Green Thumb

CSCI 3308 – Group 013-6

Alexander Eadie Aria Blondeau

Gabrielle Partch Jack Rueschhoff

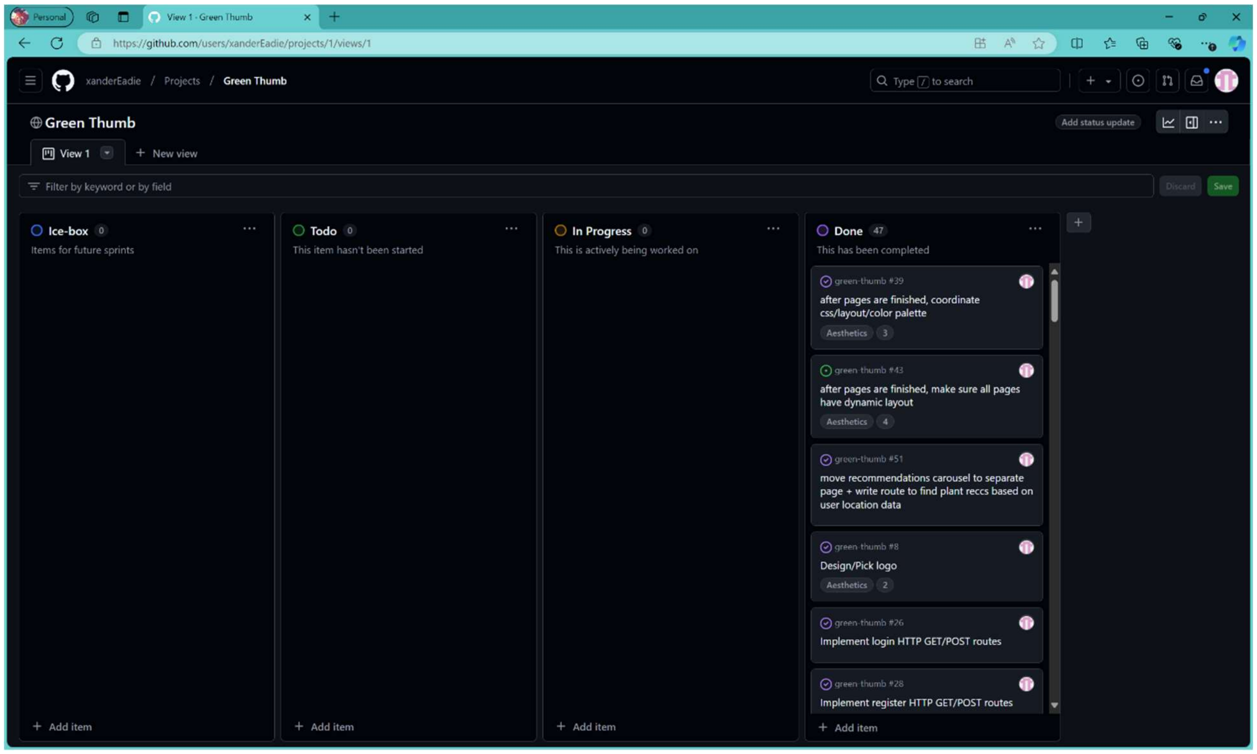
Libo Zou Yubo Wang

Project Description

Green Thumb is a home gardening application designed to assist home gardeners with tracking and finding new plants for their garden. Users can search for plants using filters such as height, cycle, hardiness, sunlight, watering, flowers, and edible to find the best plants to match their garden. Users can then view all these plants' attributes in a more detailed view and add plants to their garden. Users can also view their garden, as well as get recommendations for new plants based on user-specified location attributes. This application is powered by data from the [Perenual API](#).

