Engineering and Scientific International Journal (ESIJ)
Volume 13, Issue 2, April – June 2026
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Smart Agriculture Advisory System using Django for Real-Time Farming Guidance

Sahni Sumith Kumar[*1], S. Manjunath Reddy[2]

[1] *Student, Department of Computer Applications, Viswam Engineering College, Madanapalle, Andhra Pradesh.*
[2]*Assistant Professor, Department of Computer Applications, Viswam Engineering College, Madanapalle, Andhra Pradesh.*

Abstract — Agriculture plays a vital role in the economic development of countries like India, yet farmers often face challenges due to limited access to timely and accurate agricultural advisory services. Traditional knowledge-sharing methods are constrained by geographic, temporal, and resource limitations, leading to inefficiencies in farming practices. This paper presents a Smart Agriculture Advisory System, a web-based platform designed to bridge the gap between agricultural expertise and farmers through a centralized digital solution. The system is developed using a full-stack architecture with a frontend built using HTML, CSS, Bootstrap, and JavaScript, ensuring a responsive and user-friendly interface. The backend is implemented using Python and the Django framework, which manages application logic, user authentication, and database operations. SQLite3 is used as the database for storing farmer profiles, crop information, and query records. The platform consists of two main components: a Farmer Portal and an Admin Portal. Farmers can submit queries related to crop cultivation, fertilizers, pest management, and irrigation, while administrators manage crop data and advisory content. A keyword-based intelligent query processing engine generates real-time responses to farmer queries. The system improves accessibility to agricultural knowledge, enhances decision-making efficiency, and demonstrates the potential of web technologies in empowering rural communities.

Keywords: *Smart Agriculture; Django Framework; Crop Advisory; Web Application; Keyword-Based AI; Farmer Support System.*

1. Introduction

Agriculture has long been the backbone of human civilization and continues to play a crucial role in sustaining economies and livelihoods, particularly in developing countries like India. Despite its importance, farmers face numerous challenges, including limited access to expert knowledge, delayed advisory services, and reliance on informal information sources. Traditional advisory methods such as government extension services and community-based knowledge sharing are often constrained by scalability and accessibility issues. These limitations result in delayed decision-making and suboptimal farming practices. With the rapid advancement of digital technologies and increased internet penetration in rural areas, there is a significant opportunity to transform agricultural advisory systems. The proposed Smart Agriculture Advisory System leverages web technologies to provide farmers with instant, reliable, and accessible agricultural guidance. The system is implemented as a full-stack web application using Django, which ensures secure backend processing and efficient database management. The platform enables farmers to interact with the system through a simple interface, submit queries, and receive real-time advisory responses.

1.1 Project Objectives

The primary objectives of the system include: developing a centralized web-based agricultural advisory platform; providing real-time responses to farmer queries; enabling secure farmer registration and authentication; maintaining a comprehensive crop advisory knowledge base; designing a responsive and user-friendly interface; and supporting administrative management of agricultural data.

1.2 Problem Statement

Farmers, especially in rural areas, often lack access to timely and accurate agricultural guidance, leading to inefficient farming practices and reduced productivity. Existing advisory systems are limited by geographic constraints and limited reach, lack of real-time interaction, dependence on informal and outdated knowledge, and limited personalization of advisory services. The proposed system addresses these issues by providing a digital, accessible, and interactive advisory platform that enables farmers to receive instant and relevant agricultural guidance.

2. Literature Survey

The agricultural sector has adopted digital technologies to improve productivity and access to expert knowledge, but many systems still face challenges in accessibility, personalization, and real-time interaction. Traditional advisory methods rely on government extension services, which provide reliable guidance but are limited in reach and responsiveness. Digital platforms like Kisan Suvidha and AgriApp improve information access but mainly offer static content, requiring users to search manually.

DOI: **10.30726/esij/v13.i2.2026.1320034**

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Web-based systems provide scalable and cost-effective solutions but often lack interactive query features. Advanced technologies like AI show promise but require complex infrastructure, limiting their use in smaller setups. The proposed Smart Agriculture Advisory System addresses these gaps by offering an interactive, web-based platform with a keyword-driven query system. It provides instant responses, low computational requirements, and continuous updates through an admin module, making it accessible, efficient, and user-friendly.

