

Agrisense AI in Agriculture

Moksha Pradha H¹, Monisha R², Sangamesh Nag B Y³, Shalini S M⁴, Prof. Husna Banu⁵

Students, Department of CSE ¹⁻⁴

Assistant Professor, Department of CSE ²

Kalpataru Institute of Technology, Tiptur, India

Abstract: *Agricultural decisions depend on timely and locally relevant information about crop selection, soil health, irrigation, plant disease, weather, market prices, and government schemes. These services are often distributed across separate sources, creating access and coordination challenges for farmers. This paper presents AgriSense AI, a web-based farmer-support platform designed to bring these functions together through a single interface. The system combines farmer registration and profiles, crop recommendations, leaf-image disease assessment, soil-health analysis, irrigation guidance, weather and market information, scheme discovery, and an administrative dashboard. A bilingual conversational assistant, Krishi Mitra, is included to support English and Kannada interactions. The application follows a React and Vite front end with a Node.js and Express REST API, MongoDB for persistent records, and a locally hosted Ollama model for assistant responses. External weather data can be obtained through OpenWeatherMap when an API key is configured. The report describes a working prototype and its design, implementation, and test approach; it does not establish field-level agronomic accuracy or measured improvements in farmer outcomes. The current disease assessment and crop recommendation components have documented constraints, so their outputs should be treated as decision support rather than verified agronomic advice.*

Keywords: Smart agriculture, farmer advisory, crop recommendation, plant disease assessment, soil health, bilingual AI assistant, web application.

I. INTRODUCTION

Agriculture involves interdependent decisions about crop choice, soil fertility, water use, pest and disease response, market prices, and public support schemes. In practice, relevant information may be spread across government portals, local advisory services, weather providers, and specialist applications. Farmers may also face language, connectivity, and digital-literacy barriers when using these resources. A single advisory interface can reduce the effort required to locate information, provided that it makes its assumptions and limitations clear.

AgriSense AI was developed as a web-based platform for farmers in Karnataka, with English and Kannada as the intended interaction languages. It brings several advisory and information functions into one application and provides separate access for farmers and administrators. The purpose of this work is to describe the system architecture, core modules, implementation choices, and evaluation boundaries documented in the project report.

II. PROBLEM STATEMENT

Farmers may need to consult different services for crop selection, soil-health interpretation, disease identification, weather forecasts, market prices, and government schemes. This fragmentation increases the effort required to collect information, while language barriers and limited digital familiarity can further reduce accessibility. The project addresses this problem through a centralized web application that combines these functions and provides a conversational interface.



The system is intended to support decisions rather than replace expert agricultural advice. Some functions are rule-based or dependent on external services, and the project report does not establish field-level validation or statistically verified accuracy for all modules.

III. METHODOLOGY

A. Requirement Analysis

The project report identifies farmer-facing requirements including registration and login, profile management, crop recommendations, soil-health guidance, leaf-image disease assessment, weather and market information, scheme discovery, and conversational assistance. Administrator functions include viewing registered users and accessing platform statistics. The design also considers bilingual interaction, role-based access, and fallback behaviour when the local language model is unavailable.

B. System Design

The application is organized as a web platform with a React and Vite client, a Node.js and Express backend, and MongoDB persistence. The client presents farmer-facing pages and administrator functions. The REST API handles application requests and connects the interface to backend services and stored records. This separation supports modular development and maintenance.

C. Core Modules

The crop recommendation module uses farmer and crop-context inputs to present suitable options and related guidance. The soil-health module accepts nutrient values such as pH, nitrogen, phosphorus, and potassium and maps them to a score and fertilizer guidance. The disease-assessment module accepts a leaf image and returns a disease-related output. Weather and market modules present external information, while the government-scheme module organizes scheme-related details. Krishi Mitra provides conversational assistance.