Project Tracker

Project Tracker Link: [GitHub Project Board](#)



Personal
View 1 · Green Thumb
https://github.com/users/xanderEadie/projects/1/views/1
xanderEadie / Projects / Green Thumb
Type / to search
Green Thumb
Add status update
View 1
New view
Filter by keyword or by field
Discard
Save

Ice-box 0
Items for future sprints

Todo 0
This item hasn't been started

In Progress 0
This is actively being worked on

Done 47
This has been completed

- green-thumb #39
  after pages are finished, coordinate css/layout/color palette
  Aesthetics 3
- green-thumb #43
  after pages are finished, make sure all pages have dynamic layout
  Aesthetics 4
- green-thumb #51
  move recommendations carousel to separate page + write route to find plant reccs based on user location data
- green-thumb #8
  Design/Pick logo
  Aesthetics 2
- green-thumb #26
  Implement login HTTP GET/POST routes
- green-thumb #28
  Implement register HTTP GET/POST routes

Video

Video Link: Green Thumb Demo

Version Control System (VCS)

GitHub Repository: Green Thumb GitHub Repository

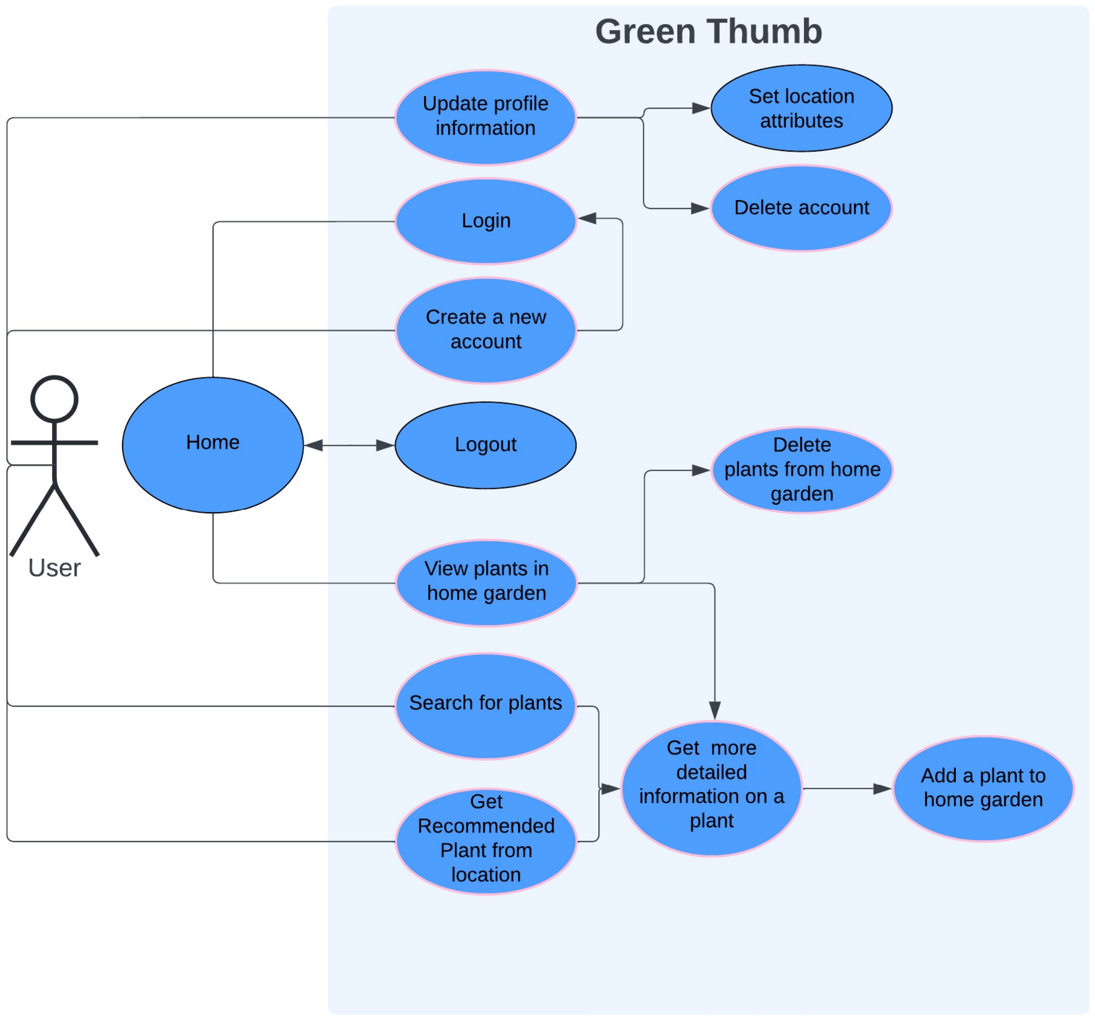
Contributions

- **Xander**
  Created backend functions for garden that helped manage the databases. Worked on other backend functions and helped with functionality of forms on the frontend.
- **Aria**
  Designed the search results page and plant information page. Created implementation and wrote API endpoint routes for search functionality, plant information, add location, delete user, assisted with debugging other API routes as well as created some test cases. Created database by requesting API data and formatting to work as an insert query. Planned and recorded demo video. Wrote README.md file.
- **Gabrielle**
  Created wireframe/ model for initial page functionality/aesthetics, UI and functionality for home, garden, and recommendations pages, configured/set up docker container, completed UI and functionality for footer, header, nav bar, and message partials, set up static resource handling with handlebars to access frontend js and css pages, wrote css for theme text color/sizing and homepage/garden aesthetics, built and/or debugged HTTP routes for home, register, login, logout, adding/updating and getting user location data for location settings, updating user account data, garden, add plant, and remove plant, wrote test cases for login and register, and general aesthetic touch-ups and responsive page layouts.
- **Jack**
  Designed UI and functionally implemented login/register/logout, account settings, location settings, account deletion and profile/settings routes, and assisted with SQL query creation.
- **Libo**
  Helped create the project frame and API endpoint. Worked on the search results page and garden pages and fixed some routes for the settings page.

- **Yubo**
  Created UI template of the profile page using DaisyUI, and setup the navigation to location, settings and other tabs accessible from the profile page. Worked on routes for settings and profile. Setup the render database and connected the web service to both the database and the project, fixed issues where render could not connect to the project database correctly.

Use Case Diagram

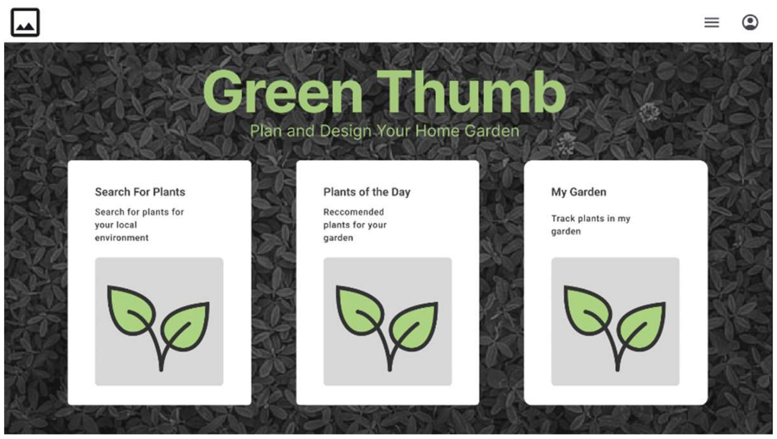
Use Case Diagram Link: Green Thumb Use Case Diagram


Green Thumb
Update profile information
Set location attributes
Delete account
Login
Create a new account
Home
Logout
Delete plants from home garden
View plants in home garden
Search for plants
Get more detailed information on a plant
Add a plant to home garden
Get Recommended Plant from location
User

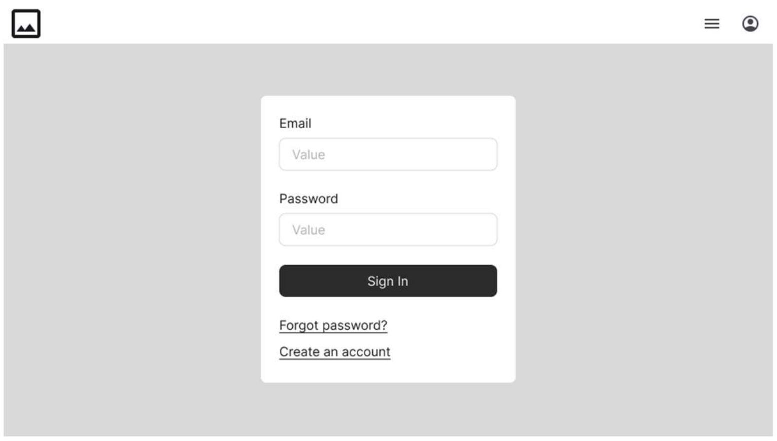
Wireframes

Wireframe Figma Link: Green Thumb Wireframe

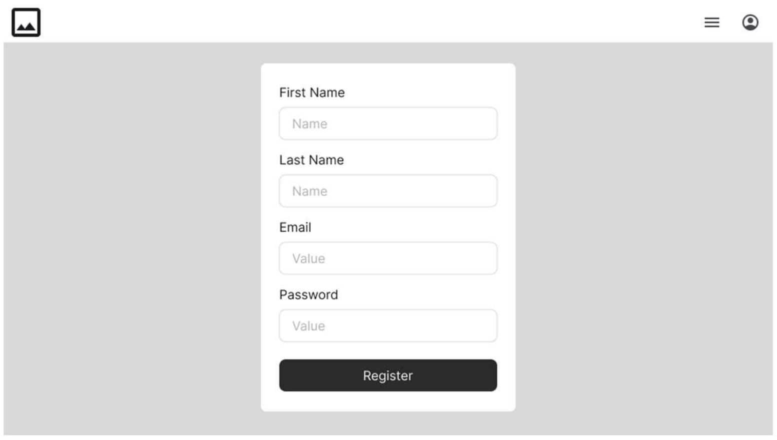
Home


Green Thumb
Plan and Design Your Home Garden
Search For Plants
Search for plants for your local environment
Plants of the Day
Reccomended plants for your garden
My Garden
Track plants in my garden

