My Garden App

An application developed by Rachel Warner

Introduction

My Garden App is a helpful tool designed to simplify plant care and garden management. It allows users to identify plants, access tailored care information, and receive timely notifications for essential tasks like watering, pruning, and fertilizing. Whether you're growing a few houseplants or maintaining a full garden, this app helps you stay organized and informed, so your plants thrive with ease.

Features

The application features tools such as:

My Garden

My garden is a portion of the application that stores the plants you identify in your garden

- A local SQLite database that stores the plants you identify in your garden.

Identification History

- Get a history of all the plants you have identified.

Care notifications:

- Get notifications for caring for your plants, such as:

Essentials:

  - Watering
  - Pruning
  - Sunlight Adjustment: (moving plants indoors/outdoors based on season, light levels, outside temperatures).
  - Fertilizing
    - Notification based on the plant's growth cycle (e.g. once a month during growing season).
  - Repotting
    - Every 6-12 months for some indoor plants.
    - Notifying based on plant age or user log.
  - Seasonal Changes
    - Bring plants indoors before frost
    - Bring plants in based on temperature outside and hardniess
    - Dormancy reminders (stop watering or fertilizing in the winter)

- Vermicomposting:
    - Reminders to feed worms

Advanced:

- Rotation:
    - For houseplants to grow evenly (every 2-4 weeks).
- Pest Check:
    - Monthly reminder to inspect leaves/soil
- Germination or Propagation updates
    - If the user logs propagation, send reminders to check the water roots or transfer to the soil
- Soil check:
    - Check pH, nutrients, and moisture
- Weather-driven:
    - Don’t water today—rain expected.
    - Water more—heatwave this week.
- Custom User Tasks:
    - The user can add their own custom tasks for reminders, such as: “Spray for fungus”

Backend

The backend of the application is built with **Python** and powered by **FastAPI**, a modern, high-performance web framework designed for building APIs quickly and efficiently. It exposes a series of RESTful endpoints that support core functionality such as searching for plants, managing a personal garden, and logging care tasks.

To minimize external dependencies and improve performance, the backend uses a local **SQLite** database to **cache plant data** retrieved from the Perenual API. This caching strategy reduces redundant API calls and improves response times, especially for frequently accessed plant records. When a user requests data that is not yet stored locally, the backend fetches it from the Perenual API, stores it in the database, and returns the response.

This approach balances the flexibility of a live third-party API with the speed and reliability of local data storage.

Database Architecture: