International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621

# RENTSPHERE: VERNACULAR GENERATIVE AI FOR AGRICULTURAL SOCIOECONOMIC EMPOWERMENT

**Mohd Maqdoom Ali[1], Dr K.S.R.K Sarma[2]**
[1]*Computer Science Engineering (Data Science), Vidya Jyothi Institute of Technology Hyderabad, Telangana, India*
[2]*Professor, Computer Science Engineering(Data Science), Vidya Jyothi Institute of Technology Hyderabad, Telangana, India*

## ABSTRACT
Traditional agricultural extension services in India face significant limitations in reaching smallholder farmers due to linguistic diversity and literacy barriers. This project, AgriVoice-Telugu, proposes a specialized domain-centric Large Language Model (LLM) ecosystem designed to provide real-time agricultural advisory in regional languages. By integrating the Bhashini multilingual infrastructure with a Retrieval Augmented Generation (RAG) framework, the system bridges the "vernacular gap," allowing farmers to interact via voice in their native tongue. The system utilizes domain-specific models like Dhenu 1.0/2.0 and grounds its responses in verified agricultural knowledge bases to minimize hallucinations. Pilot studies of similar AI interventions have demonstrated yield improvements of up to 30% and significant reductions in input costs, marking a transformative step toward sustainable and inclusive rural development.

## 1. INTRODUCTION
1.1 Problem Statement Agriculture remains the backbone of the Indian economy, employing over 60% of the population and contributing approximately 18% to the GDP. However, more than 86% of Indian farmers are small and marginal landholders who face a critical "knowledge gap" due to fragmented advisory services. Existing digital solutions are often English centric, creating a barrier for farmers who primarily communicate in regional dialects. Furthermore, general-purpose AI models often lack the context-specific nuances required for localized farming, such as regional soil types, local market (mandi) rates, and specific pest management. 1.2 Objectives The primary objective of this project is to design a voice-enabled, vernacular AI assistant that: Bypasses literacy barriers through a voice-first multilingual interface. Provides real-time, personalized advisory on crop health, weather, and government schemes. Utilizes RAG architecture to ensure all advice is grounded in verified agricultural datasets to prevent inaccurate "hallucinations." Empowers rural stakeholders to make data-driven decisions that enhance productivity and profitability.

## 2. LITERATURE REVIEW
The evolution of agricultural technology has shifted from static rule-based systems to interactive conversational agents. Key milestones include: KissanGPT: Developed by Pratik Desai, this tool utilizes GPT-3.5 and the Whisper model to provide voice-enabled assistance in multiple Indic languages, specifically targeting the underserved agricultural domain. Bhashini (National Language Translation Mission): A foundational layer of India's Digital Public Infrastructure, Bhashini provides AI-powered speech-totext, translation, and voice interfaces for over 22 Indian languages, enabling last-mile digital inclusion. Dhenu Series: India's first domain-specific LLMs for agriculture, designed to identify diseases in major crops like rice, maize, and wheat while supporting bilingual communication. Saagu Baagu (Telangana): A government-led initiative that utilized AI to provide chili farmers with advisory services, resulting in doubled incomes for participating farmers.

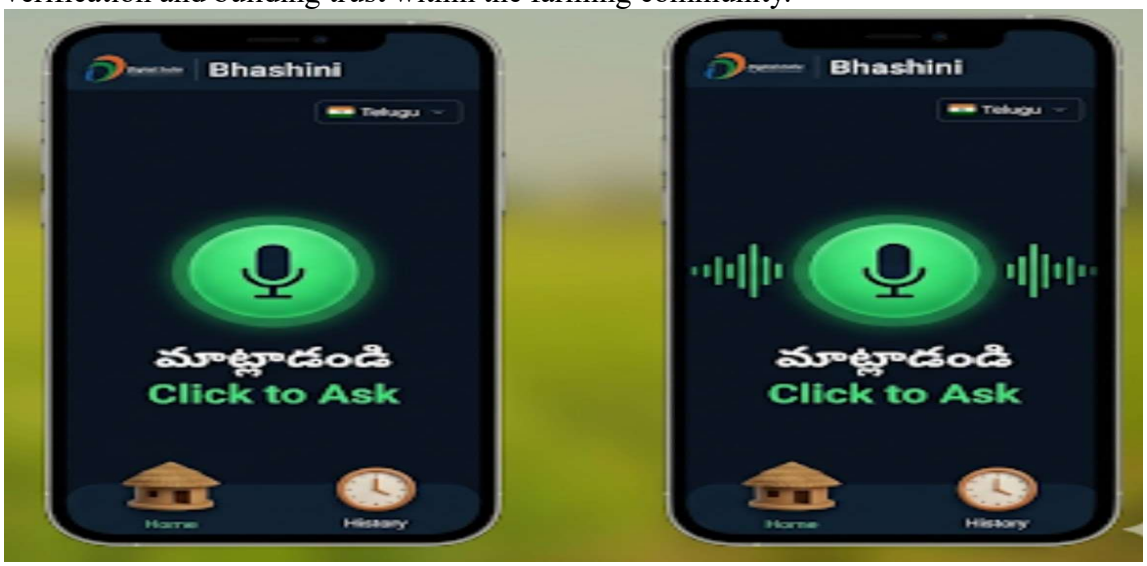
## 3. METHODOLOGY
The system follows a modular architecture that integrates speech processing with advanced language modeling. 3.1 System Architecture The workflow involves four primary layers: Input