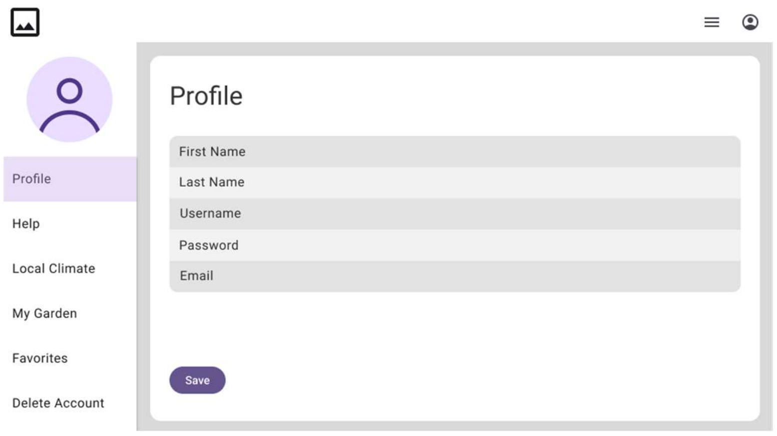
Login



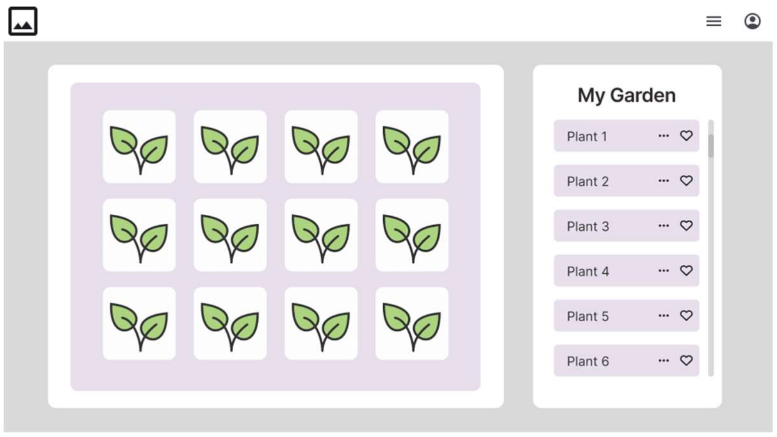
Register



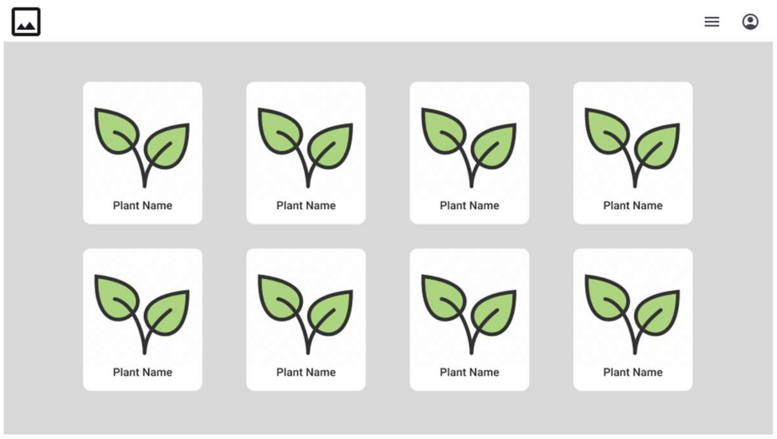
Profile/Settings



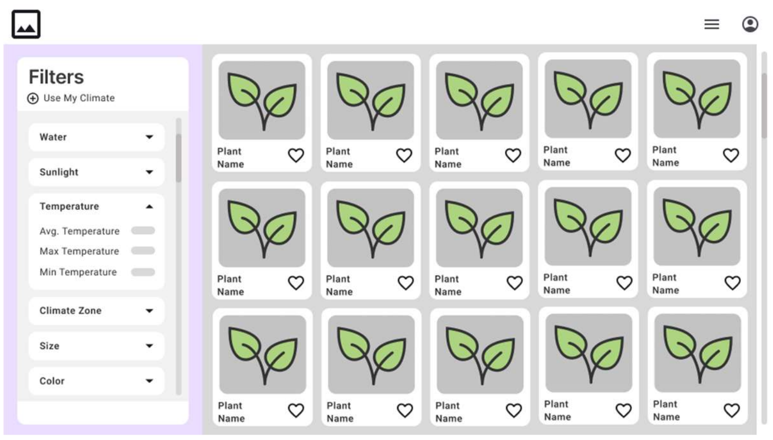
Garden



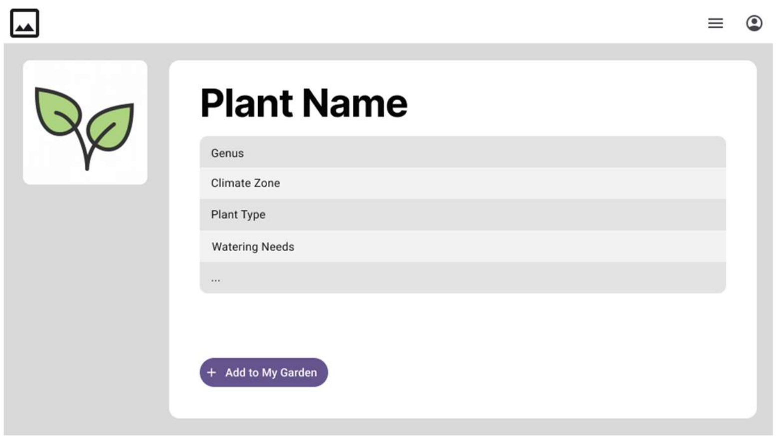
Recommendations



Search



Plant Information



Test Results

Unit Testing

We designed a total of eight test cases for four different routes. Three cases test the register API, testing a positive case where a valid new user is created, testing a negative case where we try to register the same user again and get an error because it's a duplicate, and testing an invalid registration where the password and confirm password fields do not match. For our login API we test the case where the user is valid and should log in successfully, and a case where an invalid user tries to log in that doesn't exist. For the search API we ensure that passing some known good parameters to the search field results in a 200 status. For the plantInformation API we ensure that we can view valid plant information as well as return an error if an invalid plant is specified.

We initially thought we ran into some minor issues with our test cases as the register test only passes the first time the database is initialized, otherwise it fails because the user is already added. However, we realized that this is an intentional feature of mocha/chai, not a bug. We also had some difficulty with testing the routes that required a valid user. The authentication middleware was causing issues with the search and plantInformation tests as these pages are behind our authentication middleware, so while we were running these test cases we had to comment out the authentication middleware.

User Testing

Test Scenario	**Observations/Results**
Register a new user	- When directed to registration, filled out fields as would be typical for a registration page with no issue
Login existing user	- First attempted to access search but was redirected to login - From login, used the 'create new user' link to access registration - After registering, logged in using the credentials generated in register
Add location data	- Once the user located the settings page, they were easily able to understand how to set location data via the menu on the left
Update account information	- Able to update account information as would be typical for an account setting page – no issues here
Access plant information page	- Found it intuitive to click on the image of the plant in recommendations/search/garden to navigate to a more detailed plant information page
