

CapStone

Plant Tracking app

Auth

So far i have github implemented
Want to add Facebook and Google still

Facebook was a BUST, google and github are working.

Need to create a profile settings page, there might be a template

All pages are going to be wrapped in a protected route
i need a landing page with a description that has a login button on it
Ended up just foring redirects and creating a setup page

API

Sidebar

Looking really good, all the routing is working and all the routes are authenticated.
Got user info populated in the footer.

Actual plant tracker

Figure out how to link to mongoDB
this is working! i have a template for plants set up and a card that fetches and displays all thier info

make a plant api or find one

Humidity tracking with a smart device?

Dashboard

I want to have a timeline of tracked activities, I guess this page will have a calendar with days you need to do something aswell as logging stats, watering stats etc

	Page title
	<div>activity timeline</div> <div><div>Gamification station</div><div><div>Carrousel of recently uploaded plant pics</div><div>TBD</div></div></div>

My plants

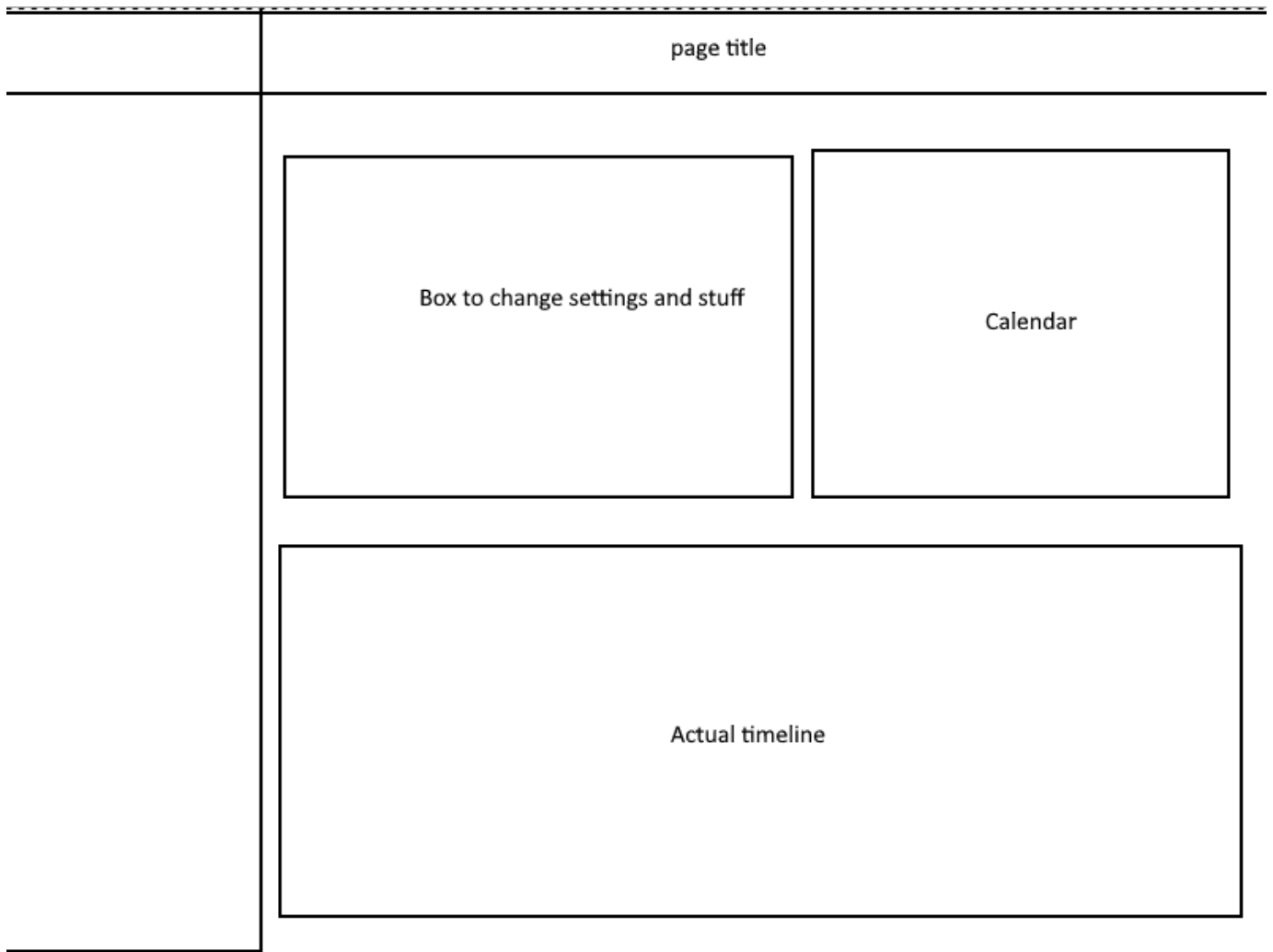
This is where you manage your plant profiles aswell as upload proggress photos and notes

	Select 1 or multiple			
<div>Pops up a modal to enter plant info and add picture</div> <div>Edit button</div>	<div>Add plant</div>	<div>delete plant</div>		
	<div>plant pic</div> <div>plant name</div> <div> <div></div> <div></div> </div>			
	<div>View Full info modal</div>			

Plant info

Routines

This is going to have the whole select list of plants you want to manage and it will generate a care timeline with tips and stuff



Notes

Starting on the mongodb integration now, will doc any issues i have
not tooo many issues, mostly just api bugs as im learning
LMAO had a folder missing an s and that was breaking the api...

Finally got my first hydration error! - it's still happening.. :(

I still need to add framer motion animations... maybe later in the process
though

Gotta figure out notifications too, surely not too hard...

12 6

So for today, i want to clean up the final design of the dashboard. Fix the picture
carousel and implement uploading images to the DB that the dashboard will pull from

I also want to eventually add a banner in the header of recently achieved badges from

other accounts, scrolling along the header.

#CHANGES TO IMPLEMENT -----

Highlight the current page in the sidebar

Hide the scroll bar or atleast restyle it

add an HR below the profile pic and name in the footer

Restyle settings pages to match the notification page

Clean up all the settings menues, add missing stuff and remove unnessecary stuff

Split the activity timeline into weeks instead of months

#MISSING CORE FUNCTIONALITY-----

Routines

uploading plants

customizing uploaded plants

Change the Plant info page to have a search box, as well as pull care tips for all of the users currently added plants, still need to wireframe this

Figure out how to do notifications - I want a next upcoming task and any pending notidications

I also really want to add an idle gardening game somewhere... maybe after the project is submitted

[12 12](#)

[12 13](#)

[12 14](#)

[12 15](#)

[12 30](#)

Bugs

When trying to edit a plant in the my plants page through the modal this error occurs i've tried adding await to the params and still get the same error..

Error: Route "/api/my-plants/[id]" used `params.id`. `params` should be awaited before using its properties. Learn more: <https://nextjs.org/docs/messages/sync-dynamic-apis>
at PATCH (file:///H%3A/School/Capstone/v3/capstone-v3/src/app/api/my-plants/%5Bid%5D/route.ts:61:35)

```
59 |  
60 |   try {  
> 61 |     const plantId = context.params.id; // Use context.params properly  
    |
```

```
62 |     const body = await req.json();
63 |
64 |     if (!ObjectId.isValid(plantId)) {
PATCH /api/my-plants/67622238a997d8a3bac0e89e 200 in 669ms
```

Capstone Proposal

Project Overview

The Plant Care Tracker is a web application designed to help plant enthusiasts manage their plant care routines, track growth, and troubleshoot common plant care issues. This app aims to provide an organized and visually appealing platform where users can keep track of each plant's unique care requirements, set reminders for watering and other tasks, and maintain a journal of their plant's growth. With personalized care instructions and timely reminders, users can ensure that each of their plants receives the proper attention to thrive.