3. Proposed Methodology

The proposed Smart Agriculture Advisory System is a centralized web platform designed to provide real-time agricultural guidance through a full-stack architecture. It enables farmers to access timely and relevant advisory services in a simple and efficient manner. The system follows a query-response model where farmers submit questions and receive answers using a keyword-based advisory engine. This ensures fast responses with low computational cost, making it suitable for resource-limited environments. After authentication, users can access the dashboard to view crop information or submit queries related to cultivation practices, fertilizers, pest control, and irrigation. The system processes these queries using keyword matching and retrieves relevant responses from a predefined knowledge base, delivering instant results to users.

3.1 System Architecture

The Smart Agriculture Advisory System follows a three-tier architecture, ensuring modularity, scalability, and maintainability. The Presentation Layer is built using HTML, CSS, Bootstrap, and JavaScript, providing Farmer Portal and Admin Portal interfaces with responsive design. The Application Layer (Django Backend) processes all system logic including HTTP requests, user authentication, sessions, and query processing, connecting the frontend with the database through Django ORM. The Data Layer uses SQLite3 for storing farmer details, crop advisory information, and query-response history. The system architecture is illustrated in Fig. 1.

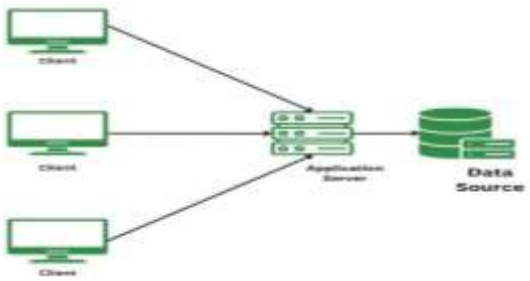

Fig. 1: System Architecture of Smart Agriculture Advisory System

3.2 System Flow Diagram

The system flow encompasses user registration and authentication, crop information browsing, query submission and processing, and response generation and display. The complete system flow is depicted in Fig. 2.

Fig. 2: System Flow Diagram of Smart Agriculture Advisory System

3.3 UML Diagrams

3.3.1 Use Case Diagram

The Use Case Diagram in Fig. 3 represents the interaction between the two primary actors: Farmer and Administrator. The Farmer can register an account, log in, view crop advisory information, submit farming queries, view advisory responses, and logout. The Administrator can log into the admin portal, manage crop data, review farmer queries, provide or update advisory responses, and maintain the agricultural knowledge base.

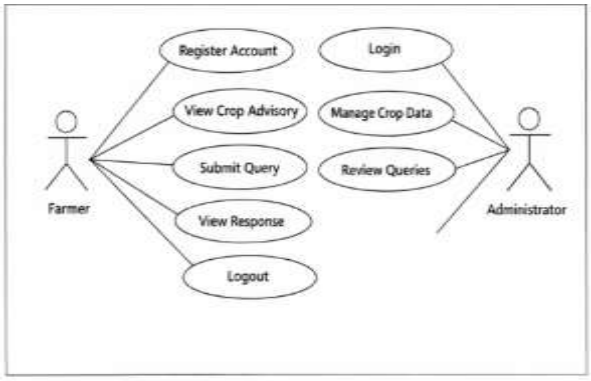

Fig. 3: Use Case Diagram of Smart Agriculture Advisory System

3.3.2 Class Diagram

The Class Diagram in Fig. 4 illustrates the structural components of the system. The core classes include Farmer (with attributes: name, email, phone, password), Crop (with attributes: crop_name, season, fertilizer, irrigation), and Query (with attributes: question, answer, timestamp). These

DOI: 10.30726/esij/v13.i2.2026.1320034

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

classes represent registered users, crop advisory knowledge, and farmer-submitted queries with corresponding responses.

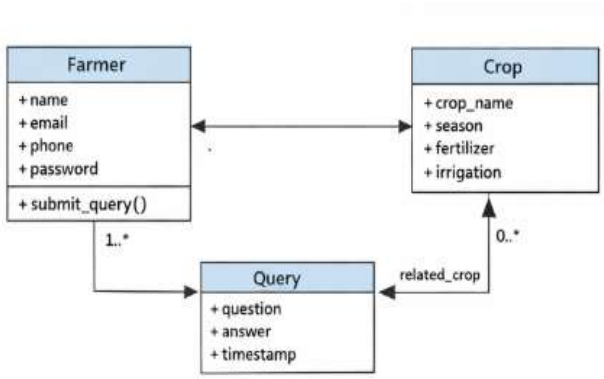

Fig. 4: Class Diagram of Smart Agriculture Advisory System

3.3.3 Sequence Diagram

The Sequence Diagram in Fig. 5 traces the interaction flow during farmer query submission. The Farmer opens the query page, the browser sends a request to the Django URL Dispatcher, which routes it to the query_page view. Upon form submission, the view validates the form, saves the query in the SQLite database, and calls the get_ai_answer() function. A keyword-based advisory response is generated and rendered back to the farmer.

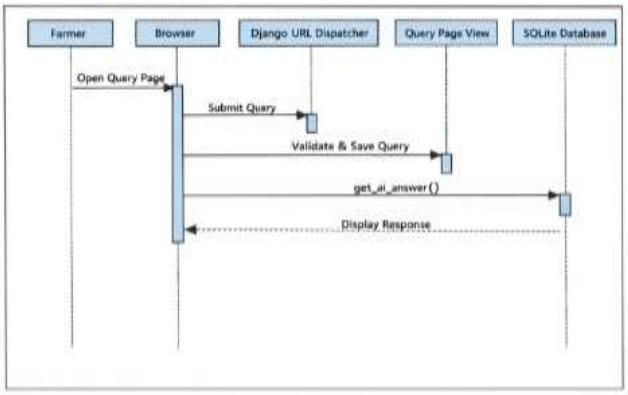

Fig. 5: Sequence Diagram of Query Submission

3.3.4 Activity Diagram

The Activity Diagram in Fig. 6 represents the system workflow. The process begins with farmer registration and login. Once authenticated, the farmer can browse crop advisory information or submit a query. If a query is submitted, the system processes the text using keyword matching, generates a relevant response, and stores the interaction. In the administrative workflow, the administrator updates crop details, manages records, and reviews past queries.

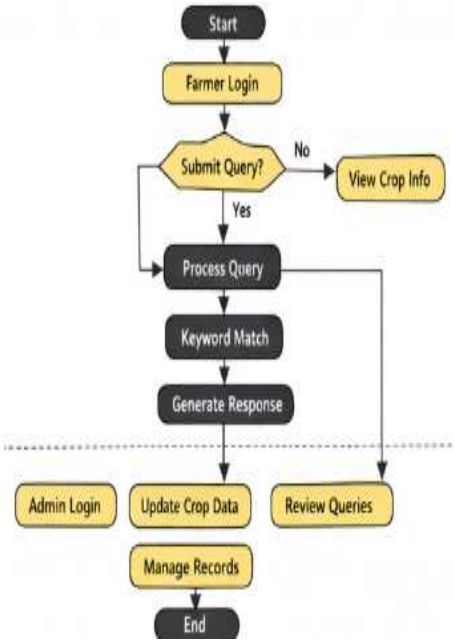

Fig. 6: Activity Diagram of Smart Agriculture Advisory System

4. Implementation

The Smart Agriculture Advisory System is implemented as a web-based application using Python and the Django framework, ensuring scalability, security, and efficient handling of user interactions. The system follows the Model-View-Template (MVT) architectural pattern, which separates data handling, business logic, and presentation layers for better maintainability and modularity. The frontend is developed using HTML, CSS, Bootstrap, and JavaScript, providing a responsive and user-friendly interface accessible across various devices.

4.1 System Modules

The system is divided into five core modules. The User Authentication Module ensures secure access through farmer registration and login functionality, session-based authentication using Django, password protection using hashing techniques, and restricted access to authorized users only. The Farmer Advisory Module provides access to agricultural information by displaying crop-related advisory data, providing information on fertilizers, irrigation, and pest control, and enabling farmers to browse crop details. The Query Processing Module is the core intelligent component that accepts farmer queries in text format, performs keyword extraction and matching, retrieves relevant responses from the database, and displays results in real time. The Admin Management Module allows administrators to add, update, and delete crop information, review farmer queries and

DOI: 10.30726/esij/v13.i2.2026.1320034

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

responses, maintain the knowledge base, and monitor system usage. The Database Management Module stores all system data including farmer registration details, crop advisory information, and query-response history. Django ORM is used for efficient data interaction and ensures data integrity.

4.2 Implementation Summary

The implementation demonstrates effective integration of web technologies (Django, HTML, CSS, Bootstrap), database management (SQLite, ORM), and intelligent query handling (keyword-based processing). The system is lightweight, scalable, and suitable for deployment in rural and resource-constrained environments. Its modular design allows easy enhancement and integration with advanced AI technologies in the future.

5. Results and Discussion

The performance and effectiveness of the proposed Smart Agriculture Advisory System were evaluated through functional testing, system performance analysis, and usability assessment. The evaluation focuses on how efficiently the system delivers real-time agricultural guidance and supports farmer interaction.

