



AgroBot Universal AI-based Agricultural Assistant Batch-3

Name: Aarti Patel

Gmail: aartipatel0427@gmail.com

Mentor: springboardmentor056@gmail.com

GitHub Repository: <https://github.com/AartiPatel07>

Domain : Artificial Intelligence

SEP - NOV 2025

1. Abstract

AgroBot Universal is an advanced AI-based agricultural assistant designed to help farmers and Agri-professionals. This web-based application provides instant answers to crop-related questions in multiple languages and enables real-time plant disease detection via image analysis. Built using Python (Flask), OpenAI, and PIL, AgroBot supports user-friendly chat interfaces and a robust admin panel for knowledge base management. The system addresses the challenges of language barriers and delayed expert help in Indian agriculture, making reliable guidance accessible to all. With features like offline Q&A, OpenAI fallback, and planned voice integration, it aims to empower farmers for smarter decision-making and reduce crop loss.

2. Introduction / Background

Modern agriculture is facing challenges like crop losses due to diseases, delayed expert intervention, and communication gaps due to language barriers. While AI technologies can empower farmers, rural India lacks accessible, real-time support.

Agro Bot Universal solves these by:

- Enabling instant crop advice in local languages
- Providing automated plant disease identification from user-uploaded images
- Offering an easy interface for both farmers and administrators

3. Problem Statement

- Farmers struggle to get reliable, timely solutions for crop issues.
- Language barriers limit access to technology.
- Manual disease detection is slow and prone to errors.
- Existing agricultural platforms do not combine multi-language AI Q&A and smart image diagnosis.

4. Objectives / Outcomes

- Deliver instant crop guidance for various crops through chatbot
- Detect plant diseases via image processing for faster decisions
- Provide the features in English, Hindi, Tamil, and other Indian languages

- Enable admin management and easy knowledge base updates
- Ensure scalability and user-friendly experience for rural farmers

5. Literature Survey / Existing Systems

Many agri-apps (like Plantix, Kisan Suvidha) help with disease detection and Q&A, but:

- Limited language support
- Dependence on internet and manual queries only
- Do not offer admin/user management with offline KB AgroBot Universal addresses all these gaps.

6. Modules to be Implemented

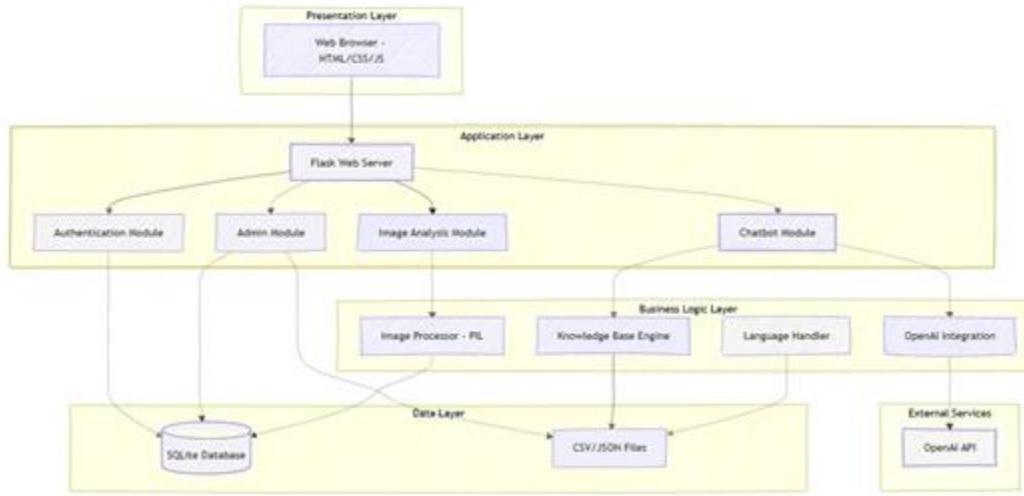
- Authentication: User registration/login, language & crop preference
- Chatbot Module: Interface for asking crop-related questions, multi-language support, offline KB search, fallback to AI (OpenAI API)
- Image Analysis Module: Upload plant/leaf images, real-time processing, health/disease result + advice
- Admin Panel: Monitor users, update KB instantly
- Database Management: Store chat history, users, KB, admin record

