**Milestone 5 Report**

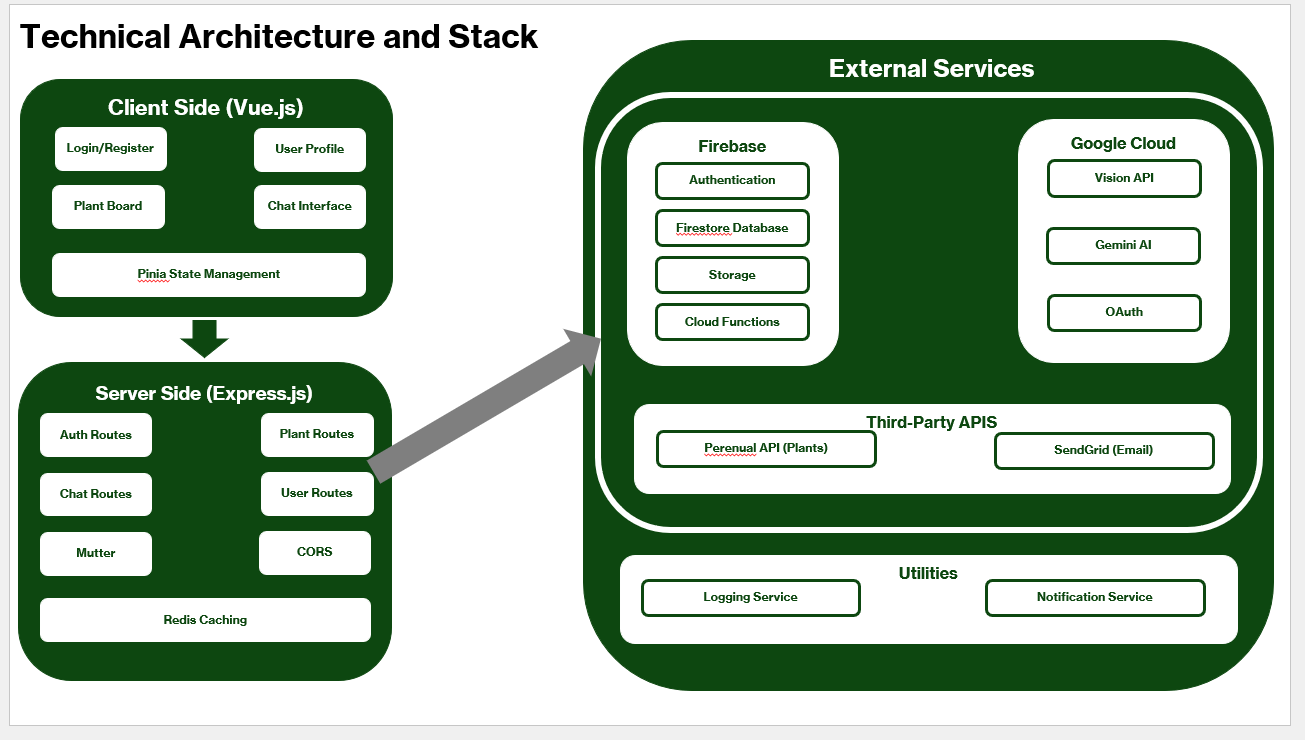
By MaShayla Kendrick, Sahil Patel, and Feifei Xie

Usernames in GitHub:

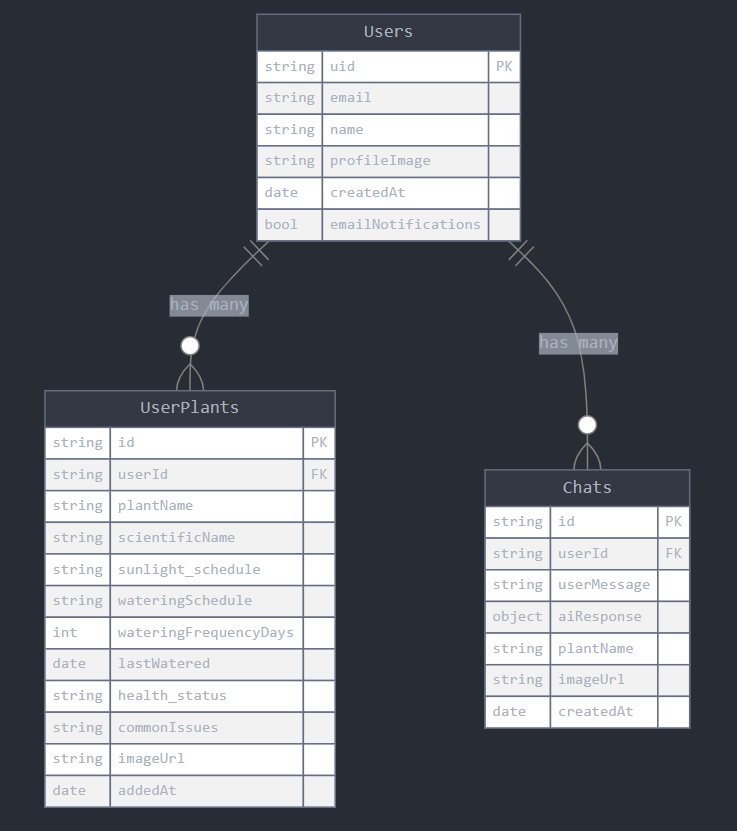
War-Nugget is Sahil Patel. luxnonbinarycode is MaShayla Kendrick. feifei925 is Feifei Xie.

# **System Architecture**

The system architecture consists of the frontend services, which were built with the Vue.js framework. The components of the frontend are the Login page, Register page, User Profile, Plant Gallery, Chat page, and the Pinia state management service. These client-side components speak with and interact with the Express.js backend through RESTful API calls. The Express backend consists of authentication routes, chat routes, Mutter, plant routes, user routes, CORS, and Redis caching. These backend technologies interface with external services such as Firebase, which provides user login and authentication services, Firestore, which is used for storing plant and user data, and Google Cloud, which is used for image references. Other external services the backend interfaces with are Vision, Gemini, and OAuth. When a user uploads a plant image, the backend sends that data to Vision, which identifies and analyzes that plant, and then Gemini is utilized to turn that information into a user-friendly message through Perenual API calls, populating the plant attributes with information before the message is returned to the frontend as an AI response. Another external service is the SendGrid service, which works with a Cloud function that checks watering schedules and then proceeds to email users reminding them of plants that need watering. Important structural utilities such as the logging service and notification service support the program as well



# **Database Schema**



# **Technology Stack & Rationale**

We used the Vue framework for proper and responsive frontend styling, Redis for cache management, Pinia for state management, Perenual API for a plant database, Firebase for user login and authentication management, Firestore for data storage, Google Vision for plant recognition, Gemini for message creation, and Express for general backend infrastructure. These tools were used since through trial and error, we determined that a system with these components works best to meet our business needs and fulfill our functional and non-functional requirements. These tools come together to optimize speed, accuracy, and data management that is perfect for our plant chatbot application.

# **Known issues, limitations, and unresolved features**

All known issues have been fixed. The product properly meets all of its requirements. Limitations include a lack of an offline mode and mobile app mode; however, these are not requirements for the app to work. One unresolved feature is a plant collage feature where you would visually view the plant’s progress in the form of timeline pictures.

# **Future Improvement Suggestions**

A collage feature where the user can view a timeline of pictures of their plant is a suggested feature that could just be added on as a bonus.

# **User Manual produced by Claude AI**

# **Verdure AI User Manual**

## **Table of Contents**

1. Introduction
2. Getting Started
   1. Creating an Account
   2. Logging In
   3. Resetting Your Password
3. Chat Interface
   1. Identifying Plants
   2. Asking Plant Care Questions
   3. Understanding AI Responses
4. Managing Your Plants
   1. Adding Plants to Your Collection
   2. Viewing Plant Details
   3. Editing Plant Information
   4. Watering Your Plants
   5. Deleting Plants
5. User Profile
   1. Viewing Your Profile
   2. Updating Profile Information
   3. Setting Notification Preferences
6. Plant Gallery
7. Troubleshooting

## **1. Introduction**

Verdure AI is an intelligent plant care assistant powered by artificial intelligence. The application helps you identify plants from photos, provides detailed care instructions, and helps you maintain your plant collection with personalized watering reminders.

Key Features:

* Plant identification from photos
* Detailed care instructions for your plants
* Watering schedule management
* Plant collection gallery
* Personalized plant care assistance

## **2. Getting Started**

### **Creating an Account**

1. Navigate to the Verdure AI landing page.
2. Click on the "Create Account" button.
3. Fill in the registration form with your:
   1. Name
   2. Email address
   3. Password (and confirmation)
4. Click "Register" to create your account.
5. Alternatively, you can click "Sign up with Google" to use your Google account.

### **Logging In**

1. Navigate to the login page.
2. Enter your email address and password.
3. Click "Login" to access your account.
4. Alternatively, click "Sign up with Google" to use your Google account.

### **Resetting Your Password**

1. On the login page, click "Forgot password?"
2. Enter your email address.
3. Click "Reset password" to receive password reset instructions in your email.
4. Follow the instructions in the email to create a new password.

## **3. Chat Interface**

The chat interface is your main tool for interacting with Verdure AI.

### **Identifying Plants**

1. Click on the paperclip icon at the bottom of the chat.
2. Select a plant image from your device.
3. Click "Send" to upload the image.
4. Verdure AI will analyze the image and provide plant identification results.

### **Asking Plant Care Questions**

1. Type your plant care question in the text field at the bottom of the chat.
2. Press "Enter" or click the send button to submit your question.
3. The AI will respond with helpful information about plant care.

### **Understanding AI Responses**

When Verdure AI identifies a plant, it provides:

* Plant Name (Common name)
* Scientific Name
* Sunlight Requirements
* Watering Schedule
* Soil Type
* Growth Habits
* Common Uses
* Potential Issues
* An interesting fact about the plant

For each identified plant, you'll see an "Add plant to collection" button to save the plant to your personal gallery.

## **4. Managing Your Plants**

### **Adding Plants to Your Collection**

**From Chat**:

1. After Verdure AI identifies a plant, click the "Add plant to collection" button.
2. The plant will be saved with all the information provided by the AI.

**Manual Addition**:

1. Go to your Plant Gallery page.
2. Click the "+ Add a new plant" button.
3. Fill in the plant information form:
   1. Plant Name
   2. Plant Type (scientific name)
   3. Sunlight Schedule (choose from dropdown)
   4. Watering Schedule (choose from dropdown)
   5. Last Watered Date
   6. Health Status
   7. Notes
   8. Plant Image (optional)
4. Click "Save Plant" to add it to your collection.

### **Viewing Plant Details**

1. Go to your Plant Gallery page.
2. Click on any plant card to view its details.
3. The expanded card shows:
   1. Plant image
   2. Plant name
   3. Type (scientific name)
   4. Sunlight requirements
   5. Watering schedule
   6. Last watered date
   7. Health status
   8. Notes
   9. Water level indicator

### **Editing Plant Information**

1. Open the plant details card by clicking on a plant in your gallery.
2. Click the "Edit" button at the bottom of the card.
3. Update the plant information in the form.
4. Click "Save Plant" to save your changes.

### **Watering Your Plants**

1. Open the plant details card by clicking on a plant in your gallery.
2. Click the "Water" button at the bottom of the card.
3. The last watered date will update to today, and the water level indicator will reset to 100%.

### **Deleting Plants**

1. Open the plant details card by clicking on a plant in your gallery.
2. Click the "Delete" button at the bottom of the card.
3. Confirm deletion in the popup dialog.

## **5. User Profile**

### **Viewing Your Profile**

1. Click on the account icon in the top right corner of any page.
2. Select "Account" from the dropdown menu.
3. Your profile page displays your personal information and account settings.

### **Updating Profile Information**

1. On your profile page, click the pencil icon next to any field you want to edit.
2. Make your changes to the information.
3. Click "Save Changes" at the bottom of the page.