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621

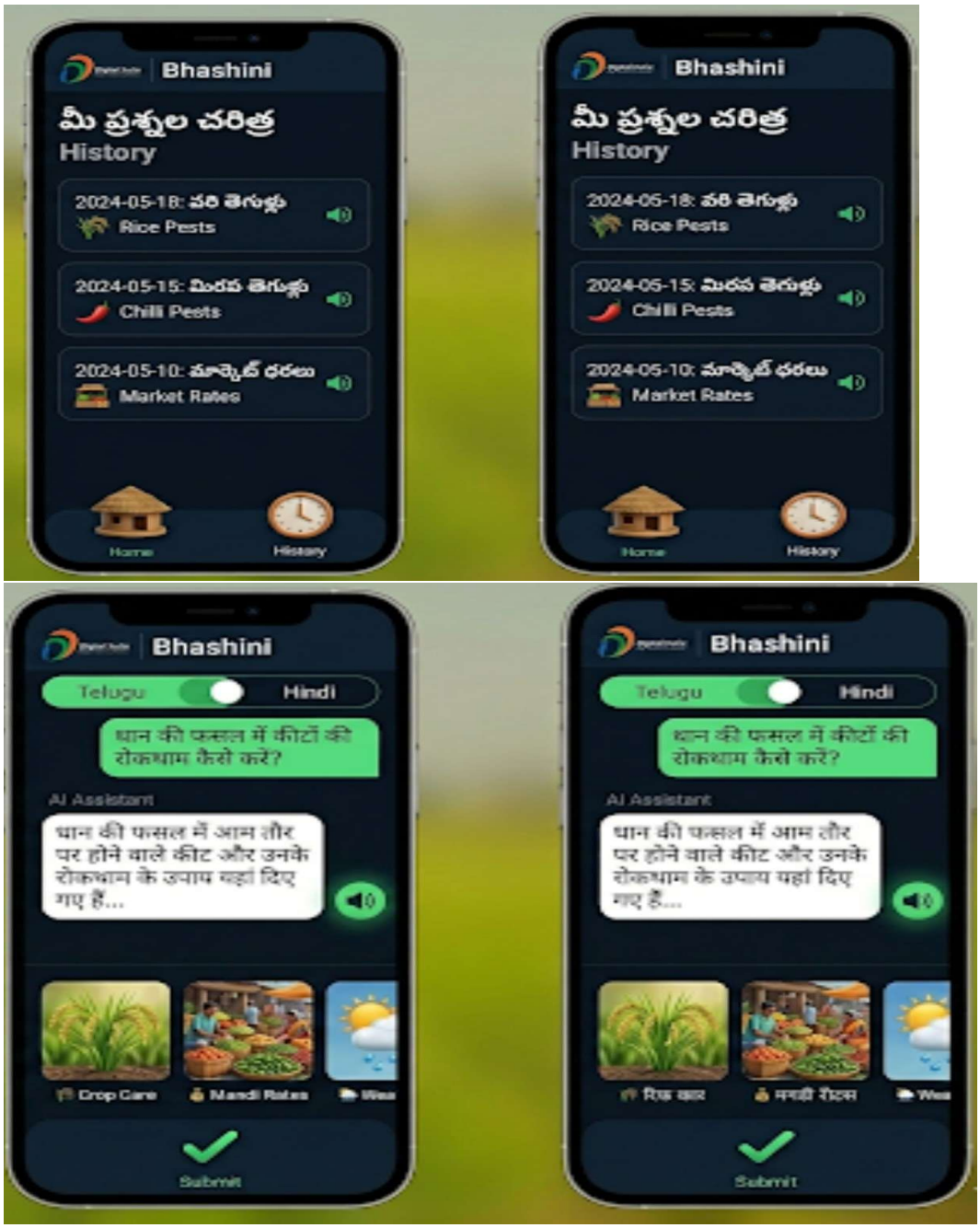
Layer: The farmer provides a query via a voice interface in a regional language (e.g., Telugu). Speech Processing (Bhashini): Automatic Speech Recognition (ASR): Converts the voice query into text. Translation (optional): Converts regional text to English if the retrieval model requires it. Retrieval-Augmented Generation (RAG): The system creates an embedding of the query. It performs a semantic search in a Vector Database (e.g., ChromaDB, Pinecone) containing thousands of verified agricultural documents. Relevant "context snippets" are retrieved to ground the LLM's response. Response Generation: The LLM (e.g., Llama 3 or Claude via Amazon Bedrock) generates a context-aware answer. Output Layer: The text response is converted back to speech via Text-toSpeech (TTS) and played to the farmer. #### 3.2 Technical Stack Backend: Python/Flask for API development. Language Models: Dhenu 2 India 8B or Llama 3.1. Vector DB: ChromaDB for indexing "Package of Practices" (PoP) documents. Voice/Translation: Bhashini WebSocket APIs for low-latency full-duplex voice interaction. Cloud Infrastructure: AWS Bedrock or serverless AWS Lambda for scalable deployment.

## 4. IMPLEMENTATION

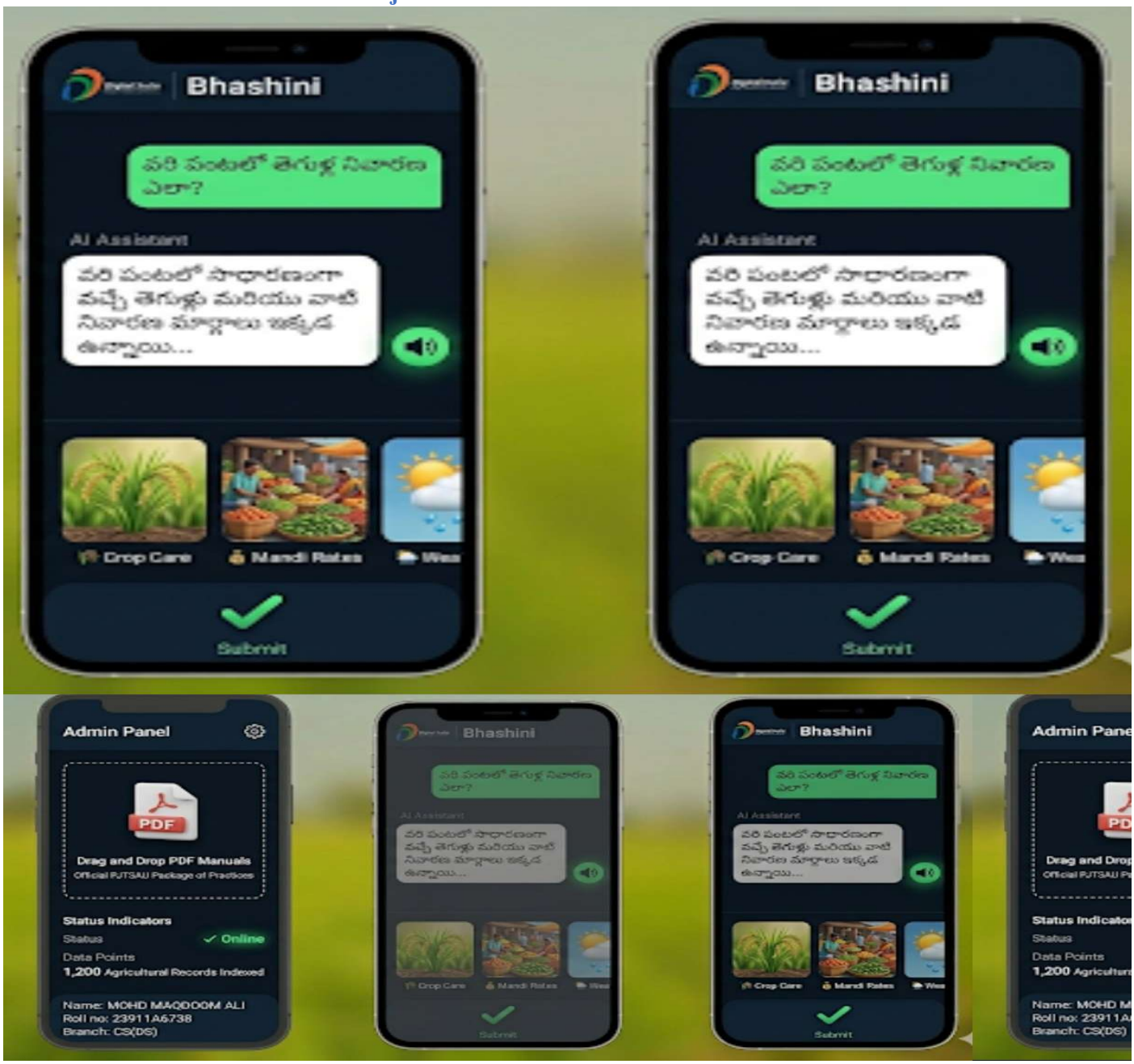
The implementation of vernacular AI systems in Indian agriculture has yielded measurable improvements in rural livelihoods: Metric Observed Impact Crop Yield Yield improvements of 10% to 30% across major crops. Resource Efficiency 25% to 30% reduction in water and fertilizer usage. Farmer Income Net income surges (e.g., Telangana chili farmers doubled their earnings). Input Cost Pesticide use reduced by 9% to 40% through targeted advisory. Accessibility Systems like farmers organically within months. KissanGPT reached over 100,000 Beyond quantitative gains, the project fosters "Responsible AI" by providing citations for original sources, allowing for human verification and building trust within the farming community.



