# Plant Pal: A Personalized Plant Care Companion

## Overview

The "Plant Pal" web application aims to empower plant owners by providing a centralized, intuitive platform for managing their plant collections and ensuring optimal care. The primary problem it addresses is the common challenge of **remembering diverse care requirements** (watering, light, feeding) for multiple plants, often leading to neglect, overwatering, or general plant distress. My motivation for this project stems from a personal love for plants and witnessing firsthand the struggles many enthusiasts face in keeping their green companions thriving. I believe a dedicated tool can significantly enhance the plant ownership experience, fostering healthier plants and more confident plant parents.

## Target Audience

Plant Pal is designed for **home plant owners of all experience levels**, from novice gardeners with a few houseplants to seasoned collectors with extensive indoor jungles. It will particularly serve individuals who:

Desire an **organized system** to track their plant collection.

Need **reliable reminders** for watering, fertilizing, and other care tasks.

Seek to **understand the specific needs** of different plant species.

Wish to **monitor the growth and health** of their plants over time.

By offering personalized insights and proactive reminders, the application will help users maintain thriving plants, reduce the guesswork in plant care, and foster a deeper connection with their botanical companions.

## Major Functions

Here are 8 core functions of the Plant Pal application:

**User Registration and Profile Management:** Securely create accounts, log in, and manage personal profile details (e.g., username, email, avatar).

**Add/Edit Plant Profile:** Users can add new plants to their digital collection, specifying details like name, species, acquisition date, and a photo. They can also modify existing plant profiles.

**Personalized Care Reminders:** Set custom, recurring reminders for watering, fertilizing, repotting, and other specific care tasks for each individual plant. The application will provide in-app notifications.

**Plant Information Database:** Utilize an external API to fetch detailed care information (e.g., light, water, humidity, soil type, toxicity) for various plant species, linking this data to the user's added plants.

**Care History Log:** Record and view a historical timeline of care activities performed for each plant, offering insights into past routines.

**Progress Tracking & Photo Journal:** Upload and view timestamped photos of plants to visually track growth, changes, or recovery over time, presented in a chronological journal.

**"Needs Attention" Dashboard:** A prominent section on the main dashboard highlighting plants with upcoming or overdue care tasks, ensuring timely intervention.

**Plant Search and Discovery:** Explore a vast database of plant species, filtering by criteria like light requirements, care difficulty, or pet-friendliness to find new additions.

## Wireframes

**(As an AI, I cannot create visual images. Below are detailed textual descriptions for conceptual wireframes.)**

### Mobile View Wireframe: "My Plants" List

+--------------------------------------------------+

| ☰ Plant Pal 👤 | <- Header

+--------------------------------------------------+

| [ + Add New Plant ] | <- Full-width CTA button

+--------------------------------------------------+

| ⚠️ 2 plants need watering today! | <- Needs Attention Banner

+--------------------------------------------------+

| 🪴 [IMG] Fernie the Fern | <- Plant Card

| Boston Fern |

| Status: Water in 2 days |

| [⋮] or [>]|

+--------------------------------------------------+

| 🪴 [IMG] Lily the Peace Lily |

| Peace Lily |

| Status: Overdue! |

| [⋮] or [>]|

+--------------------------------------------------+

| 🪴 [IMG] Cactus Jack |

| Cactus |

| Status: Healthy |

| [⋮] or [>]|

+--------------------------------------------------+

+--------------------------------------------------+

| 🏡 My Plants 🔍 Discover 📘 Care Log | <- Bottom Nav (fixed)

+--------------------------------------------------+

### Desktop View Wireframe: Main Dashboard

## +--------------------------------------------------------------------------------+

## | Plant Pal My Plants | Discover | Care Log | Profile |

## +--------------------------------------------------------------------------------+

## | 🧍‍♂️ Avatar Username | Welcome back, [Username]! |

## | [Edit Profile] | Here’s a look at your leafy friends. |

## |---------------------------------------+----------------------------------------|

## | 📅 Upcoming Reminders | 🪴 Fernie the Fern 🪴 Lily |

## | ------------------------------------- | Boston Fern Peace Lily |

## | 🌿 Fernie - Water - Jul 17 | [Large IMG] [Large IMG] |

## | 🌵 Jack - Prune - Jul 18 | Last Watered: Jul 10 Last: Jul 12 |

## | | Next Water: Jul 20 Overdue! |

## | Quick Actions: | [🟢 Healthy Icon] [🟡 Caution] |

## | [ + Add Plant ] [ 🔍 Search ] | [ View Details ] [ View Details ]|

## | |----------------------------------------|

## | | Recently Logged Care: |

## | | ✓ Watered Fernie on Jul 14 |

## | | ✓ Fertilized Lily on Jul 13 |

## +--------------------------------------------------------------------------------+

## External Data

I will utilize the following external API for comprehensive plant information:

**Trefle API (trefle.io)**: This API provides an extensive botanical database, crucial for enriching the **Plant Information Database** feature.

**Data Attributes Needed from Trefle:**

scientific\_name, common\_name: For accurate plant identification.

image\_url: High-quality images of the plant species.

growth\_habit, growth\_form: General growth characteristics.

light: Optimal light conditions (e.g., "Full sun," "Partial shade").

watering: General watering needs (e.g., "Frequent," "Moderate," "Low").

soil: Preferred soil types.

temperature\_range: Ideal temperature range.

toxicity: Information on toxicity to humans/pets.

propagation\_methods: Common ways to propagate the plant.