Search for plants	- Found the value 'hardiness' difficult to understand as someone who is not familiar with gardening
Add plants to the garden	- Found it intuitive to access plant information page and add plants to garden from there
Remove plants from the garden	- Successfully removed plants that had been added to garden without issue
Access recommendations page	- Attempted to access recommendations before setting location data and understood that was what they needed to do, but weren't sure where to access it - Needs some sort of message that the user can update their location
Other	- Interface looks different/ slightly wonky on their computer – not sure if this is a browser issue or not - Would like to see additional information such as planting season - Would like watering reminders/ ability to set a watering schedule

For the most part, the user found the webpage easy to navigate and intuitive to use. They suggested four changes. First, that we should include a way to get more information about the values in plant search/ the plant information page, especially hardiness. As someone who isn’t familiar with home gardening, they weren’t sure how the numerical value translated to their climate conditions. Second, they said that they’d like to see more plant information such as planting season that would give them more guidance on when/how to plant in their garden. Third, they thought that something like the ability to set a schedule with watering reminders for your garden would be a helpful addition. Finally, when they attempted to access the recommendations page when they hadn’t set their location data, they were confused by the message ‘no location data available.’ They understood that they needed to input their location data somewhere but weren’t sure how to access that. Unfortunately, due to time constraints, we had to skip implementing the more involved suggestions but made sure to clarify the method of setting location data by redirecting to the location settings page when the user hasn’t set their location information yet. The rest will be considered for future improvements/ releases.

Deployment

Website Link: Green Thumb

To run locally, first clone the project via the GitHub Repository. Navigate to green-thumb/ProjectSourceCode and create a file called .env with the contents shown below. Install docker and start the docker engine. Then, while inside the ProjectSourceCode directory in your terminal, enter **docker compose up** to run. Finally, in a browser, navigate to http://localhost:3000/.

.env

```
POSTGRES_USER="postgres"
POSTGRES_PASSWORD="pwd"
POSTGRES_DB="users_db"
SESSION_SECRET="super duper secret!"
```