Volume 10, Issue 11, November – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

Agricultural Advisory Management System with AI Powered Chat-Bot

Hope Wanangwa Mzumara[1]; Joel Mulepa[2]

[1](24325351004)
[2](Guide)

[1;2]DMI ST John the Baptist University Lilongwe, Malawi

Project Report

Submitted

In Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Computer Science

Publication Date: 2025/11/27

How to Cite: Hope Wanangwa Mzumara; Joel Mulepa (2025) Agricultural Advisory Management System with AI Powered Chat-Bot. *International Journal of Innovative Science and Research Technology*, 10(11), 1563-1591. https://doi.org/10.38124/ijisrt/25nov004

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

CERTIFICATION OF THE GUIDE

This is to certify that the project work entitled: AGRICULTURAL ADVISORY MANAGEMENT SYSTEM WITH AI POWERED CHAT-BOT is a Bonafide work of HOPE MZUMARA, Registration No. 24325351004 in partial fulfilment for the award of the BACHELOR OF ENGINEERING IN COMPUTER SCIENCE, DMI - ST JOHN THE BAPTIST UNIVERSITY under my guidance. This Thesis work is original one and not submitted earlier for the award of any degree / diploma elsewhere.

Signature of the Student Signature of the Guide

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

DECLARATION BY CANDIDATE

I HOPE MZUMARA hereby declare that this project report ENTITLED: AGRICULTURAL ADVISORY MANAGEMENT SYSTEM WITH CHAT-BOT is submitted to DMI ST JOHN THE BAPTIST UNIVERSITY in the partial fulfilment of requirements for the award of the degree of BACHELOR OF SCIENCE IN COMPUTER SCIENCE is a record of the original work done by us under the supervision of MR JOEL MULEPA.

Enrolment No.:

Register No:

Date:

Signature:

PROFORMA FOR APPROVAL OF PROJECT PROPOSAL

Proposed Project Team:

S. No.	Reg. No.	Name of the students	Semester	Branch
1				

Title of the Project:	
Subject Area:	
Name of the Guide:	
Designation:	
Address with Phone No.:	
Office:	DMI – SJBU
Residence:	
No. of projects & students currently working under the Guide:	

Signature of the Student Signature of the Guide with seal

Date.........................

N.B.: Please do not forget to enclose the synopsis of the project and the Bio-data of the Guide. In case the complete and signed Bio-Data of the Guide is not enclosed, the proposal will not be entertained**.**

For Office Use only:

SYNOPSIS	APPROVED		NOT APPROVED	
GUIDE	APPROVED		NOT APPROVED	

Comments / Suggestions for reformulation of the Project.

Date...................... SIGNATURE OF THE HOD

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

BIO-DATA OF THE PROPOSED GUIDE FOR PROJECT WORK

- *Personal Information*

NAME (in Block Letters):
Date of Birth & Age:
Sex:
Academic Qualification:
Official Address:
Phone No. and Fax.:
Residential Address:
Phone No., and e-mail id:

- *Details of Employment*

Designation:
Field of Specialization:
Teaching Experience (in years):
Industrial Experience (in years):
Particulars of contribution / experience in the field of specialization:
No. of Projects guided:

I Mr Joel Mulepa do hereby accept to guide Hope Mzumara student of the Bachelor of Science in Computer Science program of DMI – ST. JOHN THE BAPTIST UNIVERSITY.

Signature of the Student Signature of the Guide with Seal

Volume 10, Issue 11, November – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

BONAFIDE CERTIFICATE

DMI-ST. JOHN THE BAPTIST UNIVERSITY

LILONGWE – MALAWI

BONAFIDE CERTIFICATE

Register No: 24325351004

Certified that this is Bonafide record of work done ENTITLED: AGRICULTURAL ADVISORY MANAGEMENT SYSTEM WITH AI POWERED CHATBOT by HOPE MZUMARA of BACHELOR OF COMPUTER SCIENCE at DMI ST JOHN THE BAPTIST UNIVERSITY.

During the academic year 2024-2026

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost, I am sincerely grateful to Almighty God for granting me the strength, wisdom, and perseverance to successfully complete this project.

I extend my heartfelt appreciation to my supervisor, Mr Joel Mulepa, for his invaluable guidance, encouragement, and constructive feedback throughout the development of this project. Your expertise and insights played a significant role in shaping this work into its final form.

I would also like to thank the Department of Computer Science at DMI – St. John the Baptist University for providing the academic foundation, resources, and environment that enabled this research and development process to thrive.

Special thanks go to my lecturers, lab technicians, and academic mentors whose support and input during various stages of my study inspired me to push forward.

To my family and friends, your unwavering support, prayers, and motivation have been a constant source of strength. I am deeply thankful for your belief in me.

Lastly, I am grateful to everyone who contributed, in one way or another, to the success of this project — especially the individuals who participated in testing and gave meaningful feedback that helped improve the system.

This project, titled "Agricultural Advisory Management System with Chatbot", stands as a product of collaboration, commitment, and shared knowledge — and I am proud to present it as a contribution toward the digital transformation of agriculture in Malawi and beyond.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

ABSTRACT

Agriculture remains the backbone of Malawi's economy, supporting the livelihoods of a majority of its population. Smallholder farmers, who cultivate staple crops such as maize and sweet potatoes, face numerous challenges including limited access to timely and accurate agricultural advisory services. Traditional extension methods are often constrained by resources and infrastructure, leading to knowledge gaps that affect crop productivity and food security. This project proposes the development of an Agricultural Advisory Management System integrated with an AI-powered chatbot designed specifically to assist Malawian farmers in managing maize and sweet potato cultivation more effectively.

The system provides a comprehensive, easy-to-navigate digital repository of crop-specific farming guidelines covering all stages of crop production— from land preparation, planting, pest control, to harvesting. Beyond static information, the system incorporates an intelligent chatbot capable of interacting with farmers in natural language, offering personalized answers to queries that may not be covered in the knowledge base. Leveraging advances in artificial intelligence and natural language processing, the chatbot serves as a virtual extension officer accessible 24/7, enhancing farmer engagement and support.

This platform aims to empower smallholder farmers by increasing access to reliable agricultural knowledge, improving decision-making on the farm, and promoting sustainable farming practices. The system also supports continuous learning by capturing user feedback and evolving its knowledge base accordingly. The ultimate goal is to contribute to increased crop yields, reduced losses, and strengthened food security in Malawi.

The project is implemented using modern web technologies and AI frameworks, ensuring scalability and future expansion potential. This report details the system's objectives, design, development process, testing, and potential impact on Malawi's agricultural sector.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

TABLE OF CONTENT

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

LIST OF FIGURES

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

CHAPTER ONE
INTRODUCTION

➢ *Background*

Agriculture forms the backbone of Malawi's economy, employing the majority of the population and contributing significantly to national income. Smallholder farmers dominate the agricultural landscape, growing staple crops such as maize and sweet potatoes that are vital for food security and livelihoods. However, despite their importance, these farmers face persistent challenges including unpredictable weather patterns, pests, diseases, and limited access to modern farming knowledge and technologies.

Traditional agricultural extension services have attempted to bridge the knowledge gap but often fall short due to limited staffing, poor infrastructure, and logistical hurdles in reaching rural areas. This results in farmers relying on outdated practices, which can lead to poor crop yields and food insecurity.

The rapid growth in information and communication technologies (ICT) presents an opportunity to transform agricultural advisory services. Digital platforms can provide farmers with timely, accurate, and context-specific advice to improve productivity and resilience. In particular, Artificial Intelligence (AI) offers potential to create interactive advisory systems that provide personalized assistance at scale.

This project proposes the development of an Agricultural Advisory Management System with an AI-powered chatbot, focusing on maize and sweet potatoes — two critical crops for Malawi. The system integrates a comprehensive digital knowledge base with an intelligent conversational agent that farmers can engage with anytime to receive detailed, practical farming advice.

➢ *Purpose of the System*

The purpose of this system is to empower Malawian smallholder farmers by providing them with easy access to relevant, up-to-date agricultural information and personalized support through an AI chatbot. This will help farmers make better decisions throughout the crop lifecycle — from planting, pest management, fertilization, to harvesting.

By combining static advisory content with an interactive AI agent, the system addresses both the need for reliable information and the flexibility to answer unique, context-specific farmer queries that may arise.

➢ *Objectives*

This system aims to:

- Develop a detailed and region-specific digital knowledge base on maize and sweet potato farming.
- Implement an AI chatbot capable of natural language understanding and providing precise, context-aware answers.
- Improve farmers' access to advisory services regardless of their location or time constraints.
- Promote the adoption of best practices and sustainable farming techniques.
- Enable continuous system improvement through feedback and data analytics on user interactions.

➢ *Scope*

The system targets smallholder maize and sweet potato farmers in Malawi, providing crop-specific advice tailored to local conditions. It is designed as a web-based platform accessible via mobile devices and computers with internet access.

The scope includes the development of advisory content, AI chatbot integration, user interface design, and system testing. It excludes hardware development, real-time weather integration, and support for crops other than maize and sweet potatoes in the initial version.

➢ *Significance of the Study*

This system bridges the gap between limited traditional extension services and the increasing demand for agricultural knowledge. By leveraging AI, it offers scalable, personalized support to farmers, enhancing their ability to manage crops effectively.

Improved advisory services can lead to better yields, reduced post-harvest losses, and enhanced food security in Malawi. Additionally, data collected from farmer interactions can inform policy and extension programs, making agricultural development efforts more targeted and effective.

- *Literature Review*

- *Raman et al. (2021)*

- Title: *A Study on Application of Various Artificial Intelligence Techniques on Internet of Things*
- Summary: This paper explores how AI techniques (like machine learning and NLP) can be applied in IoT-based systems for agriculture.
- Relevance: Shows that smart advisory systems using AI can automate farm support services and improve efficiency.

- *Jain & Vohra (2020)*

- Title: *Chatbot for Agriculture using Artificial Intelligence*
- Summary: The authors designed a chatbot to provide answers on fertilizers, crop diseases, and pesticide usage using AI.
- Relevance: Supports the use of natural language interfaces to help farmers get instant, automated guidance.

- *Patil & Patil (2021)*

- Title: *AI-Based Chatbot System for Agriculture Advisory Services*
- Summary: This system used NLP and local language support to assist rural farmers with advisory information.
- Relevance: Emphasizes accessibility and how AI chatbots can close the communication gap in farming communities.

- *Shukla & Shrivastava (2020)*

- Title: *Design and Development of Agricultural Chatbot for Farmers*
- Summary: The authors created a Python-based chatbot with a feedback mechanism to answer farmers' common queries.
- Relevance: Demonstrates the role of user feedback in improving chatbot accuracy over time.

- *Kumar & Singh (2019)*

- Title: *Role of ICT in Agriculture*
- Summary: This paper outlines how ICT tools like mobile apps and web-based platforms can transform agricultural practices.
- Relevance: Provides background on how technologies like chatbots fit into the broader ICT ecosystem in agriculture.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

CHAPTER TWO
SYSTEM ANALYSIS

➢ *Problem Definition*

Agriculture in Malawi is predominantly smallholder-based, with maize and sweet potatoes as essential crops. However, many farmers struggle with accessing timely, reliable, and context-specific agricultural advice. The existing agricultural extension system faces serious limitations due to insufficient personnel, inadequate resources, and logistical challenges that make reaching rural farmers difficult. As a result, many farmers rely on traditional knowledge, hearsay, or outdated farming practices, which leads to low productivity, poor pest and disease management, and increased vulnerability to climatic and economic shocks.

Farmers also lack personalized advisory services tailored to their specific needs, crops, soil types, and pest challenges. This gap restricts their ability to optimize input use, increase yield, and improve post-harvest management. Furthermore, the rapid spread of mobile phones and internet access in Malawi has not been fully leveraged to address these challenges, leaving a critical opportunity untapped.

Therefore, there is an urgent need for a scalable, accessible, and interactive system that provides farmers with both general and customized agricultural advice, empowering them to make informed decisions.

➢ *Existing System*

Traditional agricultural advisory services in Malawi are largely delivered through government extension officers and NGOs. These officers provide training, demonstrations, and printed materials, but are constrained by high farmer-to-extension worker ratios and transportation difficulties in rural areas. The coverage is often limited, and the information may not always be up-to-date or personalized.

Some digital initiatives have emerged, including SMS-based advisory platforms and mobile apps that deliver farming tips. However, these tend to provide only one-way communication with static content, lacking interactivity and personalization. Existing chatbots or AI systems are mostly generic and not tailored to the Malawian context or specific crops like maize and sweet potatoes.

Hence, while there are some digital tools, none comprehensively integrate a rich advisory knowledge base with an intelligent AI chatbot designed specifically for Malawian smallholder farmers.

➢ *Feasibility Study*

- *Executive Summary*

This feasibility study assesses the viability of developing an Agricultural Advisory Management System with an AI chatbot for maize and sweet potato farmers in Malawi. The study evaluates technical, economic, and operational feasibility to ensure the project can be successfully implemented and sustained.

- *Technical Feasibility*

The project leverages existing web technologies and AI frameworks, such as natural language processing libraries, which are mature and widely supported. The necessary infrastructure includes basic web hosting and access to AI services, which are cost-effective and scalable. Farmers' increasing access to smartphones and internet supports system adoption.

- *Economic Feasibility*

Developing the system requires investment in software development, content creation, and maintenance. However, the potential benefits in terms of increased agricultural productivity and improved farmer livelihoods justify the costs. The system could be deployed at relatively low cost and scaled nationally, making it economically viable.

- *Operational Feasibility*

The system is user-friendly and designed with local farmer needs in mind, including language simplicity and ease of use. Training programs and awareness campaigns will enhance adoption. Ongoing content updates and chatbot learning mechanisms ensure relevance over time. Partnerships with agricultural institutions can support operational sustainability.

- *Proposed System*

The proposed system is a web-based Agricultural Advisory Management System that integrates two main components: a comprehensive knowledge base covering maize and sweet potato farming, and an AI-powered chatbot that interacts with users in natural language. The system provides detailed guidance on all aspects of crop production and allows farmers to ask specific questions and receive immediate, personalized responses.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

The chatbot is designed using advanced AI techniques that understand context and learn over time to improve accuracy. The system also collects interaction data to identify common farmer challenges and update advisory content.

- *System Objectives*

- ✓ To create an accessible digital platform offering detailed, region-specific agricultural advisory content for maize and sweet potatoes.
- ✓ To develop an AI chatbot capable of understanding and responding to farmer queries effectively.
- ✓ To provide personalized farming advice that considers local farming conditions.
- ✓ To promote sustainable agricultural practices through education.
- ✓ To enhance the reach and efficiency of agricultural advisory services in Malawi.

➢ *System Specifications*

- *Hardware Requirements*

- ✓ Web server with sufficient storage and processing capacity to host the application and AI services.
- ✓ User devices: Smartphones, tablets, or computers with internet connectivity.
- ✓ Backup and maintenance hardware to ensure data security and system reliability.

- *Software Requirements*

- ✓ Web development frameworks (e.g., Django) for the front end and back end.
- ✓ AI and natural language processing libraries (e.g., TensorFlow).
- ✓ Database management systems (e.g., MySQL) for storing advisory content and interaction logs.
- ✓ Cloud services for hosting and scaling the system.
- ✓ Security software for data protection and user privacy.

CHAPTER THREE
SYSTEM DESIGN

➢ *Introduction*

The design of the Agricultural Advisory Management System with AI chatbot aims to provide an integrated solution that bridges the knowledge gap between agricultural experts and smallholder farmers in Malawi, particularly those cultivating maize and sweet potatoes. The system focuses on usability, accessibility, and intelligent assistance to empower farmers with real-time, context-aware advice.

The design process encompasses structuring the system components into manageable layers that ensure efficient data processing, user interaction, and AI-driven query handling. This chapter details the architectural framework, system models, input/output designs, and database schema, laying a foundation for successful implementation.

- *Description*

The system design translates the functional requirements into an organized framework that supports modular development and scalability. It outlines how the various components—user interfaces, AI chatbot, advisory content repository, and administrative controls—interact seamlessly.

By adopting a modular approach, the design facilitates easy updates to agricultural content, refinement of the AI chatbot's capabilities, and ensures that user requests are processed promptly and accurately. The design also prioritizes low bandwidth usage and mobile accessibility, critical for rural farmers with limited internet infrastructure.

➢ *System Architecture*

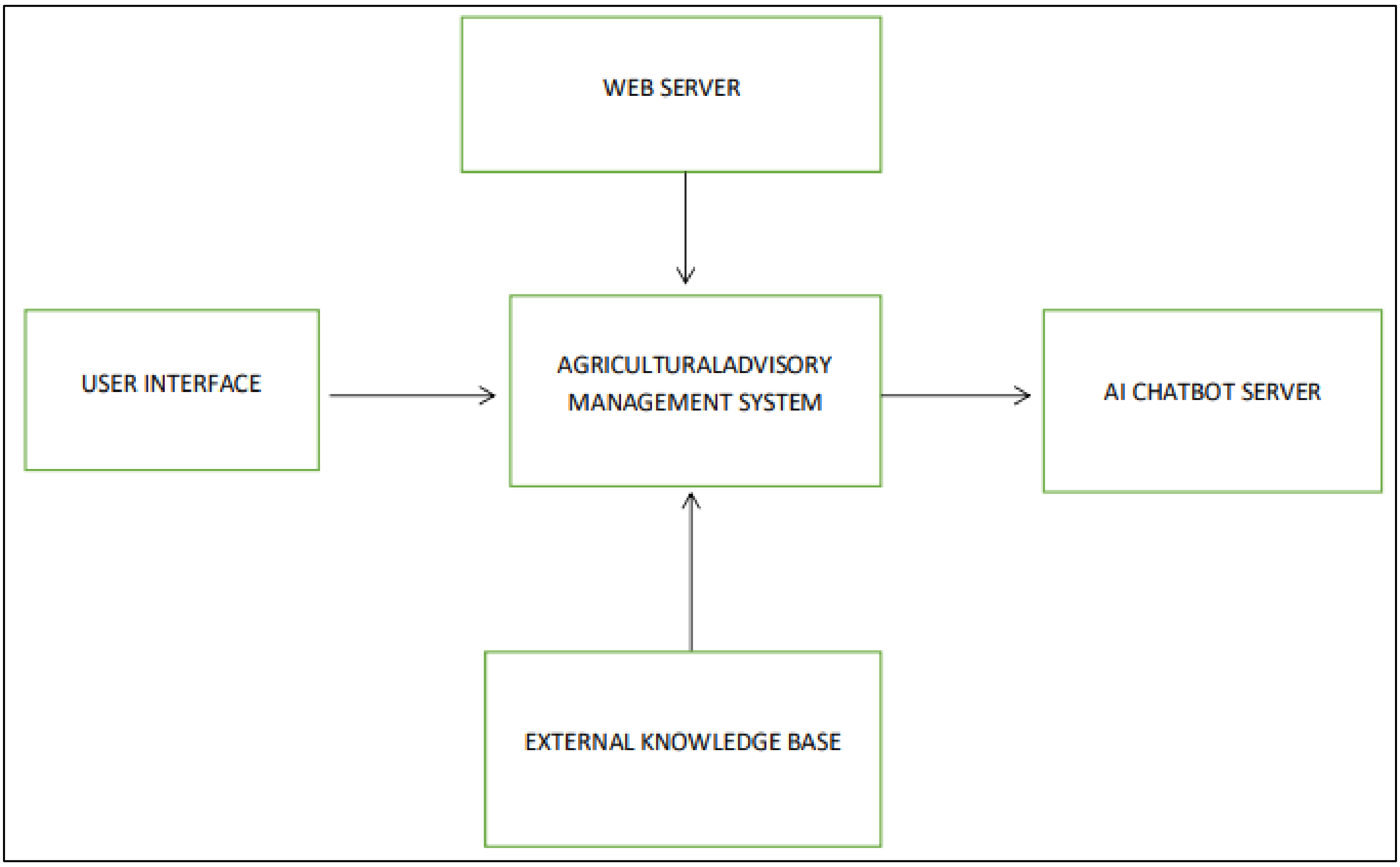

Fig 1 System Architecture

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

➢ *Use Case Diagram*

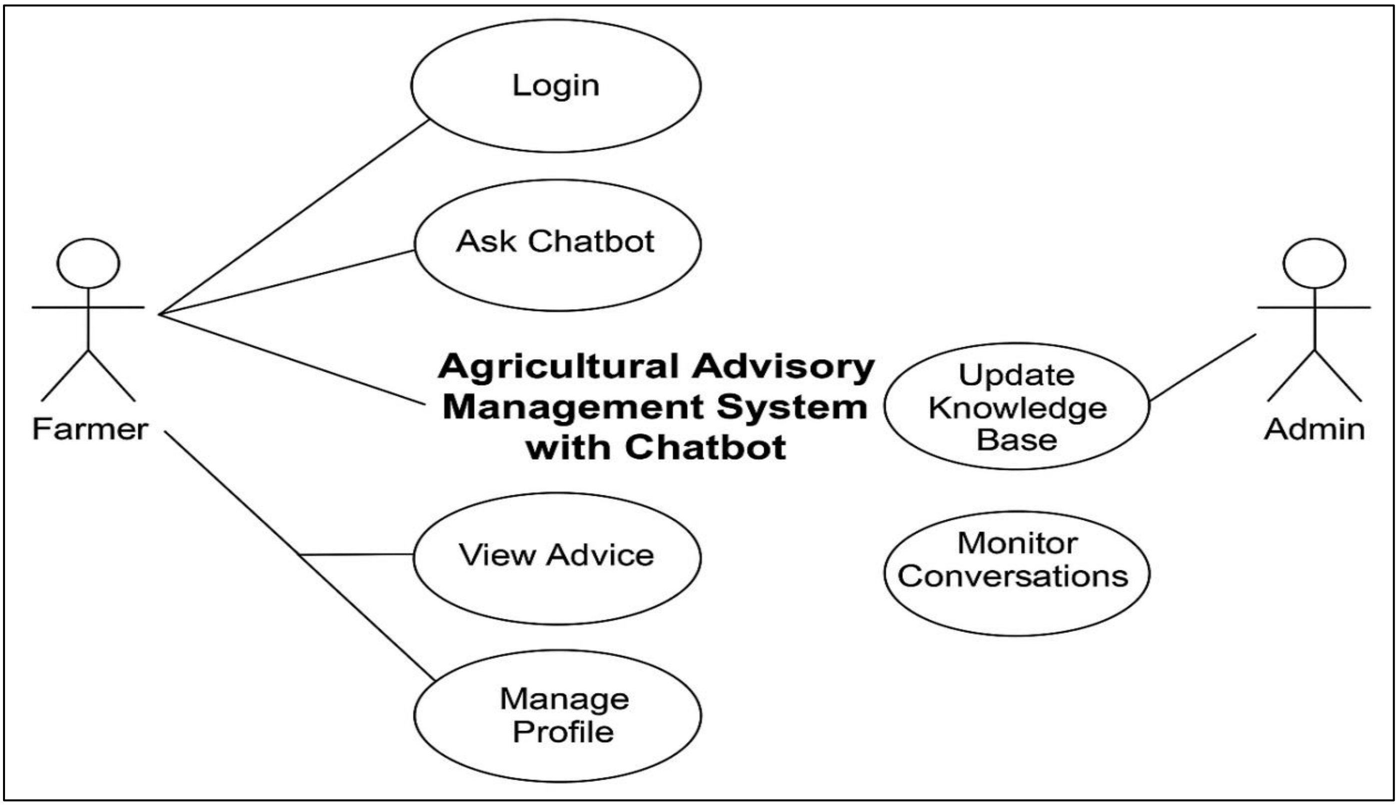

Fig 2 Use Case Diagram

➢ *Data Flow Diagram*

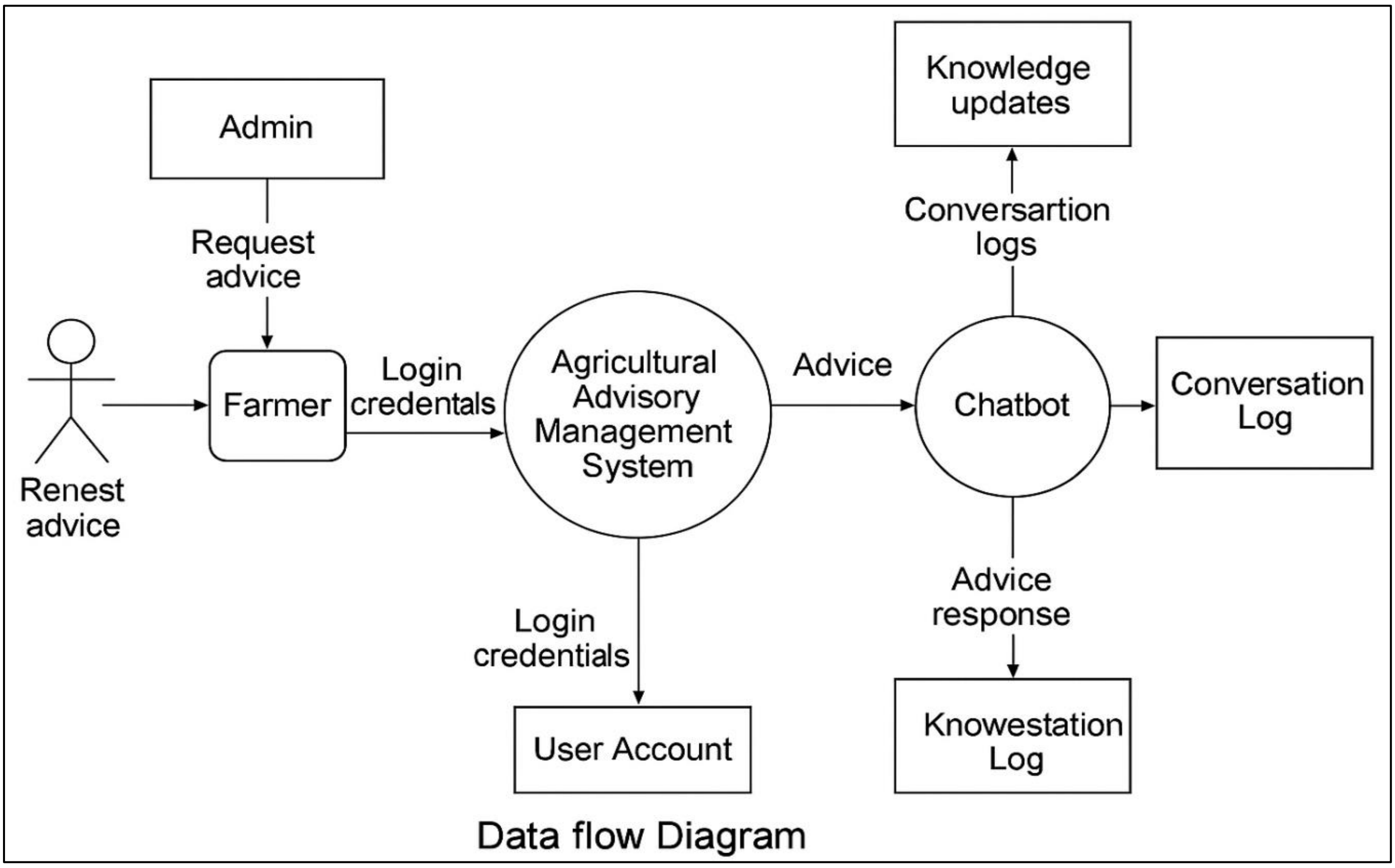

Fig 3 Data Flow Diagram

➢ *Class Diagram*

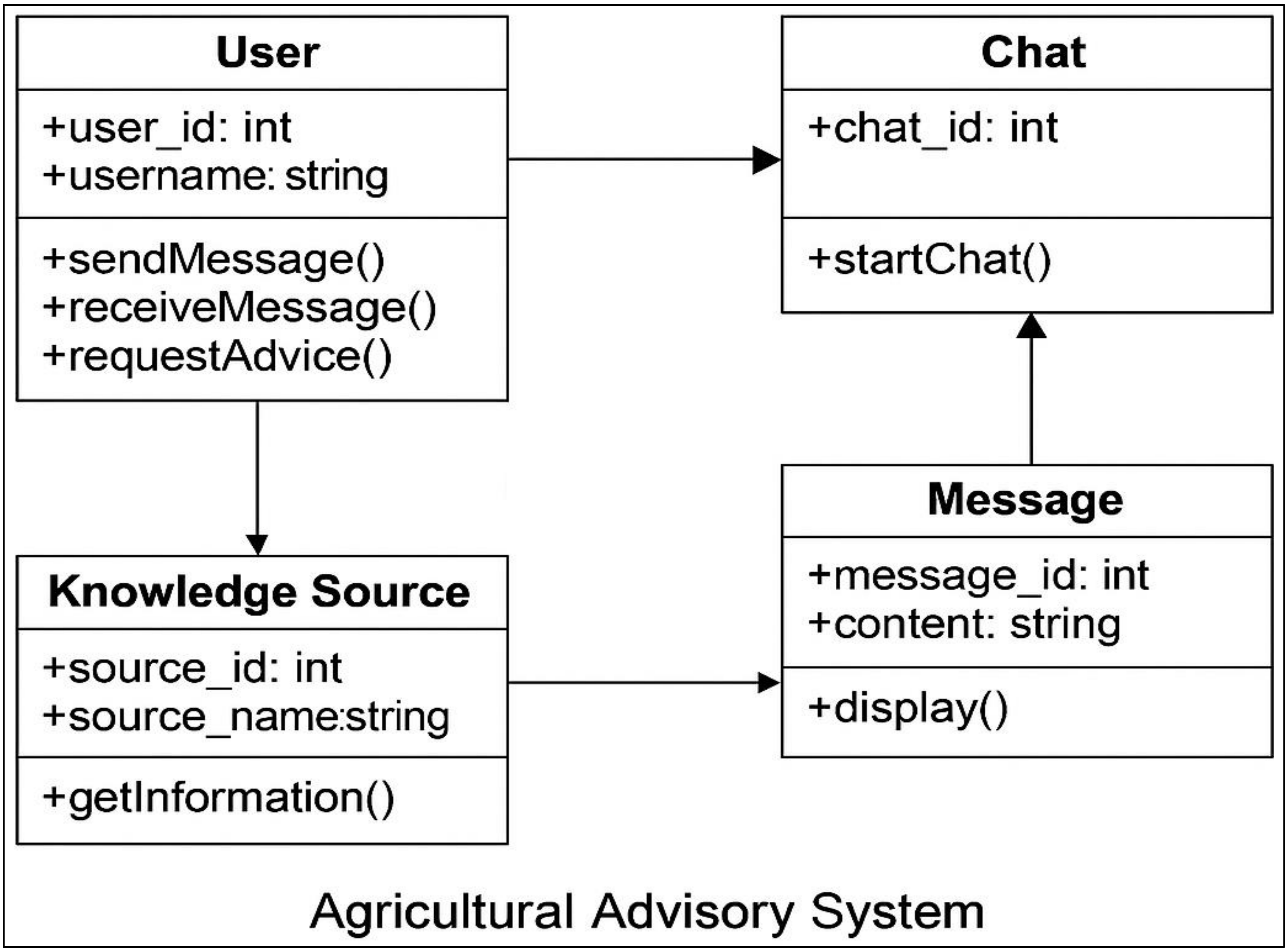

Fig 4 Class Diagram

➢ *Input Design*

Inputs are tailored for ease of use and accuracy, particularly considering farmers who may have limited digital literacy:

- Text Input: Simple, large-font input fields allow farmers to type questions or keywords in their preferred language.
- Navigation Menu: Categorized sections (e.g., planting, pests, fertilization) allow users to explore advisory content without typing.
- Search Function: Supports keyword-based searches with suggestions to guide user queries.
- Feedback Form: Structured input fields collect user suggestions, errors encountered, or additional questions to enhance system responsiveness.

The design prioritizes minimizing user errors and providing quick access to relevant information.

➢ *Output Design*

Outputs from the system are designed to be informative, clear, and supportive:

- Chatbot Replies: Generated responses use simple, local language where possible, with options for explanation or follow-up questions.
- Advisory Content Pages: Richly formatted with images, diagrams, and videos to illustrate farming techniques effectively.
- Search Results: Ranked by relevance and presented with brief summaries to help users quickly identify useful content.
- Confirmation Messages: Acknowledge receipt of feedback or confirm completion of actions to reassure users.

These outputs aim to improve user understanding and foster trust in the system.

CHAPTER FOUR
SYSTEM DEVELOPMENT

➢ *Introduction*

System development is the phase where theoretical designs and plans are translated into a functional software application. For the Agricultural Advisory Management System with AI chatbot, this stage involves coding, integration of the AI model, database implementation, and user interface creation. The development process followed industry best practices to ensure a robust, maintainable, and scalable system that meets the needs of smallholder farmers seeking guidance on maize and sweet potato farming.

This chapter discusses the module descriptions, the development methodology used, the system algorithms, and tools and technologies adopted to build the system.

➢ *Module Description*

The system is broken down into several interrelated modules, each responsible for specific functionality. This modular approach enhances maintainability and makes it easier to test and extend the system.

- *User Interface Module*

This module handles all interactions with the users (farmers). It includes screens for browsing advisory content, a chatbot interface for asking questions, user registration and login forms, and feedback submission forms. The interface is designed responsively using HTML5, CSS3, and JavaScript frameworks (such as Bootstrap) to support both desktop and mobile devices.

- *Advisory Content Management Module*

Admins use this module to upload, edit, or delete agricultural advisory materials. It supports multimedia uploads (images, videos) to complement text guidance. The content is tagged by crop type, farming stage, and topic for easy retrieval.

- *Chatbot Engine Module*

The heart of the system's AI, this module implements natural language processing (NLP) to understand farmer queries and generate relevant responses. It integrates a machine learning-based intent recognition system trained on agricultural data and common farming questions, particularly around maize and sweet potatoes. It also supports fallback mechanisms when queries are outside the current knowledge base.

- *Database Management Module*

This module is responsible for storing and retrieving data efficiently. It manages the storage of user profiles, advisory content, chat logs, and feedback. The database design supports fast query processing and data integrity.

- *Feedback and Analytics Module*

Collects user feedback and system usage data. It provides admins with reports and dashboards to monitor system performance, identify knowledge gaps, and prioritize future content updates or AI model retraining.

➢ *Methodology*

The Agile software development methodology was chosen for this project due to its iterative and flexible nature. Agile supports continuous feedback, which is essential given the evolving needs of farmers and improvements required in AI chatbot performance.

- *Key Agile Practices Applied:*

✓ Sprint Planning: Development was organized into two-week sprints, each focusing on a specific set of features such as chatbot integration, UI improvements, or content management enhancements.
✓ Daily Standups: Short daily meetings helped track progress and address blockers promptly.
✓ Incremental Delivery: Functionality was released incrementally, starting with basic advisory browsing and later expanding chatbot capabilities.
✓ User Feedback Loops: Early versions were tested by sample users (farmers and agricultural experts), whose feedback informed subsequent development cycles.

This methodology ensured the system evolved responsively to real user needs while maintaining code quality.

➢ *Algorithm*

The AI chatbot's core algorithm follows these main steps to process farmer queries and provide relevant advice:

- *Input Preprocessing:*

✓ The raw text input from the farmer is cleaned by removing punctuation, converting to lowercase, and tokenizing into individual words.

- *Intent Recognition:*

✓ Using a supervised machine learning model (such as a Support Vector Machine or a simple Neural Network), the system classifies the input into predefined intents (e.g., “pest management,” “planting time,” “fertilizer usage”).

- *Entity Extraction:*

✓ Important keywords related to crops (maize, sweet potatoes), farming stages, pests, or symptoms are identified using Named Entity Recognition (NER) techniques.

- *Knowledge Base Query:*

✓ Based on the intent and extracted entities, the system queries the advisory content database to retrieve relevant information.

- *Response Generation:*

✓ If matching content is found, the system formats a clear and concise answer.
✓ If not, the chatbot generates a fallback message suggesting alternate queries or offers to connect the farmer with a human expert (future enhancement).

- *Learning and Feedback:*

✓ User feedback on responses is collected to retrain and improve the AI model continually.

This algorithm ensures that farmers receive precise, context-aware advice and the chatbot improves over time with user interaction.

➢ *Tools and Technologies Used*

- Frontend: HTML5, CSS3, JavaScript, Bootstrap for responsive design, and jQuery for UI enhancements.
- Backend: Python Django framework to handle server-side logic, API routing, and integration of AI components.
- AI/ML Libraries: Python’s scikit-learn for training intent classifiers, NLTK and spaCy for NLP tasks including tokenization and entity recognition.
- Database: PostgreSQL for structured data storage, chosen for reliability and scalability.
- Chatbot Interface: Integrated WebSocket technology for real-time chat interactions.
- Version Control: Git and GitHub for code management and collaboration.
- Deployment: Hosted on a Linux VPS with Nginx as a reverse proxy and Gunicorn as the WSGI HTTP server.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

CHAPTER FIVE
SYSTEM TESTING

A. Introduction

Testing is a vital stage in software development. It helps in detecting errors, missing functionalities, and verifying whether the developed system meets its defined objectives and user requirements. For this project, Agricultural Advisory Management System with Chatbot, testing ensured that each component of the system — from login authentication to chatbot advisory — performed correctly and reliably. The main goal was to ensure that the application met both functional and non-functional expectations.

B. Test Plan

➢ *Unit Testing*

The first phase involved unit testing. Each module was individually tested to ensure it functioned correctly. This type of testing focused on the internal logic of specific components such as:

- User login and authentication
- Chatbot input/output
- Advisory article display
- User queries handling and storage

Errors identified in the individual modules were immediately corrected. This increased the program clarity and helped improve stability. Since the system was designed using modular architecture, unit testing was done for each module independently during development.

➢ *Integration Testing*

Once individual modules were confirmed to work properly, integration testing was performed. In this phase, the interaction between modules was tested:

- Whether the chatbot could access and retrieve data from the advisory database
- Whether the user interface could handle input and display responses correctly
- Whether routing between components (e.g., login to dashboard, dashboard to chatbot) was smooth

This was necessary because even if individual modules function correctly, there could still be data mismatches or logical failures when they are combined. Integration testing ensured all modules worked harmoniously as a complete system.

➢ *System Testing*

System testing involved testing the entire system as a whole. Inputs were provided, and outputs were compared with the expected results. This validated that the entire system adhered to the original project goals and functional specifications.

Key tests conducted during system testing included:

- Checking chatbot accuracy and response time
- Validating user login and session handling
- Testing the advisory content retrieval from the database
- Simulating incorrect or unexpected input to assess error handling

The results confirmed that the system performed correctly and consistently under realistic conditions.

➢ *Acceptance Testing*

Acceptance testing was conducted to evaluate whether the developed system was ready for deployment and actual use by end users (farmers and agricultural students). This form of testing focused on:

- Usability: Ease of navigating the system
- Functionality: Verifying chatbot responses were helpful and appropriate
- Stability: Confirming the application could run without crashing
- Performance: Evaluating system response time under user load

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

Users were given a chance to interact with the system and provide feedback. Based on this feedback, minor adjustments were made to improve chatbot interaction and error messages.

After successful acceptance testing, the system was confirmed to be functioning as intended and met the expectations of both developers and users.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

CHAPTER SIX
SYSTEM IMPLEMENTATION, SCREENSHOTS, CODING

A. Introduction

System implementation is the phase in which the actual working system is delivered. It involves translating the final design into a fully operational information system. This phase includes physical system construction, software installation, configuration, deployment, and validation to ensure the system operates as intended. Implementation also ensures the system adheres to quality standards and fulfills the user requirements defined during the analysis and design phases.

For this project, titled "Agricultural Advisory Management System with Chatbot", system implementation covered the integration of both frontend and backend components, database configuration, chatbot training, and interface testing. The development process followed an iterative approach, allowing for incremental module testing and validation.

B. Screenshots.

➢ *Log-in Page*

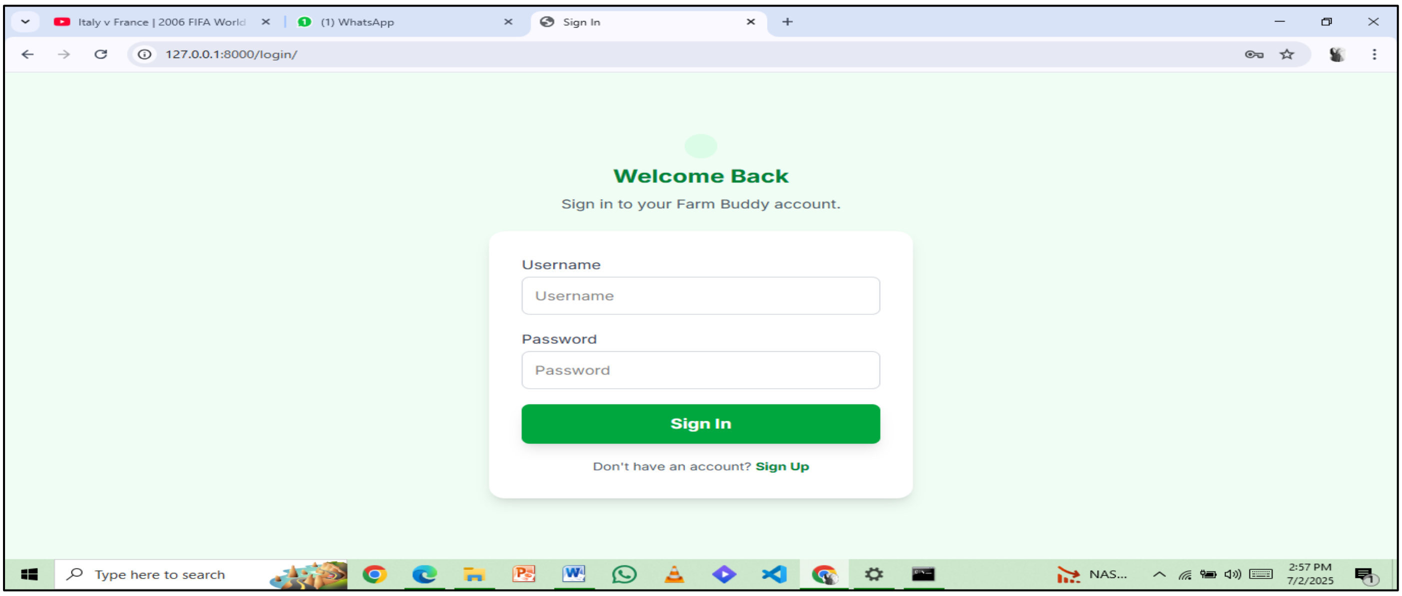

Fig 5 Log in Page

➢ *Default Routing*

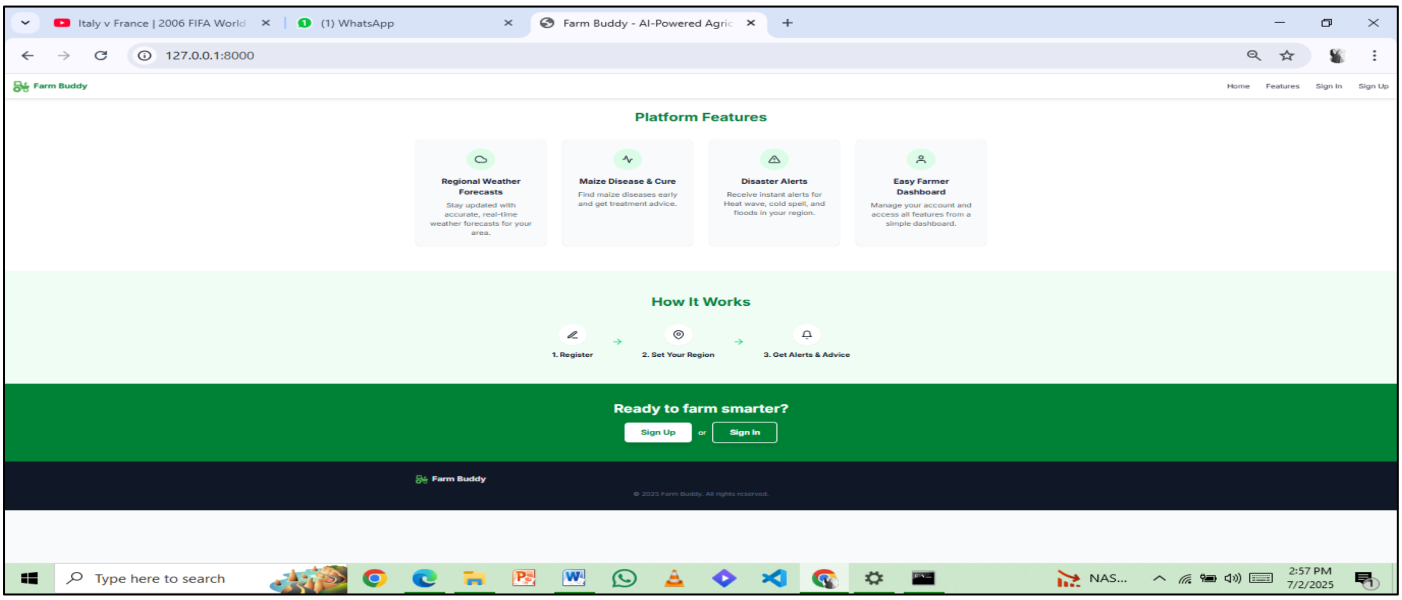

Fig 6 Default Routing

➢ *User Sign Up Page*

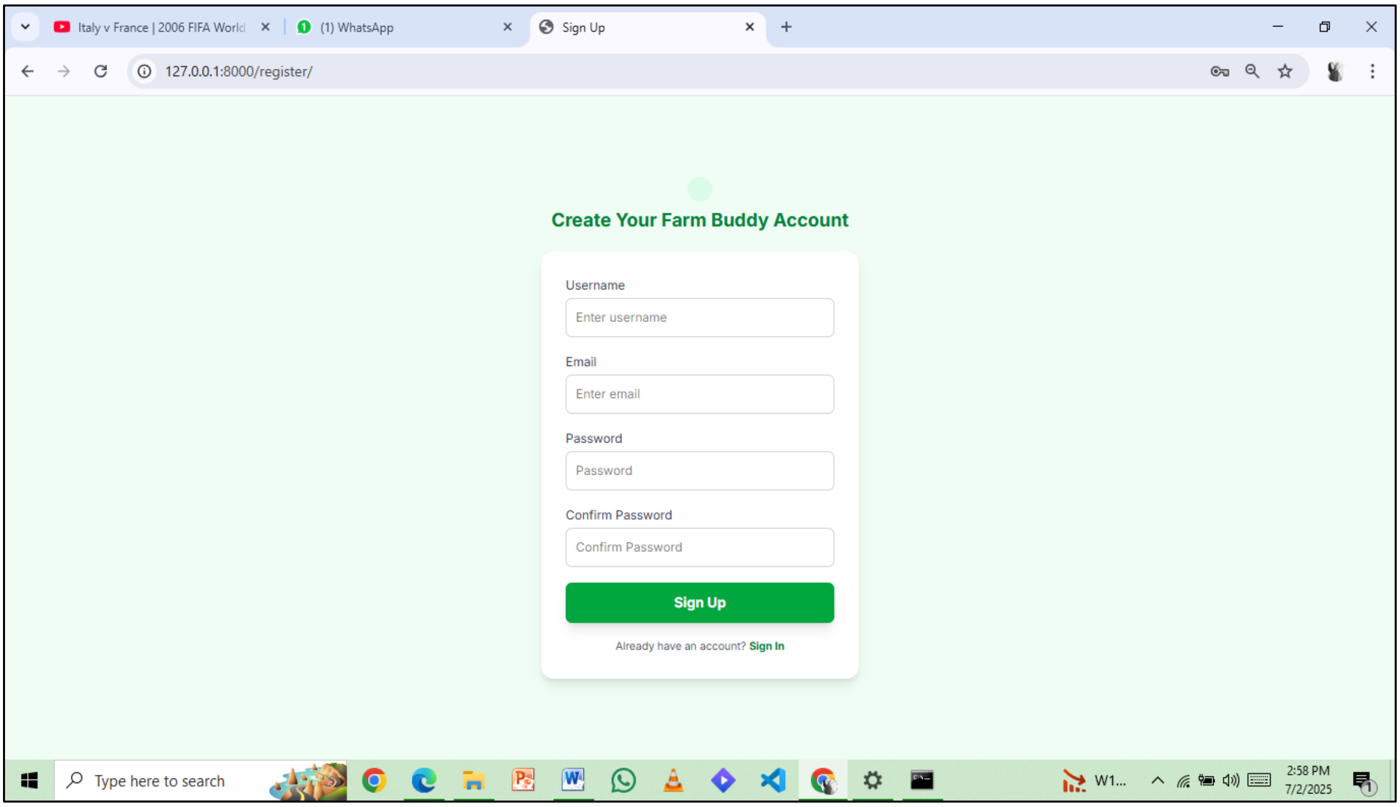

Fig 7 User Sign Up Page

➢ *Home Page*

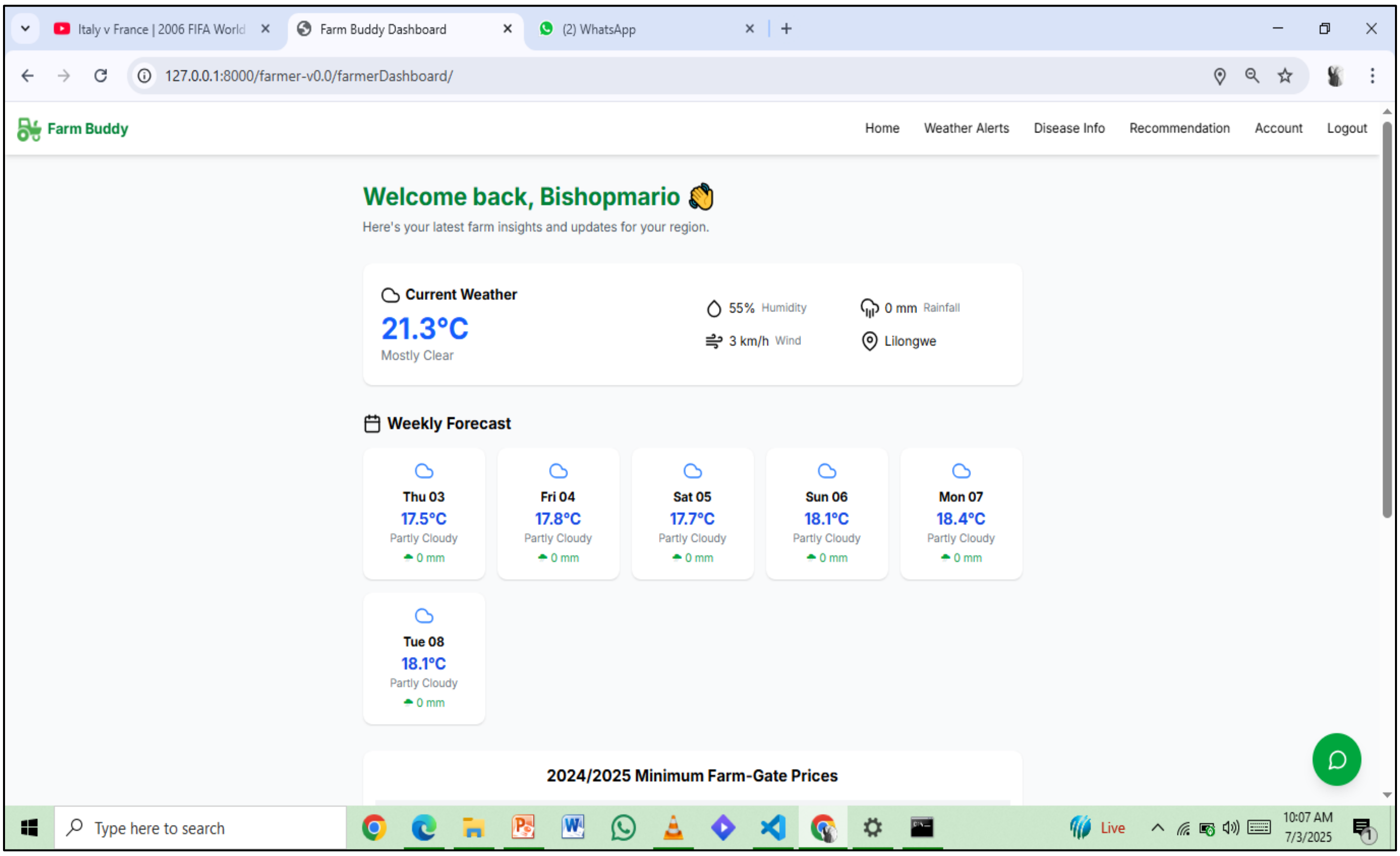

Fig 8 Home Page

➢ *Chat-Bot Section*

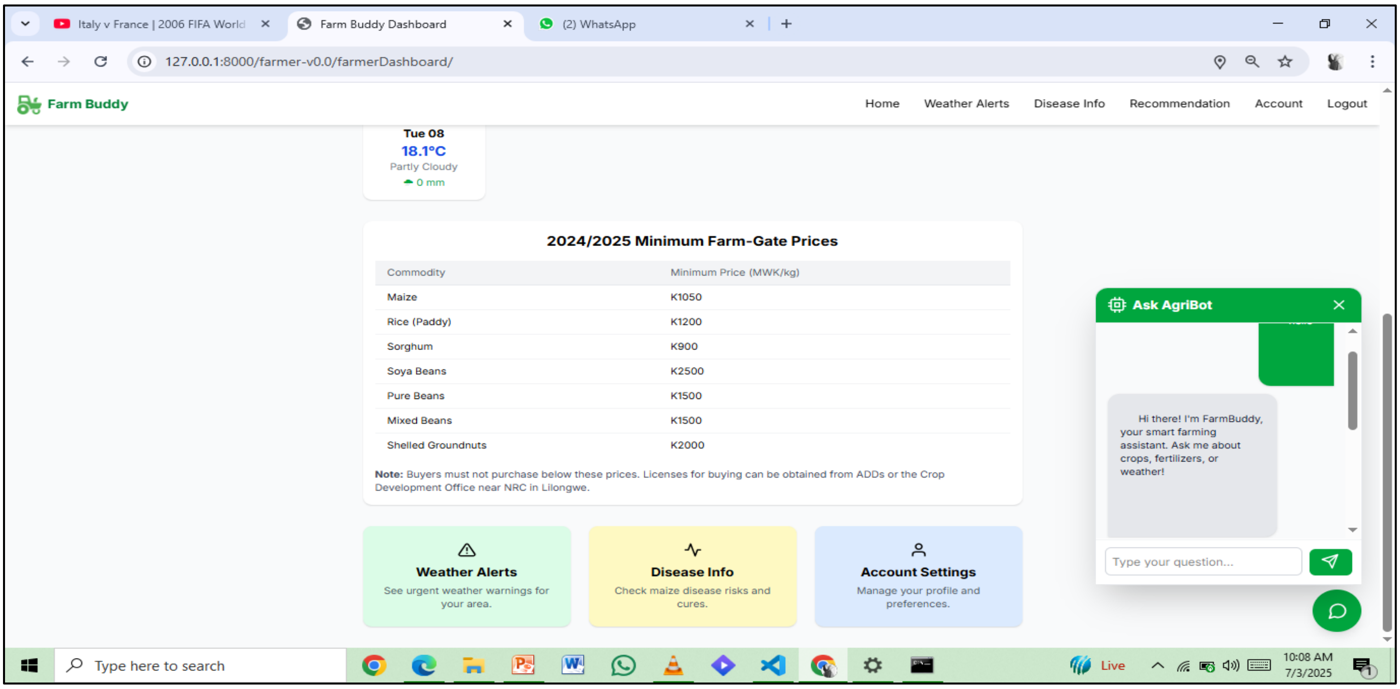

Fig 9 Chat-Bot Section

C. Coding

```
from django.contrib.auth.backends import ModelBackend
from django.contrib.auth import get_user_model
from django.contrib.auth.hashers import check_password

User = get_user_model()

class UsernameBackend(ModelBackend):
    def authenticate(self, request, username=None, password=None, **kwargs):
        print("Authenticating:", username)
        try:
            user = User.objects.get(username=username)
            if user:
                print("User authenticated successfully")
                return user
        except User.DoesNotExist:
            return None
        else:
            if user.check_password(password):
                return user

    def get_user(self, user_id):
        try:
            return User.objects.get(pk=user_id)
        except User.DoesNotExist:
            return None
```

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

```
from django.db import models
from django.contrib.auth.models import AbstractUser, UserManager
from django.contrib.auth.hashers import make_password

# Create your models here.

class CustomUserManager(UserManager):
    def _create_user(self, username, email, password, **extra_fields):
        email = self.normalize_email(email)
        user = CustomUser(username=username, email=email, **extra_fields)
        user.password = make_password(password)
        user.save(using=self._db)
        return user

    def create_user(self, username, email=None, password=None,
**extra_fields):
        extra_fields.setdefault("is_staff", False)
        extra_fields.setdefault("is_superuser", False)
        return self._create_user(username, email, password, **extra_fields)

    def create_superuser(self, username, email=None, password=None,
**extra_fields):
        extra_fields.setdefault("is_staff", True)
        extra_fields.setdefault("is_superuser", True)
        extra_fields.setdefault("user_type", "A")
        extra_fields.setdefault("last_name", "System")
        extra_fields.setdefault("first_name", "Administrator")

        assert extra_fields["is_staff"]
        assert extra_fields["is_superuser"]
        return self._create_user(username, email, password, **extra_fields)

class CustomUser(AbstractUser):
    USER_TYPE_CHOICES = (
        ("A", "Admin"),
        ("F", "Farmer"),
    )

    username = models.CharField(max_length=150, unique=True)
    email = models.EmailField(unique=True)
    user_type = models.CharField(choices=USER_TYPE_CHOICES, default=2,
max_length=20)
    password = models.CharField(max_length=12)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    USERNAME_FIELD = "username"
    REQUIRED_FIELDS = ["email"]
```

Volume 10, Issue 11, November – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25nov004

```
    objects = CustomUserManager()

    def __str__(self):
        return f"{self.last_name} {self.first_name} {self.username.capitalize()} ({self.get_user_type_display()})"

class ChatSession(models.Model):
    user = models.ForeignKey(CustomUser, on_delete=models.CASCADE)
    started_at = models.DateTimeField(auto_now_add=True)
    last_updated = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f'Session {self.id} - {self.user.username.capitalize()}'

class Chat(models.Model):
    session = models.ForeignKey(ChatSession, on_delete=models.CASCADE, related_name='chats')
    message = models.TextField(help_text="Farmer's message")
    response = models.TextField(help_text="AI model's response")
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.session.user.username.capitalize()} at {self.created_at.strftime("%Y-%m-%d %H:%M")} - {self.message[:30]}'

class FarmerLocation(models.Model):
    user = models.ForeignKey(CustomUser, on_delete=models.CASCADE, limit_choices_to={'F'})
    location_name = models.CharField(max_length=150, null=False, blank=False)
    latitude = models.FloatField()
    longitude = models.FloatField()
    created_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f"Location of {self.user.username.capitalize()} at {self.location_name}"
```

CHAPTER SEVEN
SUMMARY, CONCLUSION AND RECOMMENDATIONS

A. Summary

This project was focused on the design and development of an Agricultural Advisory Management System integrated with a Chatbot. The aim was to create an intelligent and interactive platform that provides real-time farming advice to users such as farmers and agricultural students.

The system was developed using the Django framework with a combination of front-end technologies (HTML, CSS, JavaScript) and a backend database (SQLite). The chatbot was integrated to provide conversational access to agricultural information based on user queries.

The project followed a structured software development life cycle (SDLC), including requirement analysis, system design, implementation, testing, and deployment.

➢ *Key Modules of the System Included:*

- User authentication and login
- Interactive chatbot interface
- Knowledge base for advisory content
- Admin module for content management
- Search and navigation tools

System testing, including unit, integration, and user acceptance testing, confirmed that the system met its intended goals and user requirements. The feedback received from sample users indicated high usability, relevance of information, and ease of navigation.

B. Conclusion

The Agricultural Advisory Management System with Chatbot successfully addresses the challenge of limited access to expert agricultural knowledge in real-time, especially in rural or underserved communities. The chatbot provides immediate responses to user questions regarding crop management, pest control, weather, and other agricultural concerns.

The system demonstrates how artificial intelligence can be applied in agriculture to enhance decision-making, productivity, and knowledge dissemination. It offers a convenient and user-friendly platform where users can interact with the system through natural language and receive relevant, informative responses.

Through this project, the objective of building a simple yet intelligent advisory system was achieved. The system is scalable and can be improved further to serve a broader audience with more advanced features.

C. Future Enhancements

Based on the project evaluation and user feedback, several opportunities for future improvement and expansion have been identified:

Expansion of Knowledge Base Incorporate more detailed and diversified content, including topics such as animal husbandry, irrigation methods, organic farming, and post-harvest handling.

Multilingual Support Integrate support for local languages such as Chichewa or Swahili to enhance accessibility for rural farmers who may not be proficient in English.

Mobile Application Version Develop a dedicated Android/iOS mobile application to increase reach and convenience, especially for users who primarily access the internet via smartphones.

Online Hosting and Real-time Access Deploy the system to a live server or cloud platform to allow 24/7 access and real-time updates from any location.

Integration with External APIs Use APIs for weather updates, market prices, or governmental agricultural bulletins to enrich chatbot responses with real-time external data.

Machine Learning Integration Enable the chatbot to learn from new user queries using machine learning techniques, improving accuracy and adaptability over time.

User Feedback Mechanism Implement a feature where users can rate responses or submit feedback, helping to continually refine the knowledge base and chatbot performance.

Offline Access Functionality Develop offline access modes or downloadable advisory booklets for users in low-connectivity areas.

REFERENCES

[1]. Raman, D. R., Saravanan, D., Parthiban, R., Palani, U., David, D. S., Usharani, S., & Jayakumar, D. (2021). A study on application of various artificial intelligence techniques on Internet of Things. *European Journal of Molecular & Clinical Medicine, 7*(9), 2531–2557.
[2]. Saravanan, D., Kumar, K. S., Sathya, R., & Palani, U. (2020). An IoT based air quality monitoring and air pollutant level prediction system using machine learning approach – DLMNN. *International Journal of Future Generation Communication and Networking*, *13*(4), 925–945.
[3]. David, D. S., Saravanan, M., & Jayachandran, A. (2020). Deep convolutional neural network based early diagnosis of multi-class brain tumour classification. *Solid State Technology*, *63*(6), 3599–3623.
[4]. Choi, W., Hwang, D., Kim, J., & Lee, J. (2018). Fine dust monitoring system based on Internet of Things. In *International Conference on Information and Communication Technology Robotics*.
[5]. Wang, Z., Feng, J., Fu, Q., & Gao, S. (2019). Quality control of online monitoring data of air pollutants using artificial neural networks. *Air Quality, Atmosphere & Health*, *12*(10), 1189–1196. https://doi.org/10.1007/s11869-019-00734-4
[6]. Nguyen, H., & Bui, X. N. (2018). Predicting blast-induced air overpressure: A robust artificial intelligence system based on artificial neural networks and random forest. *Natural Resources Research*, *28*(3), 893–907. https://doi.org/10.1007/s11053-018-9424-1
[7]. Ganesh, S. S., Arulmozhivarman, P., & Tatavarti, R. (2019). Forecasting air quality index using an ensemble of artificial neural networks and regression models. *Journal of Intelligent Systems*, *28*(5), 893–903.
[8]. Kumar, S., & Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 1341–1346). IEEE. https://doi.org/10.1109/CCAA.2017.8229983