Project Proposal - PlantPal

1. Team Information

Team Name: SproutLab	
Student Number	**Name**
2352280	Hengyu Jin
2352609	Qi Lin
2353409	Baoyi Hu
2352288	Sirui Da

2. Project Description

2.1 Background and Motivation

With fast-paced urban life, many people turn to plant care for relaxation and home improvement. However, beginners often face difficulties due to limited knowledge and inconsistent maintenance. Existing plant care apps mainly provide reminders or static information, lacking emotional engagement and community interaction.

PlantPal aims to make plant care smarter, more practical, and emotionally engaging. It combines AI recognition, a basic care knowledge base, AR interaction, and community sharing to deliver both scientific guidance and emotional companionship.

- **Social significance**: Encourages eco-friendly living and emotional well-being through plant interaction.
- **Technical innovations**:
  a. AI vision: Recognizes plant species and health conditions via camera input.
  b. Knowledge base: Offers beginner-friendly tips and practical care guides.
  c. AR interaction: Adds live plant expressions for engaging feedback.
  d. Community forum: Supports sharing, help, and inspiration.
- **Learning value**: Integrates AI, knowledge design, UX, and social interaction into a cohesive learning project.

2.2 Main Goals

PlantPal aims to build an all-in-one intelligent platform integrating AI recognition, personalized care, interactive gamification, and community engagement.

Key tasks:

a. Plant profile & recognition module – Identify plants via photo and auto-generate profiles.
b. Smart care reminder system – Auto-schedule watering, fertilizing, and repotting tasks.
c. AI & AR interaction system – Show real-time facial expressions and growth animations on screen.
d. Knowledge & problem solver database – Diagnose issues and provide treatment suggestions.
e. Community forum – Enable users to post, follow topics, and share plant care experiences globally.

2.3 Intended Users and Key Usability Goals

- **Target Users:** Plant beginners, family users, urban workers, and plant enthusiasts.
- **User Experience Improvements:** Combines AI recognition, care guidance, and community interaction to make plant care easy, social, and emotionally engaging.
- **Key Usability Goals:** Ease, Accessibility, Interactivity, Responsiveness

3. Competitive Product Analysis

Product	Utility	Technical Advantages	Limitations & Disadvantages
Planta	A smart plant care app offering watering and fertilizing reminders and health diagnostics.	Accurate data and reliable scheduling.	Limited interactivity and no community features.
PictureThis	Identifies plants and provides care suggestions.	High recognition accuracy and quick results.	Mainly focused on identification, lacks personalization.
Bloomscape	Combines plant management with e-commerce.	Integrated with Bloomscape ecosystem.	Sales-oriented, limited innovation and engagement.

Most plant care apps provide good recognition and reminders but lack emotional interaction and community features. PlantPal stands out by combining AI recognition, AR feedback, a care knowledge base, and community sharing, creating a more personalized and engaging plant care experience.

4. Main Functionality and Characteristics

4.1 Core Functional Modules

- **Plant Profiles & Recognition:** One-tap profiles with photo ID (Top-k + confidence) and auto care/pest timeline.
- **Smart Care Reminders:** Dynamic, explainable schedules (plant × weather × season × history) that learn from feedback.
- **AI & AR Interaction:** Q-style avatar and first-person "plant" chat with light AR overlays.
- **Knowledge Base & Problem Solver:** Rulebase v1 (symptoms → causes → actions → prevention) plus image-to-case search and beginner checklists.
- **Community Forum:** Topic-tagged posts with structured templates and curated answers/trending.

4.2 System Architecture

- **Client (App):** Cross-platform (Flutter/RN), SQLite + media cache, camera/album, basic AR, push notifications, offline-first with auto-sync; on-device fast recognition, cloud fallback for long-tail.
- **Cloud (Backend):** API layer (auth/rate-limit) → services (users/plants/events/forum); rules engine + scheduler (weather/season/profile → tasks); AI services (cloud recognition, symptom classify, vector search, hot updates); data layer (PostgreSQL + object storage + vector DB); integrations (weather, push, content safety); ops console.

4.3 Key Characteristics

- **Cross-platform & offline-first** (single codebase, auto-sync)
- **Low-latency** (local <200 ms; on-device <1.5 s; cloud <3 s targets)
- **Intelligent & adaptive** (species × weather × season × history)
- **Explainable & user-tunable** (rationale + intensity; one-tap adjust)
- **Privacy-first** (on-device inference, granular permissions, export, optional cloud)

5. Novelty and Enhancements

5.1 Innovation Highlights

- **Plant's-eye AI:** Emotive first-person tips make care advice simple and relatable.
- **Weather-based scheduling:** Auto-adjusts watering and fertilizing by species, weather, and season.
- **Explainable reminders:** Shows reason and strength level (light/normal/deep) for easy adjustment.
- **Edge–cloud recognition:** On-device AI for speed; cloud fallback for accuracy.
- **Image-to-case learning:** Photo search for similar cases and best treatment examples.
- **Privacy-first design:** Local inference, optional sync, and portable user data.

5.2 Comparative Advantages

Dimension	Common Approach	Our Improvement
Scheduling	Fixed frequency	Dynamic, adaptive frequency/dosage
Reminders	No explanation	Explainable + one-tap tuning
Recognition	Cloud-only, slow	Edge–cloud hybrid: fast on-device, accurate in cloud
Knowledge	Pure popular-science content	Stepwise plan + image-to-case retrieval
Community	Unstructured posting	Structured help template → faster, more accurate answers
Privacy	Full cloud upload	Edge-first, minimal upload, exportable

5.3 Technical Approach & Design Principles

- **Three-layer engine:** rules (species/pot/medium/light) → heuristics (weather/season; ET_0 → moisture risk) → lightweight learner (adapts via on-time/delayed/skipped).
- **Edge–cloud synergy:** on-device for common classes, cloud for long-tail/complex; vector search for image-to-case.
- **Explainable & testable:** human-readable reminder logic + key factors; continuous **A/B** on strategies.
- **Privacy & scalability:** offline-first, optional sync; **hot updates** for rules/species/models; modular rollout.
- **Gamified feedback loop:** growth points/animations as a "feedback API" across tasks, recognition, and fixes.

6. Team Organization and Preliminary Planning

6.1 Roles & Responsibilities

Role	Owner	Key Responsibilities	Core Deliverables
PM & Ops (Product+QA)	Hengyu Jin	Scope/roadmap, KPIs, user research, release & QA	PRD & prototype, test checklist, weekly reports
Mobile Lead (App)	Qi Lin	Flutter/RN app, camera/gallery, offline & notifications, basic AR	Running app, timeline/reminders, export/offline
Backend & Rules (DevOps)	Baoyi Hu	APIs, scheduler/rules, weather/storage, CI/CD	REST APIs, scheduler, admin console
CV/ML & Data	Sirui Da	On-device (common), cloud long-tail, image-to-case, analytics	Inference services, Top-k results, KPI dashboard

6.2 Timeline

W1–2 · Research & Planning: Define goals, complete user and competitor research, finalize PRD v1 and tech stack.

W3–4 · Prototype: Build a runnable prototype with core app, backend, and basic rule engine.

W5–6 · Testing & Refinement: Add smart reminders, improve UX through pilot testing, prepare Beta version.

W7–8 · Beta Launch: Enable image search, forum, admin tools, and cloud sync; release Beta.

W9–10 · Final Release: Optimize performance, conduct A/B tests, finalize and release v1.0.

7. Engineering Process and Methodologies

- **Methodology:** Agile + Design Thinking → flexible, user-centered.
- **Iteration:** 2-week sprints: plan → design → code → test → review.
- **User Insights:** Interviews + empathy mapping → user stories & prototypes.
- **Workflow:** Feature branches + CI + unit/integration/acceptance tests.
- **Improvement:** Build–Measure–Learn loop.

8. Team Collaboration Platforms

Efficient teamwork is achieved through an integrated digital workflow:

- **GitHub** for code hosting, version control, and task management.
- **Figma** for collaborative UI/UX design and a shared **component library**.
- **Feishu (Lark)** for communication, video meetings, and synchronized updates.
- **Feishu Docs & Drive** for centralized documentation, real-time editing, and secure cloud storage.

9. Potential for Further Development

PlantPal is designed with modular scalability and strong future potential:

- **Smart Hardware Integration**: Connects with soil sensors or smart pots for automated care.
- **E-commerce Expansion**: In-app marketplace for plants and tools via brand partnerships.
- **Enterprise Solutions**: Smart “green space” management for offices and hotels.
- **Community Growth**: Expert Q&A, courses, and plant contests to enhance engagement.
- **Research Potential**: Using time-series and multimodal AI for plant health prediction and precision agriculture.

10. Related Technologies

- **Platform:** Mobile (Android / iOS)
- **Languages:** Dart (Flutter), Python or Swift
- **Frameworks/Libraries:** TensorFlow Lite, FastAPI, PostgreSQL
- **Tools:** VS Code, Git, Docker

11. Challenges and Risks

- **Recognition accuracy:** Diverse and similar plant species require large labeled datasets and continuous model updates.
- **AR performance:** Ensuring smooth tracking and rendering on mobile devices with limited resources.
- **Dynamic scheduling:** Building adaptive care plans using weather, season, and user data.
- **Multilingual & offline:** Supporting offline recognition and multi-language reminders.

Risk Mitigation: Develop an MVP focusing on core features; adopt modular architecture for phased updates; conduct regular testing and risk reviews; promote team training and knowledge sharing.

12. Professional Growth Reflection

- **Cross-disciplinary skills:** Gain experience in computer vision, deep learning, AR, mobile development, and backend architecture.
- **Product thinking:** Learn to design valuable products through user research, data analysis, and A/B testing.
- **Engineering practice:** Apply Agile, CI/CD, and version control to improve teamwork and software quality.
- **Business insight:** Explore ecosystem partnerships and business models for innovation and sustainability.
- **Personal growth:** Develop expertise in AI, front-end, backend, or operations while improving communication and documentation skills.

13. Project-Related Resources

- Plant Care App: Uses TensorFlow Lite and Gemini API to build an AI-driven plant care app with AR and community features.
  https://rjpn.org/jetnr/papers/JETNR2502017.pdf
- Plant Care App with AR: Combines AR and image recognition for instant guidance, with potential for community and smart hardware expansion.
  https://www.oasis-of-ideas.com/ideas/plant-care-app-with-ar-identification-and-guidance
- AI-Generated Mockup: