Title: PlantDocBot : AI Plant Disease Diagnosis via Chat and Image Upload

Objective:

To develop an AI-powered chatbot that allows users (especially farmers and gardeners) to upload images of plant leaves or describe symptoms via text, and receive accurate plant disease diagnosis and treatment recommendations using image processing and natural language processing techniques.

Outcomes:

- Gain hands-on experience with computer vision and NLP integration.
- Learn how to build and fine-tune image classification models for disease detection.
- Develop a conversational AI chatbot capable of interpreting symptoms via text.
- Deliver an interactive tool capable of diagnosing common plant diseases from photos or messages in real time.

Dataset:

- **PlantVillage Dataset** (for image-based plant disease classification): Open-source dataset from PlantVillage, covering over 50,000 annotated plant leaf images.
- **PlantDoc Dataset** (real-world noisy images): Available on Kaggle.
- **Text Corpus for Plant Symptoms**: Collected from agricultural forums, blogs, and research papers (open-access sources).

Modules to be Implemented:

1. **Dataset Collection and Preprocessing**
   - Clean and augment image datasets, prepare NLP corpus.
2. **Image-based Plant Disease Detection Module**
   - Train CNN models like ResNet or MobileNet on leaf images.
3. **Symptom-based Text Diagnosis Chatbot**
   - NLP model using BERT or GPT-based architecture to understand and respond to symptom descriptions.
4. **Recommendation System for Treatment**
   - Link detected diseases to cure methods and preventive measures.
5. **User Interface and Image Upload Mechanism**
   - Web or mobile frontend allowing chat-based interaction and image uploads.
6. **Testing, Evaluation & Deployment**
   - Combine and test modules, evaluate performance, prepare documentation.

Architectural Diagram:
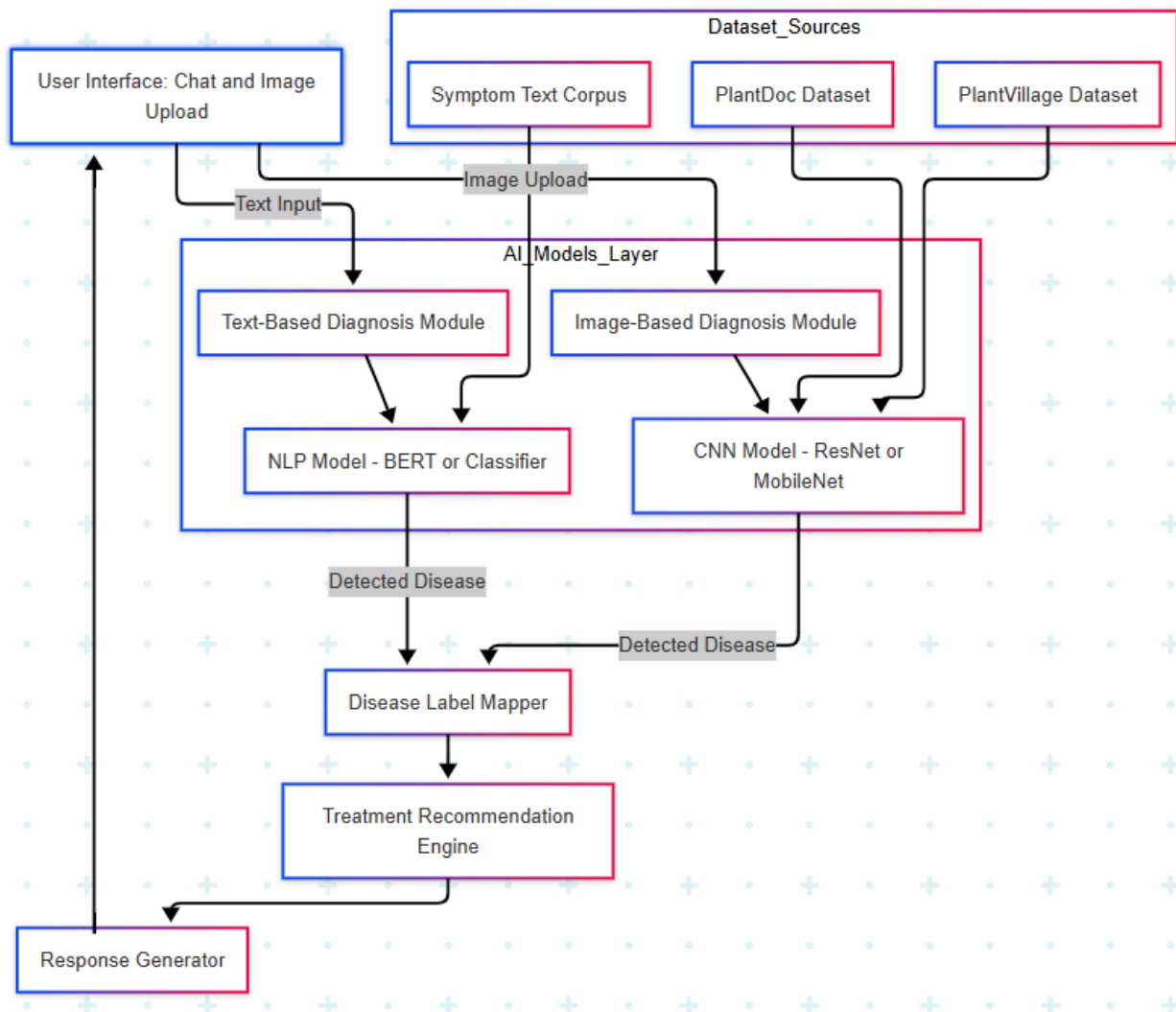
Week-wise Module Implementation:

Milestone 1: Week 1 & 2 – Requirement Gathering & Dataset Preparation

- Finalize tech stack and tools (e.g., Python, TensorFlow, HuggingFace Transformers).
- Download and organize PlantVillage and PlantDoc datasets.
- Begin collection of symptom texts and perform basic preprocessing.
- Perform image augmentation and create training/validation splits.

Milestone 2: Week 3 & 4 – Image and Text-Based Diagnosis Models

- Train and test CNN model for classifying plant diseases from images.
- Fine-tune BERT-based model for symptom text classification.
- Save model outputs for integration into the chatbot logic.

Milestone 3: Week 5 & 6 – Chatbot & Recommendation Engine

- Build chatbot logic to route image/text input to appropriate model.
- Create a rule-based or ML-based mapping from disease label to treatment.
- Integrate frontend UI with backend models and testing endpoints.

Milestone 4: Week 7 & 8 – Testing, Final Integration & Deployment

- Connect chatbot interface to both NLP and vision models.
- Conduct extensive testing with real and dummy inputs.
- Deploy project on web (using Flask or Streamlit) or mobile.
- Finalize documentation, prepare demonstration video and slides.

Evaluation Criteria:

Milestone 1:

- Successful preprocessing and preparation of both image and text datasets.
- Well-defined disease and treatment mappings.

Milestone 2:

- Accuracy of the CNN model on validation set.
- Precision/recall of text-based diagnosis model.
- Confidence scores for both inputs.

Milestone 3:

- Functional integration of chatbot with both input modes.
- Accuracy of mapping and treatment suggestions.

Milestone 4:

- Clean and intuitive UI.
- Smooth end-to-end flow from input to diagnosis to recommendation.
- Professional demo and complete documentation.