D. Implementation

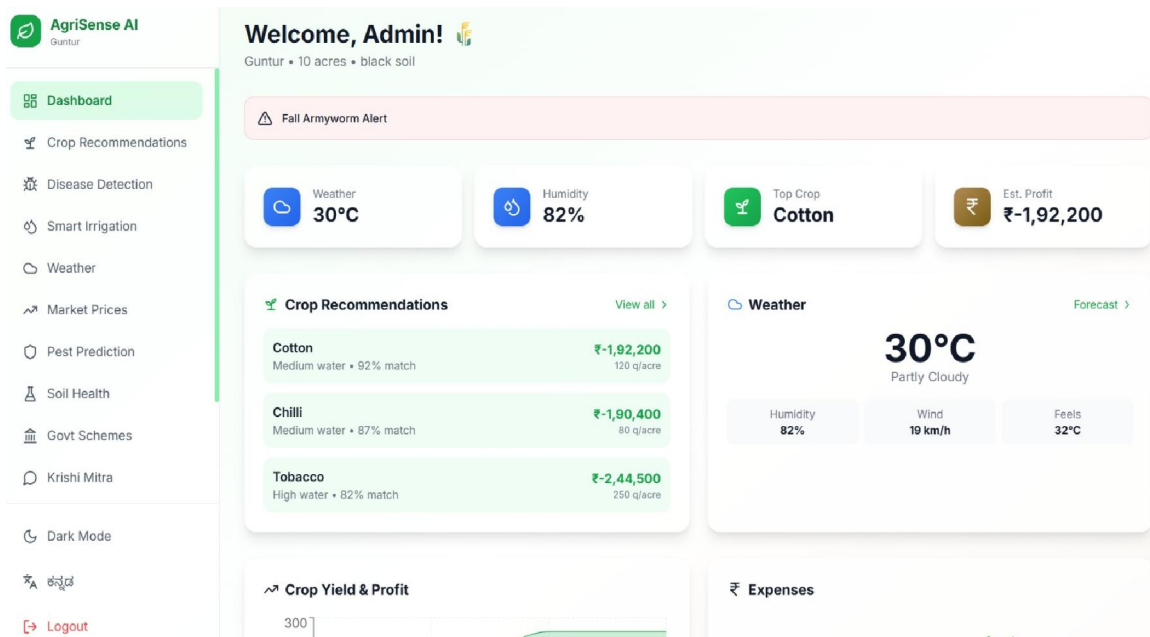
The user interface is implemented with React and Vite. The backend uses Node.js and Express to expose REST endpoints, and JSON supports communication between the client and server. MongoDB stores application records, including user/profile data and advisory-related records described in the project design. Authentication and role-based access are intended to separate farmer operations from administrative functions. Krishi Mitra uses a locally hosted Ollama model, and weather retrieval can use the OpenWeather API when configured.

E. Testing

The project report lists unit, integration, API, authentication, authorization, security, performance, and assistant-conversation test cases. These tests are intended to check individual functions, interactions between modules, and key user flows. The report distinguishes designed test cases from measured outcomes; not all performance targets or field-level effectiveness measures have been established.



IV. RESULTS AND DISCUSSION



The project integrates multiple farmer-support functions within a single web application. The intended user flow allows farmers to access profile features, crop and soil guidance, disease assessment, weather and market information, scheme details, and Krishi Mitra through one interface. Separate administrator functions support basic oversight of users and platform information.

The report describes a structured test plan covering core application flows and module interactions. However, the evidence should not be interpreted as proof of agricultural accuracy or field effectiveness. Crop recommendations are rule-based rather than a statistically validated predictive model. The disease-assessment component has not been established as field-validated, and results depend on input quality, available data, external API configuration, and local model availability.

Future evaluation should report measurable outcomes separately for each module. Useful measures include disease-classification performance on a labelled and regionally representative dataset, agreement of crop recommendations with expert assessments, consistency of soil-health guidance, API response times, and usability feedback from intended users. These evaluations would help distinguish implemented functionality from validated impact.

V. CONCLUSION

This paper presented AgriSense AI, a web-based smart agriculture advisory platform combining farmer profiles, crop recommendations, soil-health guidance, leaf-image disease assessment, weather and market information, government-scheme discovery, and a bilingual conversational assistant. Its modular architecture combines React, Vite, Node.js, Express, MongoDB, a locally hosted Ollama model, and an optional OpenWeather integration.

The platform demonstrates an integrated approach to organizing agricultural information, but it remains a prototype and should not be treated as a validated agronomic decision system. Further work should prioritize field validation, expert review, representative disease-image testing, measured usability and performance evaluation, and stronger evidence for the recommendations delivered to farmers.

VI. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our project guide and faculty members for their continuous support,

Copyright to IJARSCT
www.ijarsct.co.in



DOI: 10.48175/IJARSCT-40304



valuable suggestions, and encouragement throughout the development of this work. Their guidance helped us understand the concepts and complete the project successfully.

We also extend our appreciation to our institution for providing the necessary resources and a supportive learning environment. Finally, we thank all our friends and family members for their motivation and constant support during this project.

REFERENCES

1. S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming - A review," *Agricultural Systems*, vol. 153, pp. 69-80, 2017.
2. A Kamlaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70-90, 2018.
3. T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 1877-1901.
4. K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, Art. 2674, 2018.
5. N. Patel, D. Chittamuru, A. Jain, P. Dave, and T. S. Parikh, "AvaajOtal: A field study of an interactive voice forum for small farmers in rural India," in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, Atlanta, GA, USA, 2010, pp. 733-742.
6. Government of India, "PM-KISAN: Pradhan Mantri Kisan Samman Nidhi portal." [Online]. Available: <https://pmkisan.gov.in>
7. Government of India, Directorate of Marketing and Inspection, "Agmarknet: Agricultural marketing information network." [Online]. Available: <https://agmarknet.gov.in>