For data storage, I will use **Firebase Firestore**. The following data will be stored:

**User Profiles:** userId, email, username, profilePictureUrl.

**User Plants:** plantId (unique), userId (FK), userGivenName, speciesScientificName (links to Trefle data), acquisitionDate, lastWatered, nextWaterDate, lastFertilized, nextFertilizeDate, customCareNotes, userUploadedPhotoUrls (array).

**Care Logs:** logId (unique), plantId (FK), activityType (e.g., "water", "fertilize", "repot"), activityDate, notes.

**Reminders:** reminderId (unique), plantId (FK), type (e.g., "water", "fertilize"), frequency (e.g., "every 7 days"), nextDueDate, isActive.

## Module List

The Plant Pal application will be structured into the following key modules and files:

**HTML Files:**

index.html: The main entry point, including login/registration and routing logic.

dashboard.html: The user's main hub displaying their plants and reminders.

plant-details.html: Dedicated page for individual plant profiles and care history.

add-edit-plant.html: Form for adding new plants or modifying existing ones.

discover-plants.html: Page for searching and exploring the plant database.

profile-settings.html: User profile management.

**CSS Files:**

main.css: Global styles, typography, and foundational layout.

components.css: Styles for reusable UI elements (buttons, cards, forms).

dashboard.css: Specific layout and styling for the main user dashboard.

responsive.css: Media queries for mobile, tablet, and desktop adaptations.

**JavaScript Files:**

app.js: Main application entry, handles routing, initializes Firebase, and orchestrates modules.

auth.js: Manages user authentication (login, registration, logout, user session).

plantManager.js: Handles CRUD operations for user's plants with Firestore.

apiService.js: Encapsulates all interactions with the Trefle API.

reminderScheduler.js: Logic for calculating and managing care reminders based on user input and history.

uiRenderer.js: Responsible for dynamically updating the DOM based on application state and data.

validation.js: Utility functions for input validation in forms.

utils.js: General helper functions (e.g., date formatting, data transformation).

## Graphic Identity

**Color Scheme:**

**Primary Accent:** #4CAF50 (A vibrant, fresh green, symbolizing growth and health).

**Secondary Accent:** #8BC34A (A lighter, softer green, for complementary elements and subtle highlights).

**Neutral Text:** #333333 (Deep charcoal for excellent readability on all backgrounds).

**Backgrounds:** #F8F8F8 (Light off-white for a clean, airy feel).

**Alert/Highlight:** #FFC107 (A warm yellow, used sparingly for urgent reminders or calls to action).

**Typography:**

**Headings:** **'Montserrat'** (Sans-serif, bold and modern, conveying clarity and approachability).

**Body Text:** **'Open Sans'** (Sans-serif, highly legible and friendly, ensuring comfortable reading of plant care details).

**Fallback:** sans-serif for both.

**Specific Element Styling Details:**

**Buttons:** Softly rounded corners, subtle shadows on hover, promoting an organic feel.

**Plant Cards:** Clean, minimalist design with a subtle border and shadow, allowing plant images to stand out.

**Input Fields:** Clean, understated design with a clear focus state (e.g., a subtle green border on focus).

**Icons:** Simple, line-art style icons for clear visual communication.

**Application Icon Description:** A stylized **leaf icon** in the primary green (#4CAF50), with a subtle, abstract **water droplet** shape (#2196F3 – a light blue) nested within its outline. This combines the core themes of plants and care, creating a fresh, recognizable, and inviting symbol for the application.

## Timeline

### Week 5: Foundational Setup & API Integration

**Deliverables:**

**Initial HTML Structure:** index.html, dashboard.html, add-edit-plant.html with basic layouts.

**Core CSS:** main.css and components.css with global styles and component foundations.

**Firebase Authentication:** Basic user registration and login functionality implemented using auth.js.

**Trefle API Integration (Basic):** apiService.js can successfully fetch and display plant names/images from Trefle based on a search query.

**Initial Trello Board:** All known tasks defined as cards.

### Week 6: Plant Management & Basic Reminders

**Deliverables:**

**Add/Edit Plant Functionality:** User can add new plants to their collection via add-edit-plant.html, saving basic details to Firestore via plantManager.js.

**Display User's Plants:** dashboard.html dynamically displays the user's added plants from Firestore.

**Basic Care Reminders:** Users can set simple watering frequencies for plants; reminderScheduler.js calculates next due date, and these are displayed on the dashboard.

**Care History Logging (Basic):** User can manually log a "watered" event for a plant.

**Graphic Identity Application:** Begin applying defined color scheme and typography across core pages.

### Week 7: Enhanced Features & Polish

**Deliverables:**

**Detailed Plant Profiles:** plant-details.html displays comprehensive Trefle API data integrated with user-specific plant information.

**"Needs Attention" Dashboard:** Logic implemented to dynamically show overdue/upcoming tasks.

**Progress Tracking:** Users can upload (client-side only for demo) and view photos of their plants in a simple journal.

**Responsive Design Refinement:** Ensure all major views are optimized for mobile and desktop using responsive.css.

**Code Quality:** Refactor JavaScript code for modularity and maintainability; implement robust error handling and user feedback.

**Final Styling:** Polish UI elements, ensuring consistency with the defined graphic identity.

## Project Planning (Trello Board)

(A live link to the Trello board will be provided here once it's set up: **[Trello Board Link Placeholder]**)