### **Setting Notification Preferences**

1. On your profile page, find the "Notifications" section.
2. Check or uncheck the "Enable email notifications" option.
3. Click "Save Changes" to update your preferences.

When enabled, email notifications will alert you when your plants need watering based on their individual watering schedules.

## **6. Plant Gallery**

The Plant Gallery provides an overview of all your plants in an easy-to-browse format.

1. Access the Plant Gallery by selecting "Plant Gallery" from the account dropdown menu.
2. View all your plants as cards in a grid layout.
3. Each card shows:
   1. Plant image
   2. Plant name
   3. Click on a card to view detailed information about the plant.
4. Use the "+ Add a new plant" button to manually add new plants to your collection.

## **7. Troubleshooting**

**Problem**: Unable to log in

* Verify your email and password are correct
* Try resetting your password
* Check your internet connection

**Problem**: Plant identification is incorrect

* Ensure the image is clear and well-lit
* Focus on the plant's distinctive features (leaves, flowers)
* Try taking a new photo from a different angle

**Problem**: Water level indicator doesn't update

* Make sure you clicked the "Water" button (not just updated the date)
* Refresh the page to see the updated water level
* Check if your plant has a valid watering schedule set

**Problem**: Not receiving watering reminders

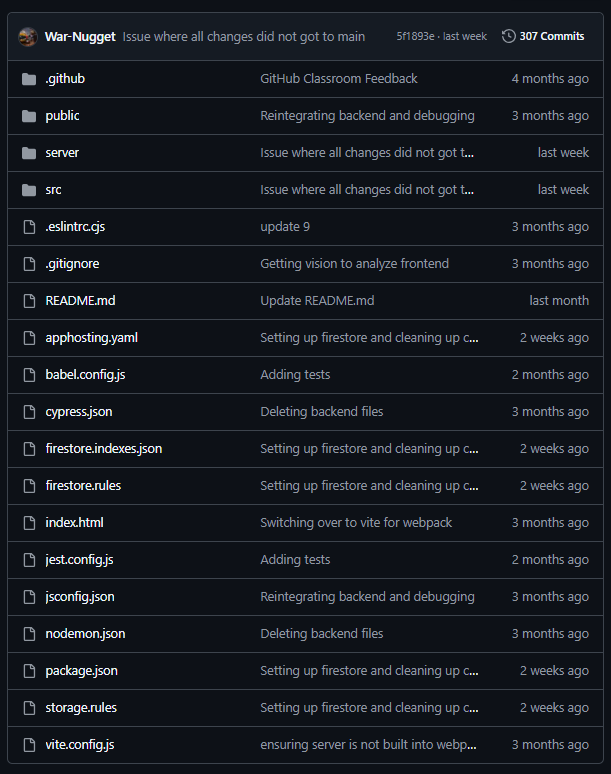
* Verify your email notifications are enabled in your profile
* Check your spam/junk folder
* Ensure your plants have proper watering schedules set

**Problem**: Image upload fails

* Check that your image is in a supported format (JPG, PNG)
* Ensure the image file size is not too large
* Check your internet connection and try again

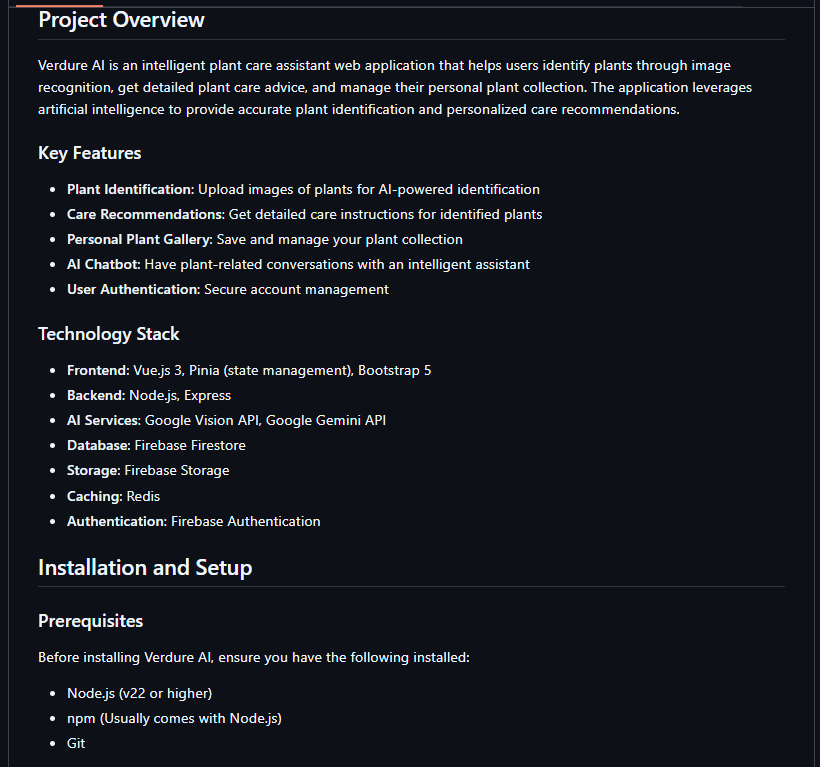
# **GitHub Practices**

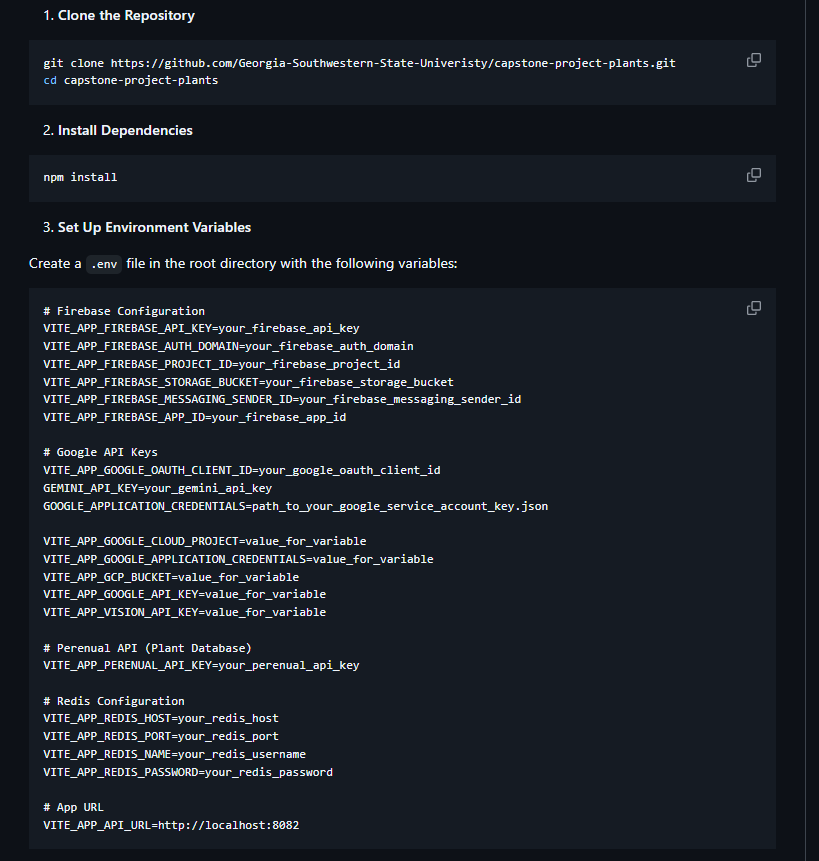
## **Final Commit to main branch**

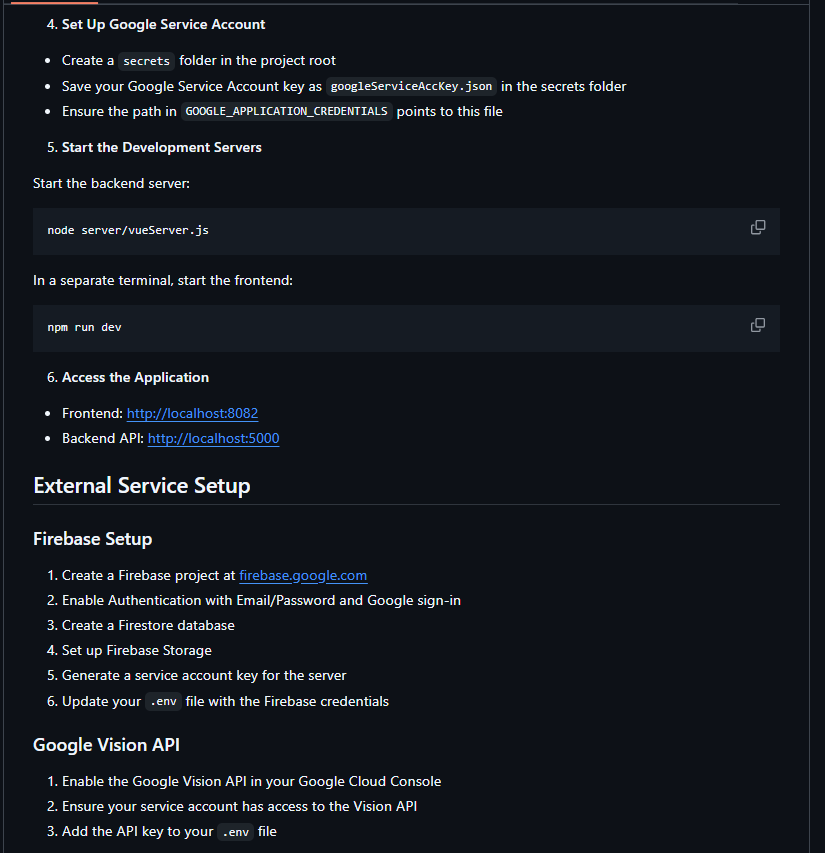


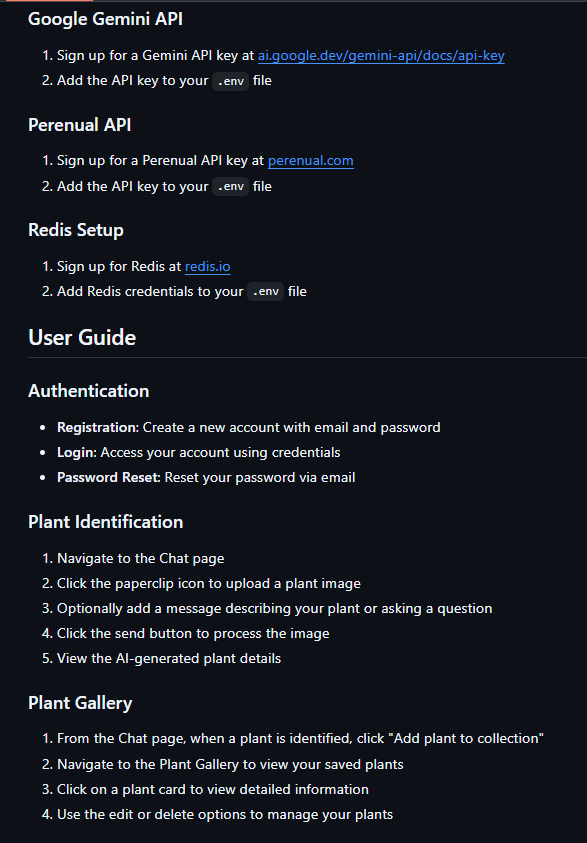
Our final commit was last week where we were trying to transfer important changes to main.

