days?**	**4/1/2025**
20	Designed Graphical Display for Health M...	1 day?	4/1/2025
21	Created Dropdown Menu to Switch Betw...	1 day?	4/2/2025
22	Implemented Checklist of Health Remind...	1 day?	4/3/2025
23	Added Login UI	1 day?	4/4/2025

UI Design and Styling Guidelines

UI/UX Overview:

- **Color Schemes**: Custom-defined global color palette with consistent themes across light and dark modes.
- **Typography**: System-default fonts with clear heading (text-xl, font-bold) and descriptive text styles (text-sm, text-muted-foreground).
- **Layout & Spacing**: Consistent vertical spacing (space-y-8) and standardized component padding.

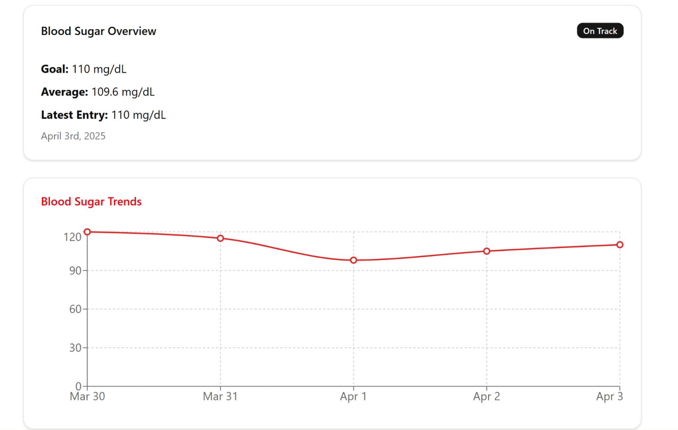
Accessibility:

- Semantic HTML elements with explicit label-input associations.
- Strong color contrasts and clear interactive states (hover, focus, active).

Responsive Design:

- **TailwindCSS** responsive utility classes (sm:, md:, lg:, xl:).
- Adaptive **Shadcn** components tested across multiple device sizes.

Full Guidelines: Link to Style Guidelines



Blood Sugar Overview

On Track

Goal: 110 mg/dL

Average: 109.6 mg/dL

Latest Entry: 110 mg/dL

April 3rd, 2025

Blood Sugar Trends

Annotation: This screenshot is a chart for tracking blood sugar levels. The red color quickly highlights critical health trends, enhancing immediate visibility. Bold typography and the clean layout with ample spacing ensures readability. The responsive chart clearly communicates data trends, improving usability and accessibility.

Performance Considerations

State Management for Metrics/Logs:

- As user logs grow we must avoid storing the entire user log in global state which can be done efficiently by fetching summaries for each chart utilizing caching or background refetching. If we ignore this and the amount of logs grow to a large amount, Pages like Journal or Home could lag or crash trying to iterate those logs.

Authentication and User State

- We should store and fetch our user session using something like Context or cache, to make sure components are not redundantly fetching the same user info, which can lead to poor performance

Efficient Chart Rendering:

- As the data sets for the logs begin to grow, our graphs may become less responsive, cause freezes, and even spike a systems CPU/Memory if they are trying to load all the logs at once. We should try to limit as much recalculations of the chart for each re-render, possibly storing it and rendering it when it needs to be visible to the user

Lazy Loading/Cutting Components

- We should pay attention to where we can apply lazy load. Even though much of the data is shared among pages, we can wait to load pages like the prescription page and most of its contents until the user decides to visit that page. Also, making sure we don't have components dominating loads unnecessarily and can be split up with its function. These two focuses will trim our initial load.

Individual Team Member Slides

Luke Walsh - Assigned Work Summary

Issues:

- Journal Page View (#13)
- Design journal overview page for user-created health metrics (#14)
- Create chart to track progress over time (User entered data) (#15)
- Implement daily log entry feature (#16)
- UI for creating a new journal with goals and optional reminders (#17)
- Add dropdown menu to switch between metrics (#18)

Pull Request:

- Implemented Journal Page UI (#45)

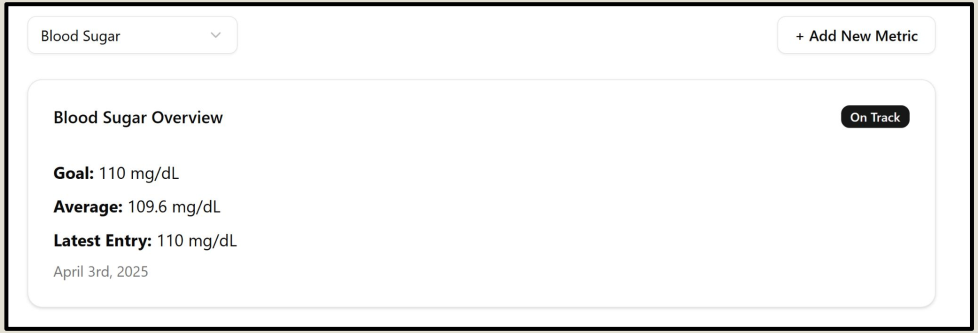
Commits:

- Implemented Journal Page UI

Summary:

- In this milestone, I implemented the Journal Page View which allows users to track custom health metrics, visualize their progress over time, and log daily entries. The journal includes a dropdown to switch between tracked metrics, a summary card showing the goal, average, and latest entry, and a dynamic line chart that visualizes trends using mock data. Users can also add new entries through a form built with input validation and date selection. All UI was built using the Shadcn component library to maintain consistency and enhance visual clarity. Routing was configured so the page is accessible via /journal.

Luke Walsh - Code & UI Explanation



Blood Sugar ⌄

+ Add New Metric

Blood Sugar Overview

On Track

Goal: 110 mg/dL

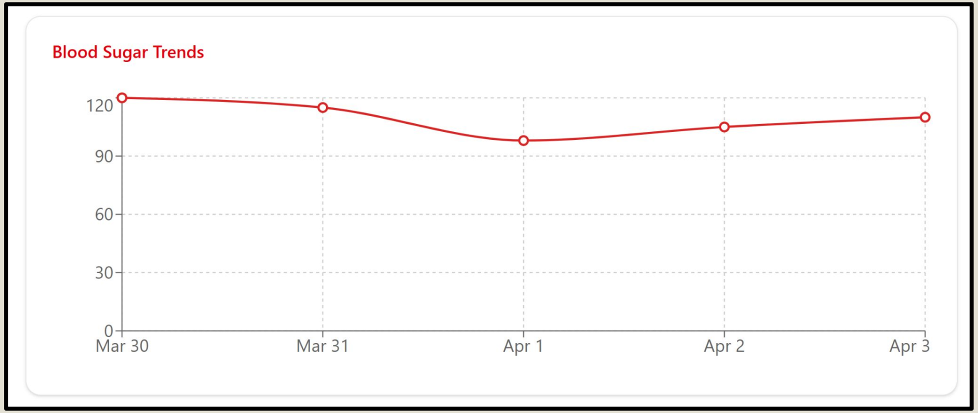
Average: 109.6 mg/dL

Latest Entry: 110 mg/dL

April 3rd, 2025

The metric selector allows users to switch between different health metrics they are tracking, such as Blood Sugar or Sleep. The “Add New Metric” button is a placeholder for creating custom metrics with personalized goals and reminders. The card displays the user’s goal, current average, and latest entry to provide a clear summary of progress. The status badge updates automatically based on the average value, helping users quickly understand if they are on track.

The trend chart visualizes metric values over time, allowing users to spot patterns in their health data. The X-axis displays the dates for each logged entry, while the Y-axis shows the values using a red line that matches the app’s theme. This chart provides users with a clear and intuitive way to monitor changes and progress.



Luke Walsh - Challenges and Insights

.Challenges

- Working with Shadcn/ui was a learning curve at first, especially when customizing components like dropdowns, date pickers, and charts to match our red-and-white theme. I also had to find the right balance between keeping components modular without overcomplicating the structure, which took some trial and error as I built out the Journal page.

Insights

- This milestone helped me get more comfortable with React concepts like state management, prop drilling, and using hooks like useMemo to optimize rendering. I also gained experience structuring components in a clean, scalable way. Working with controlled components and integrating them with mock data gave me a better understanding of how React handles form inputs and dynamic UI. Overall, I feel more confident building interactive views that respond to user input in real time.

Future Improvements: The next step will be integrating backend support so that journal entries are stored and retrieved for each user. I’ll need to update components to fetch data asynchronously, handle loading states, and post new entries to the server. I also plan to connect user accounts to their health data so logs and goals are personalized.

Andrew Flammia- Assigned Work Summary

Issues:

- Set up routes (#40)
- UI/UX Design Document (#42)
- App navigation bar (#24)
- Create UI for Info Page (#32)
    - Add logout button (#35)
    - Form for user info (#28)
    - Form validation (#29)
    - Save and edit info (#30)
- **Not Completed:**
    - Notifications feature (#33)
    - Not a front end feature will require backend API.

PRs:

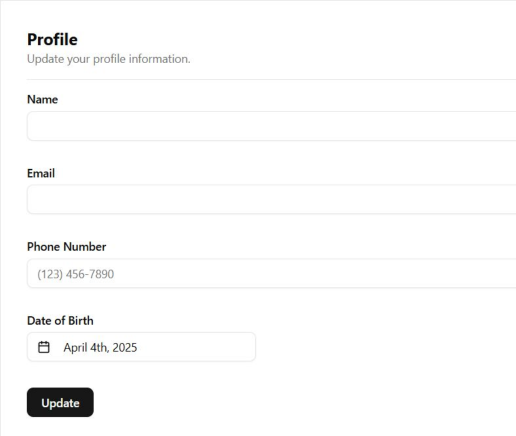
- Created info page and sidebar (#41)
- Created UI/UX Design Documentation (#43)

Commits:

- Create application
- Added Info page, Navigation sidebar, and setup routing
    - I accidentally made a bunch of changes in another repo and had to merge them all into the main repo at once so had multiple features committed in one commit, I could not figure out a better way to do it unfortunately.
- Add logout button
- Created UI/UX Design Document

Summary: In this milestone I created the Info/Profile page where the user can enter and update their personal information in a form, that will be used throughout the application for things like notifications. I also implemented a header and a sidebar for navigation between each of the different pages in the application. I used the Shadcn component library to help in building my features.

Andrew Flammia- Code & UI Explanation



Profile

Update your profile information.

Name

Email

Phone Number

(123) 456-7890

Date of Birth

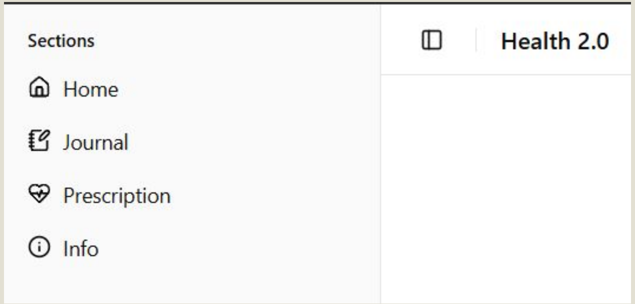
April 4th, 2025

Update

Profile form component: Allows user to entire personal information used throughout the application. The form allows the user to seamlessly update their personal information which will be necessary as later on I integrate the backend process that will allow the user to be notified about important health information.

Sidebar and header: collapsible sidebar that allows user to navigate throughout the application. The sidebar contains the routes to each of the pages in the application with custom icons and texts to portray each page. This feature makes it easy for the user to switch between pages and follows clear styling standards. In addition I added in the header with the title of the application and button to allow the user to collapse the sidebar and create a larger screen view.

Code: shows how I leveraged the Shadcn sidebar ui components to create this custom sidebar. The code makes it so it is seamless to add in more options in the sidebar and easily connect their routing. This dynamic building of the navigation system makes the UI scalable because if future features are added they can easily be integrated into the existing UI.



Sections

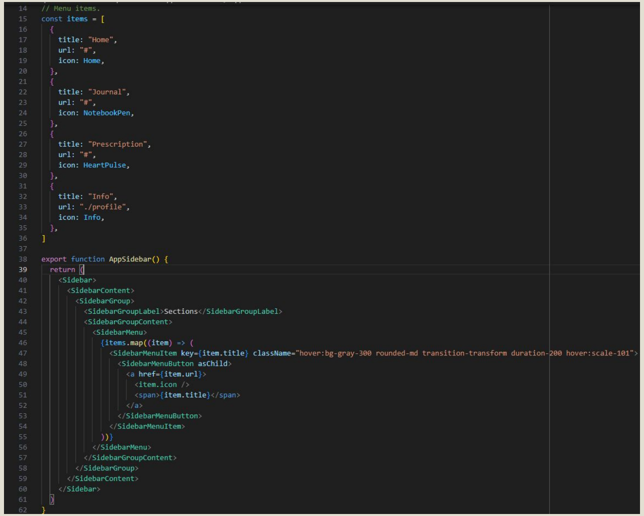
Home

Journal

Prescription

Info

Health 2.0



```
// Menu items.
const items = [
  {
    title: "Home",
    url: "#",
    icon: Home,
  },
  {
    title: "Journal",
    url: "#",
    icon: NotebookPen,
  },
  {
    title: "Prescription",
    url: "#",
    icon: HeartPulse,
  },
  {
    title: "Info",
    url: "./profile",
    icon: Info,
  },
]

export function AppSidebar() {
  return (
    <Sidebar>
      <SidebarContent>
        <SidebarGroup>
          <SidebarGroupLabel>Sections</SidebarGroupLabel>
          <SidebarGroupContent>
            <SidebarMenu>
              {items.map((item) => (
                <SidebarMenuItem key={item.title} className="hover:bg-gray-300 rounded-md transition-transform duration-200 hover:scale-101">
                  <SidebarMenuButton asChild>
                    <a href={item.url}>
                      <item.icon />
                      <span>{item.title}</span>
                    </a>
                  </SidebarMenuButton>
                </SidebarMenuItem>
              ))}
            </SidebarMenu>
          </SidebarGroupContent>
        </SidebarGroup>
      </SidebarContent>
    </Sidebar>
  )
}
```

Andrew Flammia- Challenges and Insights

Challenges:

- At first working with the Shadcn/ui component library was a bit of a learning curve, when adapting their prebuilt components to the vision I had for my features. It took some studying of the documentation as well tweaking of styles and components to get each part to work and look correct.
- Specifically getting the collapsing sidebar to work properly and the rest of buttons and page to stay connected when collapsing and opening the bar. With time and digging into generated DOM tree I was able to get everything working perfectly.

Insights:

- I learned a lot about react and how to properly implement reusable UI components in this milestone.
- It’s very important in this process to incrementally test features, by having the app running on another screen, so that it was easier to fix problems when they arise.
- When working in a team environment I learned it is important to have clear plan so everyone is on the same page and can easily integrate everyone's features.

Future Improvements: What will come next for me will be integrating backend APIs that will take the users email and phone number and will provide them with important updates on their health metrics that are tracked on the other pages. I also work on figuring out the process to connect to things like CVS or Mychart so the user can easily refill their prescriptions or get messages from their doctor. As well I will work with my team and make updates to the UI if we see fit in the future.

Mattheus Ferreira - Assigned Work Summary

Issues:

- Design graphical display for health metrics over time (#3)
- Create a dropdown menu for switching between metrics (#4)
- Implement checklist of health reminders (#5)
- Add login UI (without authentication) (#6)
- Create Mock Data to display ‘User Health Metrics”(#34)

PRs:

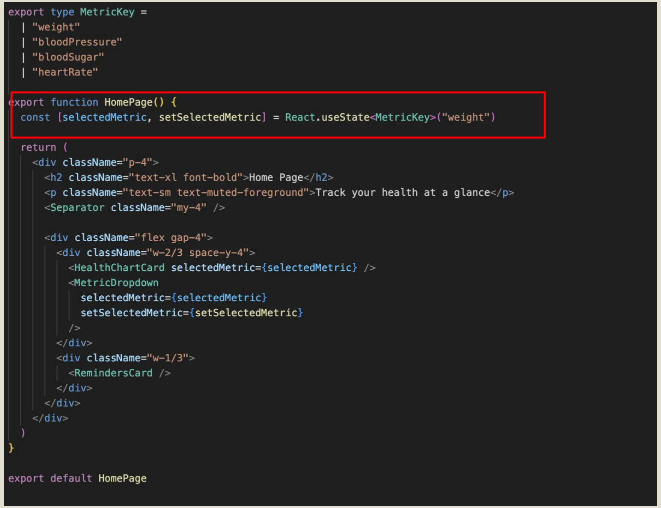
- Home/login page full implementation with route setup (#46)

Commits:

- Updated app-sidebar.tsx to include routing to home page
- Added login and sign up form component
- UI components for graph, dropdown, card styling
- Added home and login page routes
- Updated App.tsx to work with login page route to home
- Components for metric graph, dropdown, and welcome
- Upcoming reminders component with mock reminders

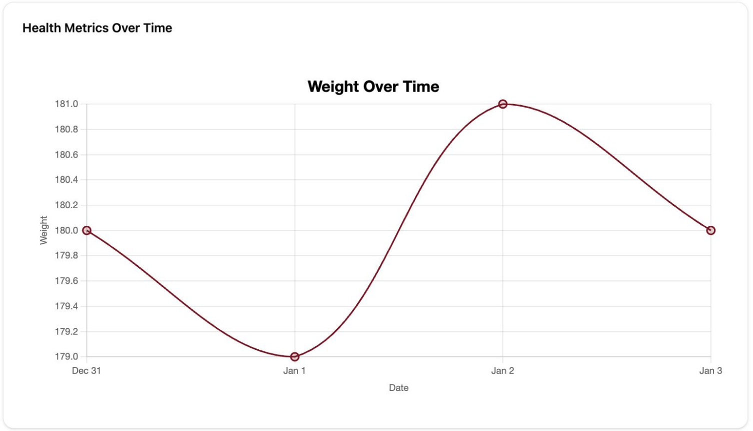
Summary: I built the home and login pages for the application. The home page has a dynamic line chart to visualize mock health metrics like weight, blood pressure, etc that is determined by dropdown menu present below the graph. The home page also hold a list of all upcoming reminders that are either user-set like ‘don’t forget to drink water’ or determined like ‘Need to refill ‘X’ prescription”. The login page is a standard login without authentication page that routes automatically after a successful login or signup to the homepage.I set the routing for the homepage for /home and the login page for /login both being accessible by the sidebar and utilized Shadcn for my styling throughout my pages and components.

Mattheus Ferreira - Code UI



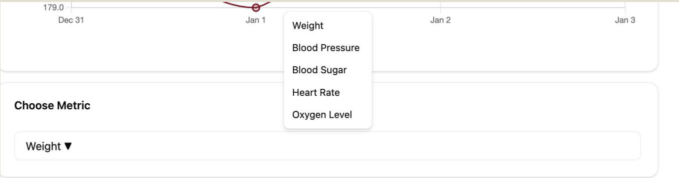
```
export type MetricKey =
  | "weight"
  | "bloodPressure"
  | "bloodSugar"
  | "heartRate"

export function HomePage() {
  const [selectedMetric, setSelectedMetric] = React.useState<MetricKey>("weight")

  return (
    <div className="p-4">
      <h2 className="text-xl font-bold">Home Page</h2>
      <p className="text-sm text-muted-foreground">Track your health at a glance</p>
      <Separator className="my-4" />

      <div className="flex gap-4">
        <div className="w-2/3 space-y-4">
          <HealthChartCard selectedMetric={selectedMetric} />
          <MetricDropdown
            selectedMetric={selectedMetric}
            setSelectedMetric={setSelectedMetric}
          />
        </div>
        <div className="w-1/3">
          <RemindersCard />
        </div>
      </div>
    </div>
  )
}

export default HomePage
```

Interactive line chart built using Chart.js and Shadcn to visualize mock health data over time. Metric is selected from the dropdown shown below and updates graph in real time.



Health Metrics Over Time

Custom dropdown component that was built using Radix UI and Shadcn for selecting different health metrics. Selecting an option dynamically updates the graph seen above.



Choose Metric

Weight ▼

In the homeCard component, I lifted the *selectedMetric* state so both the dropdown and chart could access and respond to it. This allowed me to create a dynamic UI where selecting a metric from the dropdown immediately updates the graph below without reloading or re-rendering the entire page. The component structure keeps logic clean and scalable, with *MetricDropdown* handling user input and *HealthChartCard* handling visual output seamlessly fora positive user experience while maintaining a scalable and modular component structure.

Mattheus Ferreira - Challenges and Insights

Challenges

- Chart.js has a complex type system, and integrating it in a typescript environment caused multiple issues at first with types and tooltip configuration, so I found to use generics like ChartData<”line”> and typed props explicitly, and then defined callback type to avoid ‘*any*’, which improves type safety

- I started my home page in a monolithic sense that left me lost at times because as the complexity grew, keeping all the components together in one spot became hard to manage. So the solution was to split up the 3 main UI functionalities into independent components that interact with each other in the same way but made it much easier later on to continue building up the page.

Insights

- Styling with something different like shadcn helped enforce this easy and seamlessly experience that helped us all maintain UI styling standard while still producing an appealing product

- Lifting state up is powerful, it allows my components to become reusable and tightly synchronized with them being completely independent from each other

- Although I always review my code before a PR myself plenty of times, having someone with a fresh set of eyes look at my code and instantly notice something is refreshing and much more comforting knowing that the product I am pushing out has been reviewed and vetted for merging or even if it needs a few meaningful tweaks that just gives the piece of mind that the code is being perfected.

Owen Gibbons - Assigned Work Summary

Issues:

- Prescriptions Page #7
    - Display prescription list with mock data #8
    - Expandable Prescription Details Field #9
    - Implement Form for adding new prescriptions #10
    - UI for text message reminder toggle #11
    - Display notifications for low supply #12

Commits:

- Added types
- Skeleton for hooks and components
- Hooks
- Components + added route
- added route to the sidebar and fixed issues.
- Added sms reminder toggle when creating new rx
- Fixed form to actually use usePrescriptionForm. Now has all appropriate fields and submitting also works.
- Put the prescription form in another component
- Removed phone number input

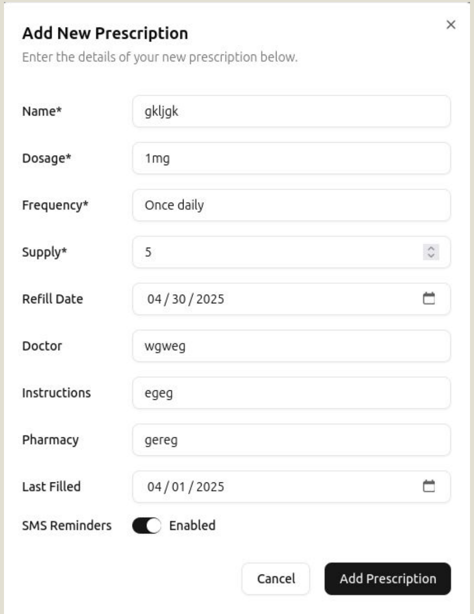
PRs:

- Prescriptions Page #44

Summary:

I made a mostly functional prescriptions page where you can add a new prescription, you can get alerted when supply is low, sign up for text notifications, see the details of your current prescriptions, and there is a view for alerts and upcoming refills to be on top of supplies. There are also options to request refills, but these do not do anything yet. The button in the expanded view to set a reminder has also not been implemented yet.

Owen Gibbons - Code and UI Explanation



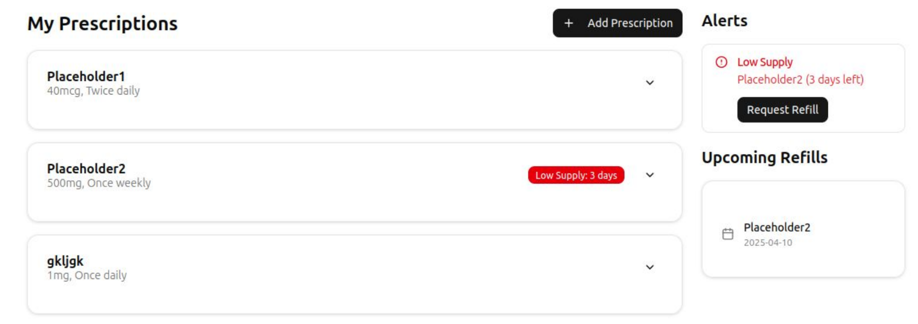
Add New Prescription

Enter the details of your new prescription below.

Name*	gkljgk
Dosage*	1mg
Frequency*	Once daily
Supply*	5
Refill Date	04 / 30 / 2025
Doctor	wgweg
Instructions	egeg
Pharmacy	gereg
Last Filled	04 / 01 / 2025
SMS Reminders	● Enabled

Add prescription form

This is a relatively simple form that uses Shadcdn to take in the details to pass on to the main prescriptions page. It uses my usePrescriptionForm to validate input and handles the submission process.