7. System Architecture (Flow Diagram)

Description:

- User accesses web app from browser
- Can login/register and set profile
- User asks questions in any supported language; chatbot matches KB, or uses AI for answer
- User can also upload plant/leaf image
- Image is processed using ML and image libraries (PIL), diagnosis sent back
- Admin can manage users and KB
- All data is persisted in the database

System Architecture Diagram:



fig(1): System Architecture

8. Technology Stack

- Frontend: HTML, CSS, JS, Bootstrap
- Backend: Python Flask, SQL Alchemy ORM
- AI/ML: OpenAI for chatbot, PIL/Pillow for image processing
- Database: SQLite (can upgrade)
- Other: Docker (optional), GitHub

9. Database Schema

Entities:

- User: (id, name, email, password, preferred_language, crop)
- Chat: (id, user_id, message, response, timestamp)
- KB: (id, question, answer_en, answer_hi, answer_ta, etc.)
- Admin: (id, email, password)

ER diagram:

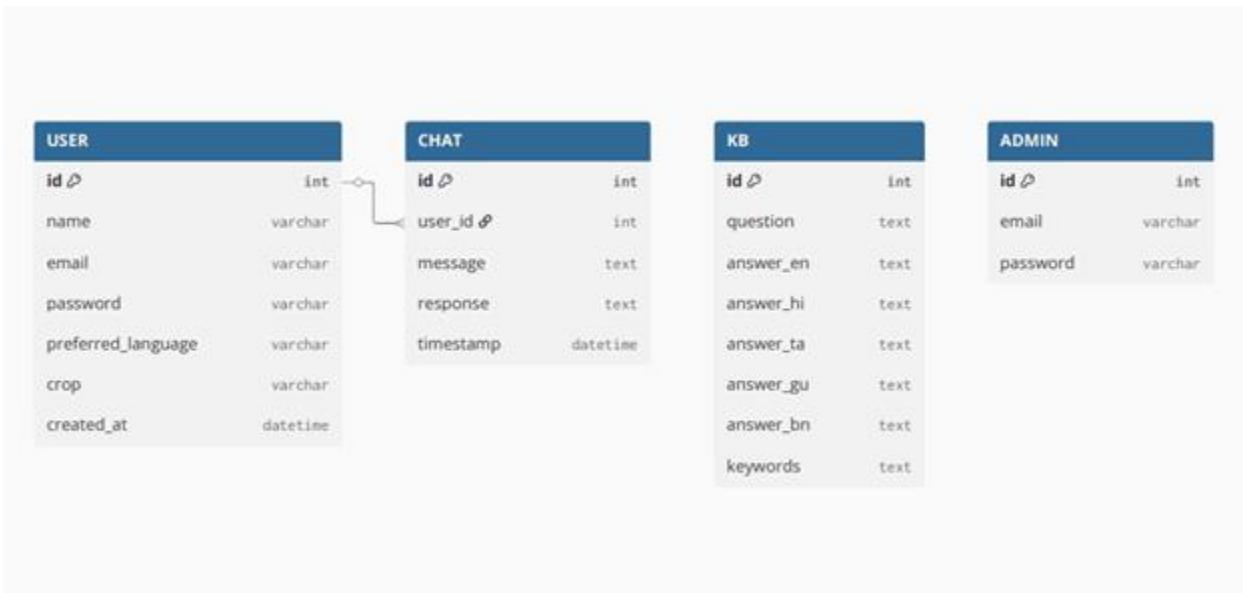


Fig (2): E-R Diagram

10. Detailed Implementation

10.1 Authentication & Profile

- Secure user registration/login via app.py
- Profile info saved (crop, region, language)

10.2 Chatbot

- User inputs query
- App finds answer in KB (checks language field based on profile)
- If not found, sends query to OpenAI, gets AI-powered response
- Response displayed in chat panel