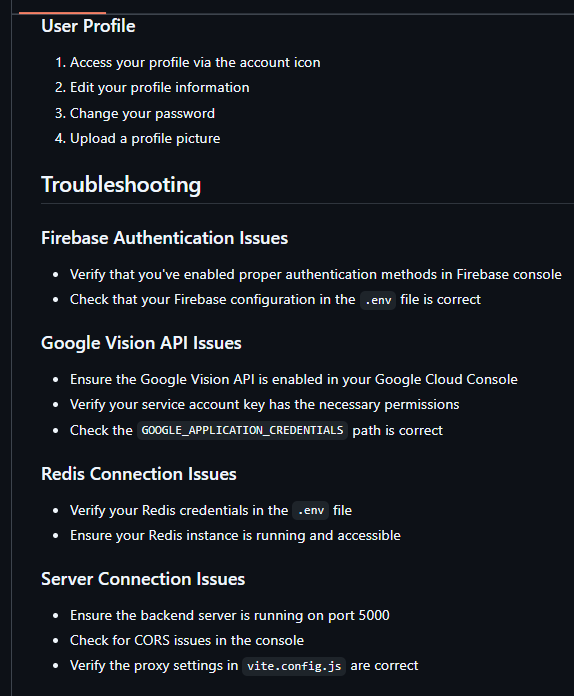
# **README.md**

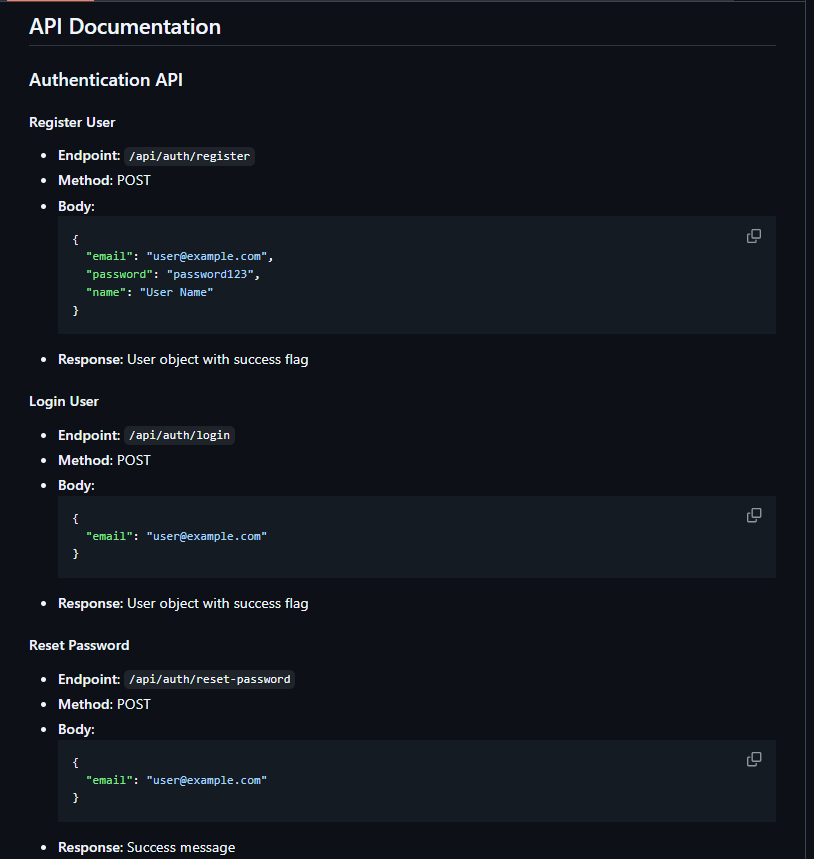


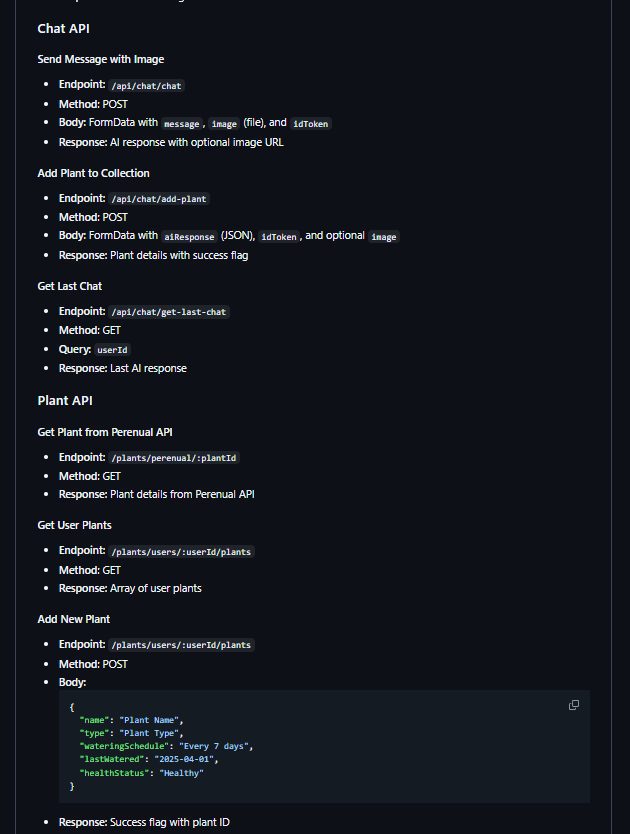


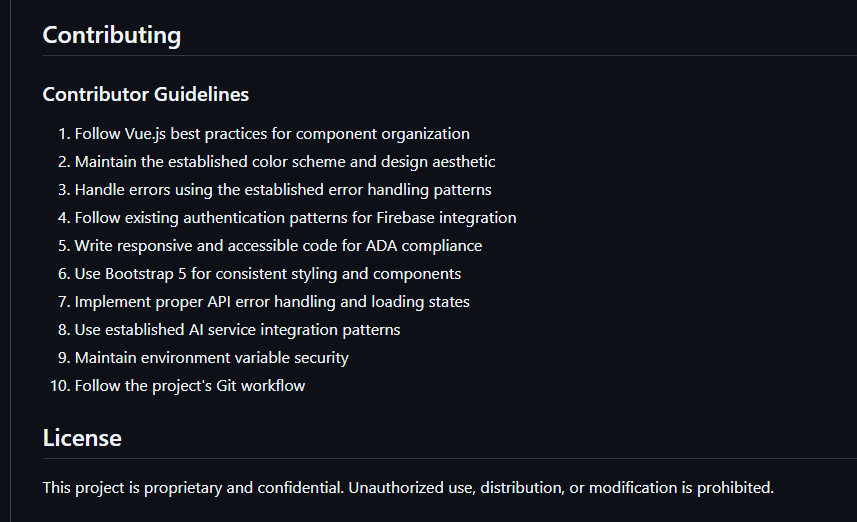






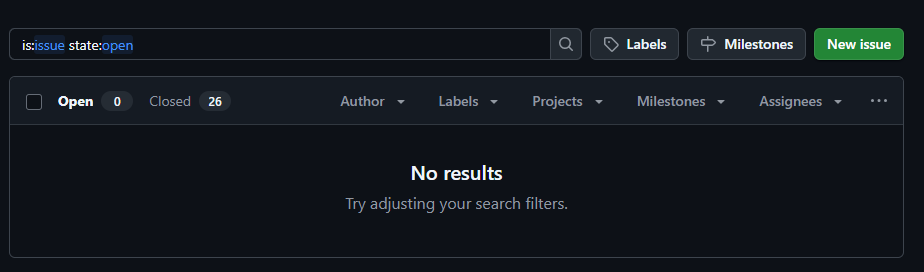




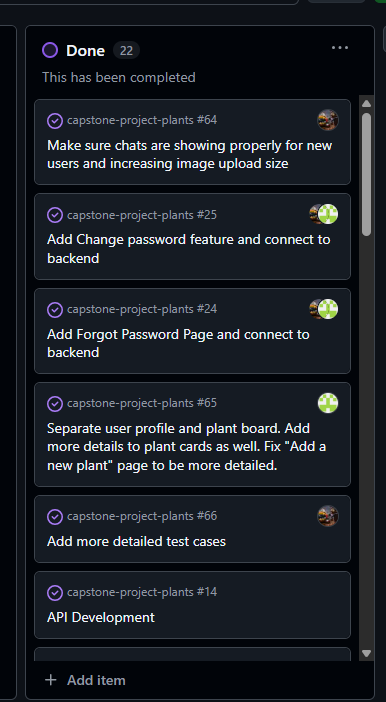


# **Issues and Project Board**

## **All issues closed**



## **Project Board Finalized**



# **Jest Testing and clean, commented codebase**

The Jest Testing Framework is included in the codebase. All code is commented and efficient.