My Prescriptions

Placeholder1
40mcg, Twice daily

Placeholder2
500mg, Once weekly
Low Supply: 3 days

gkljgk
1mg, Once daily

Alerts

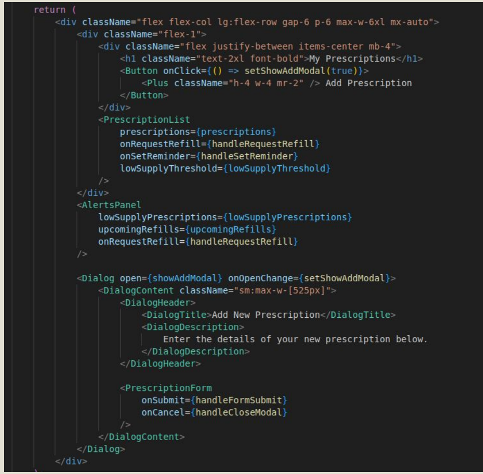
Low Supply
Placeholder2 (3 days left)

Upcoming Refills

Placeholder2
2025-04-10

Prescriptions Page

This is a snippet from my main prescriptions code. It utilizes several Shadcdn components as outlined in our styling guidelines, in addition to my own components (PrescriptionsList, PrescriptionForm, AlertsPanel). These components help with modularity and maintaining clean code (this code looked much worse before I made the form into its own component). I leverage the functions within my components to easily take in input and output the results to the main page. You can see how submitting the form added a new card with the details from the form to the main page.



```
return (
    <div className="flex flex-col lg:flex-row gap-6 p-6 max-w-6xl mx-auto">
        <div className="flex-1">
            <div className="flex justify-between items-center mb-4">
                <h1 className="text-2xl font-bold">My Prescriptions</h1>
                <Button onClick={() => setShowAddModal(true)}>
                    <Plus className="h-4 w-4 mr-2" /> Add Prescription
                </Button>
            </div>
            <PrescriptionList
                prescriptions={prescriptions}
                onRequestRefill={handleRequestRefill}
                onSetReminder={handleSetReminder}
                lowSupplyThreshold={lowSupplyThreshold}
            />
        </div>
        <AlertsPanel
            lowSupplyPrescriptions={lowSupplyPrescriptions}
            upcomingRefills={upcomingRefills}
            onRequestRefill={handleRequestRefill}
        />

        <Dialog open={showAddModal} onOpenChange={setShowAddModal}>
            <DialogContent className="sm:max-w-[525px]">
                <DialogHeader>
                    <DialogTitle>Add New Prescription</DialogTitle>
                    <DialogDescription>
                        Enter the details of your new prescription below.
                    </DialogDescription>
                </DialogHeader>

                <PrescriptionForm
                    onSubmit={handleFormSubmit}
                    onCancel={handleCloseModal}
                />
            </DialogContent>
        </Dialog>
    </div>
```

PrescriptionsPage Code

Owen Gibbons - Challenges and Insights

Challenges

- I had several issues when it came to dependencies. At one point, I had to delete my package-lock file and run npm install again to fix the problem.
- I had to re-reference the useState and useEffect functions to make sure I fully understood how they worked before I tried implementing them in my code.
- Implementing the form was tough for me, since it integrated a lot of Shadcdn components, and I also had to remember how to use local storage correctly to ensure the cards stayed even if you refreshed the page.
- There were some weird errors with Typescript that appeared when I tried building the application, but didn’t when I just ran the app, which I found interesting

Insights

- Code reviews are absolutely essential – there were issues that I hadn’t even thought of that Andrew brought up in code review, like making the form into its own component to make the main component simpler, and there were also small things that I had overlooked after coding on the same thing for so long, like not needing a phone number because the account will handle that along with discrepancies in what the form takes in vs what the page shows.
- External libraries make things so much easier – Shadcdn allowed me to quickly make beautiful UI components that otherwise would have taken a while to think up and get right with regular CSS or even Tailwind.

What’s Next

Future Improvements & Next Steps

- We will next be implementing a notification system to notify users via text message and/or through email with prescription filling reminders and reminders to log health metrics.
- Figure out and implement the process to connect to things like CVS or Mychart so the user can easily refill their prescriptions or get messages from their doctor.
- Create the account and login functionality so users can securely log into their account and view their information.
- Connect all of our forms to a backend database so information can be stored and does not get refreshed every time they log in.
- Implement a toggle to switch between light and dark mode for the UI.
- Implement an achievement/rank system so that the goals are gamified.