10.3 Image Analysis

- User uploads plant image using HTML form (index.html)
- JS (script.js) sends image via AJAX to backend
- Backend (app.py) processes image using PIL, applies classification model, returns disease/health label + advice (JSON)

- Response shown below upload box

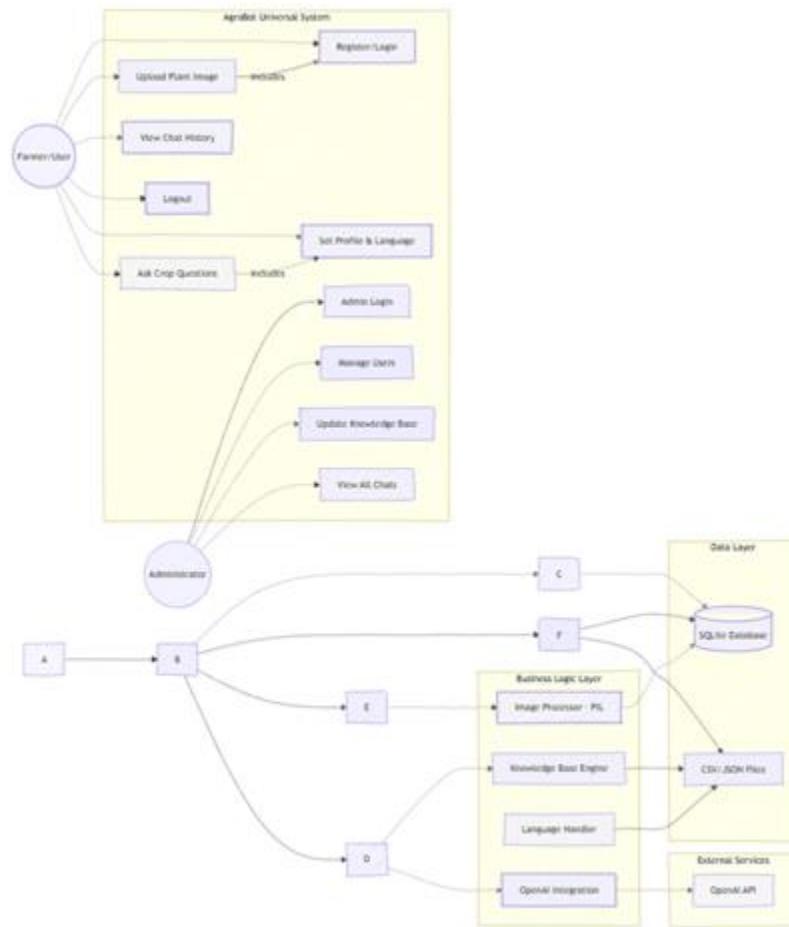
10.4 Admin Panel

- Admin login
- Can view user list/chats
- Can add/edit KB from dashboard

10.5 Data Management

- CRUD operations for user/profile, chat history, KB

Use Case Diagram:



fig(3): Use Case

11. User Guide

- Sign up/login: Go to sign up page, enter crop/language/region
- Ask questions: Use chatbot panel
- Upload image: Use image analysis panel, select image, get instant advice
- Admin: Login as admin, use dashboard for management

12. Testing & Output

Attach screenshots for:

- User login and profile page
- Chat/KB answer demo
- Image analysis upload/result
- Admin dashboard

13. Limitations & Future Enhancement

- Current image detection can be improved with more data/models
- Add more Indian languages for regional reach
- Add SMS, voice, or mobile app support
- Integration with weather/fertilizer APIs

References

1. OpenAI API Documentation
<https://platform.openai.com/docs/>
2. PyCharm IDE
<https://www.jetbrains.com/pycharm/>
3. GitHub
<https://github.com/>
4. Python Official Documentation
<https://docs.python.org/3/>
5. Research Paper: "Deep Learning for Plant Disease Detection"
IEEE Xplore / ResearchGate
6. Research Paper: "AI-based Chatbots in Agriculture"
International Journal of Computer Applications