CA - Prerequisites Document

Name of the Student	**Riya Varyani**
Class and Roll No	**D15A 61**
DOP	
DOS	
Sign and Grade	

TITLE: Food Ordering Website using Flask .

INTRODUCTION:

This project is a fully responsive Crop Recommendation System built using Flask and machine learning, designed to help farmers or users identify the most suitable crops to grow based on soil and weather conditions. The system integrates the OpenWeatherMap API to fetch real-time weather data and uses trained AI/ML models to recommend the best crops. It features secure user authentication, a user-friendly interface, and visualizes data using interactive charts for better insights.

SYSTEM REQUIREMENTS:

1. Operating System:

Compatible with Windows 10/11, macOS, or any Linux distribution that supports Python and Node.js.

2. Software Requirements:

- Python 3.9 or above
- Flask Web Framework
- scikit-learn, pandas, NumPy (for ML model)
- OpenWeatherMap API key
- Matplotlib / Chart.js / Plotly (for charts)
- Node.js (for JavaScript runtime, if needed for frontend visualization)
- Code Editor (e.g., VS Code, Sublime Text)

3. Hardware Requirements:

- Minimum 4 GB RAM (8 GB recommended for ML model performance)
- At least 1 GB of free storage
- Dual-core processor or above
-

4. Browser Compatibility:

- Fully tested on Google Chrome
- Also compatible with Firefox, Microsoft Edge, and Safari

TECHNOLOGY STACK:

Frontend:

Utilizes HTML5, CSS3, and Bootstrap to create a responsive and visually appealing UI. Jinja2 is used for dynamic content rendering with seamless integration into Flask. **Charts** are rendered using JavaScript libraries like Chart.js or Plotly for visual representation of prediction and weather data.

Backend:

Built using Flask, handling backend logic and routing. The **AI model**, trained using scikit-learn, processes user inputs like soil data, temperature, and humidity to provide accurate crop recommendations.

Machine Learning:

Uses a trained ML model (e.g., Decision Tree, Random Forest, or XGBoost) to predict the most suitable crop based on features like:

- Nitrogen, Phosphorus, Potassium levels
- Temperature
- Humidity
- pH
- Rainfall

Weather Integration:

Uses the **OpenWeatherMap API** to retrieve real-time temperature and humidity based on user-input location (city name or coordinates).

Authentication:

Implements secure user login and session management using Flask-Login, allowing users to save or view previous recommendations.

SETUP INSTRUCTIONS:

1. Install Node.js and NPM (for frontend charting libraries)
Download and install from Node.js Official Website.
Verify installation:
bash
CopyEdit
node -v
npm -v

2. Install Python and Flask
Download from the Python Official Website.
Verify installation:
bash
CopyEdit
python --version
pip --version
Install necessary Python packages:
bash
CopyEdit
pip install flask flask-cors pandas numpy scikit-learn requests matplotlib
3. Setup OpenWeatherMap API

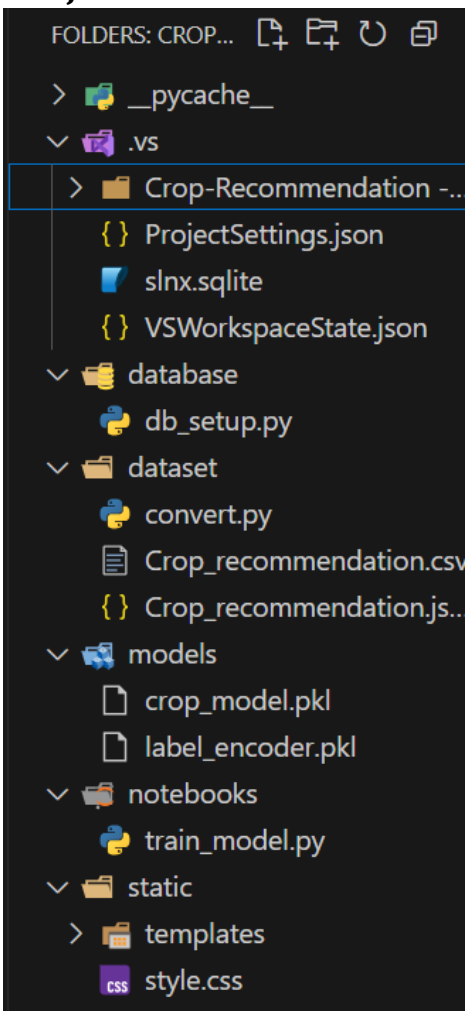
- Create a free account at https://openweathermap.org/
- Get your API key and store it securely (use .env file if needed)

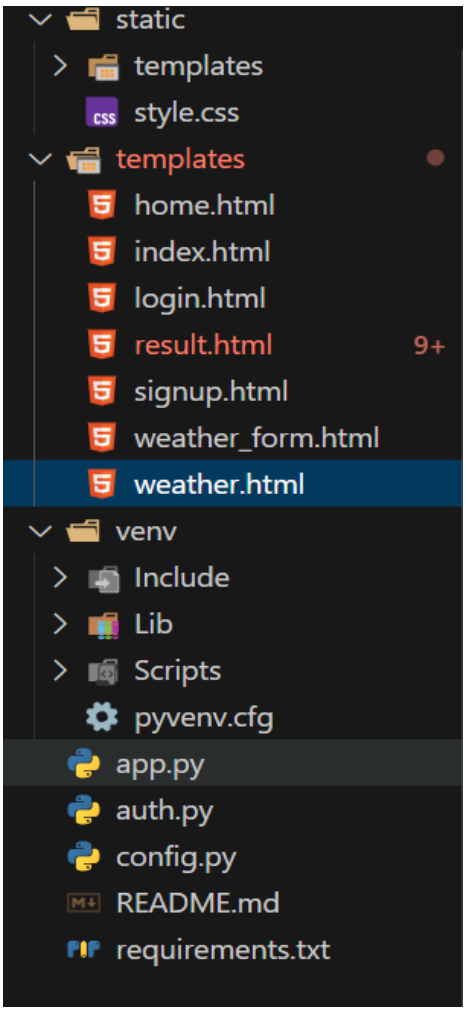
4. Run the Application

- Train your ML model (or load pre-trained model using joblib or pickle)
- Start Flask server:

bash
CopyEdit
python app.py

PROJECT STRUCTURE:





CONCLUSION:

This document outlines the requirements and setup for the Crop Recommendation System. With proper configuration and dependencies, the system can efficiently help users make informed agricultural decisions using AI and real-time weather data, enhanced with visual insights through charts.