🌱 Plantify Project Documentation

---

1. Project Planning

Objective:

Develop a **machine learning–powered web app** that predicts the most suitable crops for a given soil based on **soil composition (N, P, K, pH)** and **environmental parameters (temperature, humidity, rainfall)**.
The system also integrates with **IoT sensors** to monitor water and nutrient levels in real time.

Scope:

- Machine learning model for crop prediction
- Data visualization and analysis tools
- Chat assistant (AI-based) for data exploration and recommendations
- Integration with IoT sensor data

Main Deliverables:

1. Cleaned and processed dataset for training
2. ML model for top 5 crop predictions
3. Web dashboard with 4 main sections
4. Database for storing user, soil, and environmental data
5. Documentation and user guide

Phases & Timeline (Example):

Phase	Description	Duration
1. Data Collection	Gather soil & climate data from open datasets and universities	1 weeks
2. Data Preprocessing	Clean, analyze, and visualize data	1 weeks
3. Model Development	Build and train ML model for crop prediction	1 weeks
4. Web App Development	Build front-end(Streamlit) , and integrate ML model	2 weeks
5. Testing & IoT Integration	Test predictions and sensor data flow	2 weeks
6. Deployment	Host app and create documentation	1 week

---

2. Stakeholder Analysis

Stakeholder	Role / Interest	Benefits	Engagement Method
Farmers	End users; use the system to know best crops for their land	Higher yield, efficient fertilizer use	web interface, training workshops
Agricultural Engineers	Analyze data, advise farmers	Decision support, advanced insights	Dashboard, AI chat
Government	Policymaking, data collection	National productivity, sustainable agriculture	Reports, shared data access
Compound Owners / Businessmen	Use system for agricultural planning	Cost-efficient land use	Web access, consultation
Universities / Labs	Provide data and validate results	Research collaboration	API or data-sharing portal

Stakeholder	Role / Interest	Benefits	Engagement Method
Developers	Build and maintain system	Professional experience, research contribution	Development reports, meetings

---

3. Database Design

1. Dataset Completeness and Size:
    - The dataset contains 66,000 entries and 8 columns, with no missing values across any column. This means the data is clean and ready for analysis without needing imputation.
2. Feature Types:
    - There are 7 numerical features (N, P, K, temperature, humidity, ph, rainfall) which describe soil nutrients and environmental conditions.
    - There is 1 categorical feature, label, which is likely the target variable representing different crop types.
3. Variability in Nutrient Levels (N, P, K):
    - Potassium (K) shows the highest variability (standard deviation of 151.68) and the widest range (10 to 900), with a strong right skew (mean 125.54 vs. median 61.07). This suggests that potassium requirements can differ drastically across different crop types.
    - Nitrogen (N) also exhibits high variability (std of 107.58) and a broad range (0 to 600).
    - Phosphorus (P) has a more moderate range and variability compared to N and K.

4. Variability in Environmental Factors:
    - Rainfall has a significant range (200 to 3000 mm) and high variability (std of 592.63), along with a right skew (mean 1064.74 vs. median 859.39), indicating diverse water needs for different crops.
    - Humidity has a wide range (0 to 100%). The minimum value of 0% could be an outlier or represent specific, very dry conditions, and might warrant closer inspection if it's considered unphysical for relevant conditions.
    - Temperature and pH are relatively stable and well-distributed, with their means closely matching their medians, suggesting typical agricultural ranges.
5. Implications for Analysis:
    - The dataset is well-structured for a classification task, where you would predict the label (crop type) based on the 7 numerical features.
    - The high variability observed in N, K, and rainfall suggests these features will likely be strong predictors and play a crucial role in distinguishing between different crop requirements.

---

4. UI/UX Design

Design Overview

Plantify's user interface is built using **Streamlit**, providing an interactive and responsive dashboard experience for users ranging from farmers to data scientists.
The design emphasizes **simplicity, clarity, and data interactivity**, combining visual analysis, intelligent chat, and machine learning recommendations in one unified platform.

Design Principles

- **Clean Dashboard Layout:** Uses Streamlit's sidebar navigation for intuitive tool selection.
- **Consistent Color Scheme:** Green tones and natural hues to reflect agricultural context.
- **Wide Layout:** Optimized for desktop viewing (`layout="wide"`) to support data visualization and detailed insights.
- **Minimalist Icons & Emojis:** Improve user engagement and guide navigation visually.

- **Responsive Feedback:** Uses Streamlit components like st.spinner(), st.success(), st.warning(), and st.error() for user-friendly communication.

Application Structure

Sidebar Navigation

- Displays the **Plantify logo/title** and description.
- Allows users to switch between modes:
    1. **Crop Recommendation** (default)
    2. **Chat with AI**
    3. **Data Overview**
    4. **Plotting**
- Includes **CSV upload section** to import datasets for analysis.
- Gives instant feedback on upload status.

Main Pages and Functional Flow

1. Crop Recommendation

- **Purpose:** Predict the most suitable crops based on user-provided soil and climate parameters.
- **Features:**
    - Two-column input layout for **Soil Parameters** and **Climate Conditions**.
    - Dynamic parameter inputs for:
        - Nitrogen (N), Phosphorus (P), Potassium (K)
        - pH, Temperature, Humidity, Rainfall
    - **Top-k crop selection** (3, 5, or 10 crops).
    - Results displayed with:
        - Ranked crop predictions and probabilities.
        - Ideal condition comparison tables.
        - Interactive difference bar charts (Matplotlib).
    - Includes **What-if Analysis** to compare user input vs. ideal conditions.
    - **Feedback Section** for user satisfaction.
    - **Model Retraining Simulation** (educational feature to demonstrate ML adaptability).

---

2. Chat with AI

- **Purpose:** Provide an AI assistant for dataset understanding and exploratory data analysis.
- **Implementation:** Integrated with **Google Gemini API**.
-
- **Functionality:**
  - Interactive chat history stored in session state.
  - Users can ask natural-language questions like:

    "What are the key correlations in my dataset?"

  - Gemini responds with analytical explanations or ready-to-use Python code snippets.

---

3. Data Overview

- **Purpose:** Offer quick insights into the uploaded dataset.
- **Features:**
  - Dataset preview
  - Descriptive statistics
  - Data information summary
- **UI Elements:** Simple text and table outputs for readability and transparency.

---

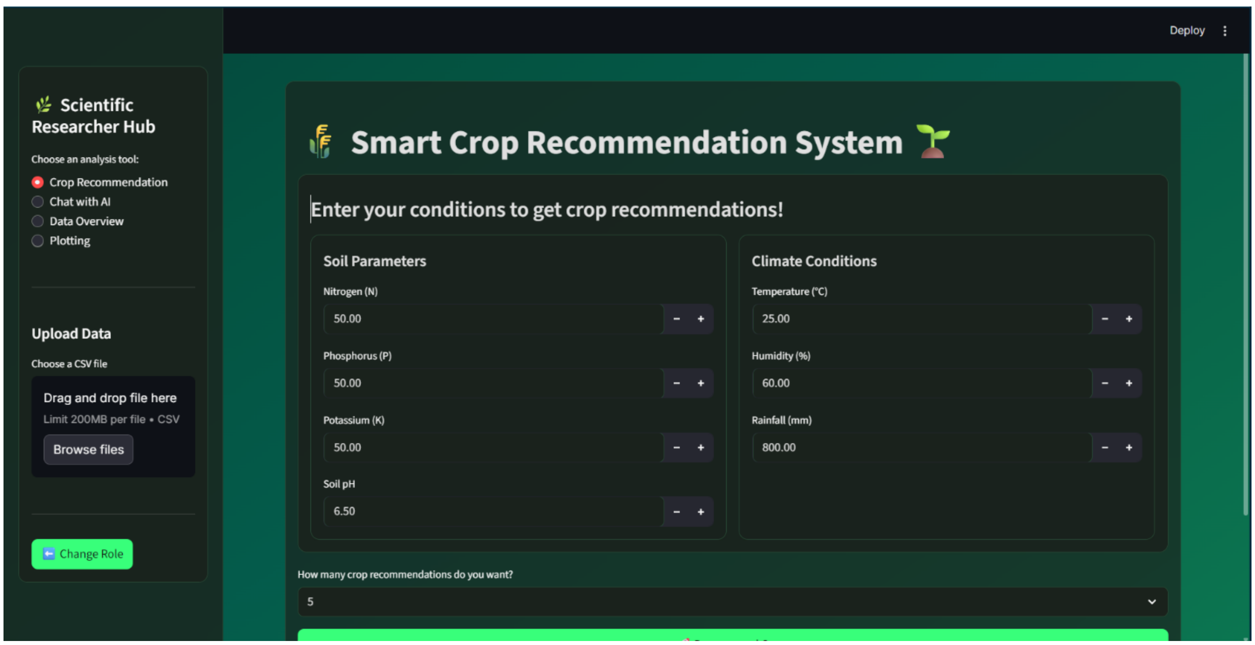
4. Plotting

- **Purpose:** Provide interactive data visualization.
- **Powered by:** Plotly Express for high-quality, responsive charts.
- **Plot Types Supported:**
  - Histogram
  - Box Plot
  - Scatter Plot
  - Bar Chart
- **Dynamic Selection:**
  - Users choose variables (columns) for X/Y axes.
  - Charts automatically update and render inline.

UX Considerations

- **Guided Interaction:** Each section includes tooltips, subheaders, and examples to guide non-technical users.
- **Instant Results:** Spinner animations and alerts maintain engagement during processing.
- **Error Handling:** Clear messages for missing files, incorrect API keys, or data format issues.
- **Session Persistence:** Uses (st.session_state) to store uploaded data and chat messages across interactions.

Streamlit Interface Design



Optional Future Add-ons:

- Integrate **real-time IoT dashboards** showing soil sensor data updates.
- Add **dark/light mode toggle** for better accessibility.
- Provide **multi-language support** (English/Arabic toggle).
- Enable **export of visualizations** and AI chat transcripts.