5.1 Functional Evaluation

The system successfully implements all required functionalities for an effective advisory platform. The farmer module allows users to register, log in, browse crop information, and submit queries without difficulty. The interface is intuitive, enabling farmers with minimal technical knowledge to navigate efficiently. The query processing module accurately identifies keywords from farmer queries and retrieves relevant responses from the knowledge base with immediate response generation. The admin module effectively manages crop data and advisory content dynamically.

5.2 System Performance Analysis

Fig. 7: Performance Analysis of Smart Agriculture Advisory System

The system was tested under moderate usage conditions. As shown in Fig. 7, query processing time is less than one second, system response is immediate due to keyword matching, database operations are efficient with SQLite, and the system supports multiple simultaneous users without delay. The lightweight design ensures high performance even on low-resource systems.

5.3 User Experience Evaluation

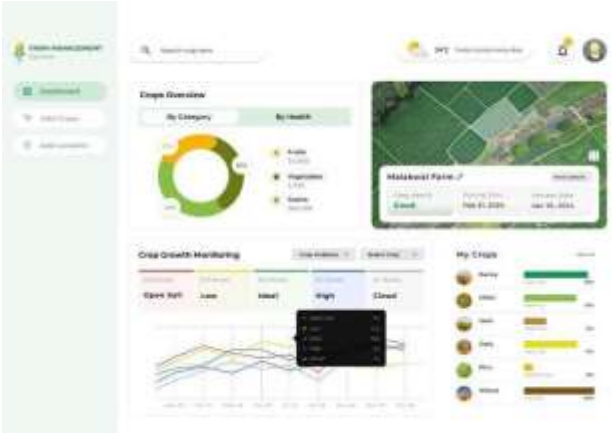

Fig. 8: User Interface of Smart Agriculture Advisory System

As illustrated in Fig. 8, the system demonstrates strong usability with a simple and clean interface suitable for rural users, easy navigation between modules, fast response that improves user satisfaction, and responsive design that supports mobile accessibility. Farmers can quickly access information and receive guidance without technical barriers.

5.4 Discussion

The results confirm that the proposed system effectively bridges the gap between farmers and agricultural knowledge. Compared to traditional advisory methods, the system offers instant access to information, a centralized knowledge base, and a real-time query-response mechanism. The keyword-based approach ensures efficiency but may limit response depth for complex queries. However, it provides a strong foundation for integrating advanced AI techniques in future versions.

6. Conclusion

The Smart Agriculture Advisory System provides an efficient solution for delivering timely and accessible agricultural guidance. It bridges the gap between expert knowledge and farmers through a simple web-based platform with features like registration, advisory browsing, and query handling. Built using Django, the system ensures a secure, scalable, and well-structured backend. Its keyword-based query engine enables instant responses, helping

DOI: 10.30726/esij/v13.i2.2026.1320034

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

farmers make quick decisions, while the admin module supports continuous updates. The platform performs well in usability and reliability, with a responsive design suitable for mobile devices. Overall, it is a cost-effective and scalable solution that improves agricultural productivity and supports digital transformation in agriculture.

7. Future Enhancements

While the system meets its current goals, several enhancements can improve its effectiveness and reach. Integrating AI and NLP can replace simple keyword matching, enabling better understanding of user queries and more accurate responses. Adding multilingual support would make the system more accessible to farmers from diverse linguistic backgrounds.

Integration with weather APIs and market price data can further support informed decision-making. Developing a mobile application with push notifications would improve usability and engagement. Features like image-based crop disease detection can enhance practical value. For scalability, migrating to cloud infrastructure with databases like PostgreSQL or MySQL can support larger deployments and more users.

References

[1] Django Software Foundation, “Django Documentation,” Version 5.x, 2024.
[2] Python Software Foundation, “Python Documentation,” 2024.
[3] Bootstrap Team, “Bootstrap 5 Documentation,” 2024.
[4] Mozilla Developer Network, “JavaScript Guide,” 2024.
[5] SQLite Consortium, “SQLite Documentation,” 2024.
[6] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 7th ed. Pearson, 2016.
[7] I. Sommerville, Software Engineering, 10th ed. Pearson, 2015.
[8] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Pearson, 2016.
[9] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[11] FAO, “Digital Agriculture Report,” 2022.
[12] Government of India, “Agricultural Extension Services Report,” 2023.
[13] OpenWeatherMap, “Weather API Documentation,” 2024.
[14] TensorFlow Developers, “TensorFlow Documentation,” 2024.
[15] Kisan Suvidha Portal, “Agricultural Information Services,” Government of India, 2023.

DOI: **10.30726/esij/v13.i2.2026.1320034**