Purpose and Problem Statement

Houseplants are a popular addition to many homes, and caring for them requires consistent attention to detail. Without a structured care system, it's easy for plant owners to overlook watering, fertilizing, or other care tasks, which can lead to plant health issues. The Plant Care Tracker offers a convenient solution for plant owners to keep their plants organized and healthy by reminding them of scheduled tasks and tracking plant growth. This app targets users of all experience levels—from beginners who need guidance on basic care to seasoned plant enthusiasts managing larger collections.

Target Audience

- Indoor Gardeners and Plant Enthusiasts: Individuals who enjoy indoor gardening and want to track plant care.
 - Beginners and New Plant Owners: Those new to plant care who need help with consistent routines and guidance on plant health.
 - Advanced Users and Collectors: Experienced plant owners managing multiple plants with complex care needs.
-

Core Features

1. Dashboard Overview

- A central dashboard displays all upcoming care tasks and recent entries for all plants.
- Highlights plants needing immediate attention, such as overdue watering.

3. Plant Profiles

- Users can create individual profiles for each plant, including nickname, species, light requirements, and care instructions.
- Each plant profile allows users to add photos and notes, creating a detailed care record over time.

5. Reminders and Notifications

- Users can set reminders for recurring tasks like watering, fertilizing, and pruning.
- Notifications are sent through email or browser notifications, helping users stay on top of plant care schedules.

7. Plant Growth Tracker

- Users can document plant growth by periodically adding photos and notes to each profile.
- Visual growth timelines display trends over time, giving users a sense of their plant's progress.

9. Care Tips and Troubleshooting

- The app provides a built-in knowledge base with care tips and troubleshooting guides.
- An interactive troubleshooting section offers solutions to common plant problems (e.g., yellowing leaves, pests, or overwatering).

11. Search and Filter

- Users can search for specific plants or filter by criteria, such as care requirements or task status.
- Allows users to organize plants by category, making it easy to manage multiple plants.

Technology Stack

- Frontend:
 - Next.js: Chosen for its powerful framework and component-based architecture, Next.js enables a responsive, dynamic user interface with optimized performance. Its versatility supports seamless feature integration, while server-side rendering and static generation ensure fast loading times and smooth transitions between sections.
 - ShadCN and Tailwind CSS: ShadCN will provide a cohesive design system with ready-made components for cards, lists, and modals, which are essential for the plant profile and dashboard views. Tailwind CSS will be used alongside ShadCN to add custom utility classes for spacing, alignment, and colors, ensuring a polished and consistent design.
- Backend:
 - Next.js: A versatile full-stack framework that handles API routes, reminders, and notifications efficiently. Its built-in API routing enables streamlined endpoint management, making it an ideal choice for all backend operations alongside dynamic server-side features.
 - NextAuth.js: For secure and seamless user authentication, NextAuth.js will handle user sessions, login, and registration flows. It offers flexible integration with multiple authentication providers (e.g.,

OAuth, email) and allows for custom user management. NextAuth.js will help ensure robust security across the app.

- Database:
 - MongoDB: A document-based database that offers flexibility in handling plant profiles, growth logs, and care schedules. MongoDB's structure is ideal for storing varied data (such as user notes, images, and reminders) and supports quick retrieval and updates..
-

User Flow

1. Onboarding and Profile Setup: Upon logging in, new users can create profiles for each of their plants, entering care instructions and setting initial reminders.
 2. Dashboard Navigation: The dashboard provides an overview of upcoming tasks and quick access to plant profiles.
 3. Routine Reminders: Users receive notifications for upcoming tasks, and each completed task is logged in the plant's profile history.
 4. Growth Tracking: Users can periodically upload photos and notes to track each plant's progress over time.
 5. Troubleshooting and Tips: When issues arise, users can reference the knowledge base for specific advice, fostering better plant care.
-

Conclusion

The Plant Care Tracker is designed to be a comprehensive tool for plant enthusiasts, enabling them to manage, document, and improve their plant care routines. This project showcases the ability to build a full-stack application with a modern tech stack, integrating React's UI capabilities with a robust backend and a flexible database. By providing a user-friendly, visually appealing, and feature-rich app, the Plant Care Tracker aims to enhance the experience of plant care for users of all skill levels, promoting healthier plants and a more organized approach to indoor gardening.

**