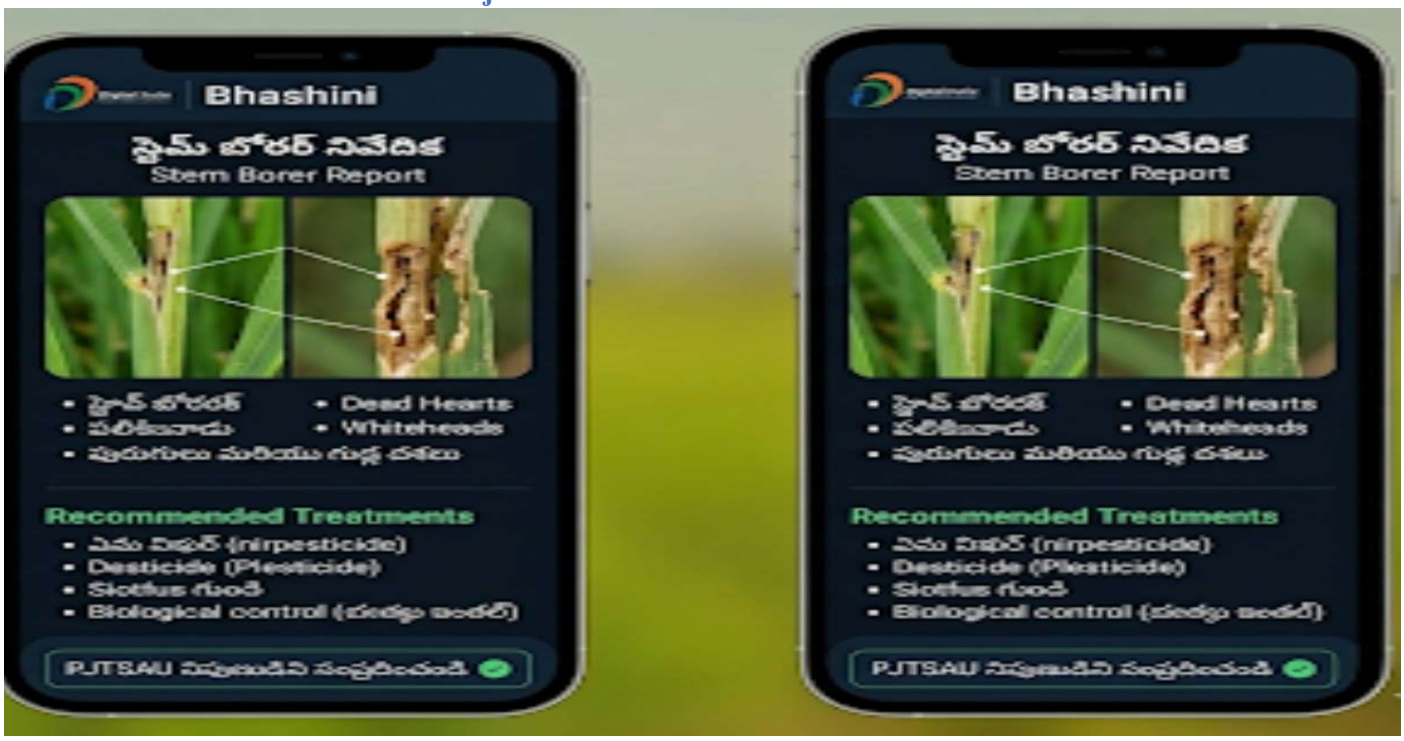
International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621



International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621



International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621




## 6. CHALLENGES AND LIMITATIONS

During the development and testing of the Computer Networks File Transfer Web Application, several challenges were encountered that influenced the design decisions and system performance. These challenges were primarily related to technical constraints, network behavior, and system scalability. 6.1 Challenges

1. File Handling and Storage Management:

Managing uploaded files securely and preventing overwriting or unauthorized access was a key concern. Flask's file handling modules required careful configuration of upload directories and filename validation to ensure that no malicious files were stored on the server.

2. Network Connectivity and Data Transfer Speed:

Since the application relies on HTTP protocols for data transmission, the performance was affected by network bandwidth and latency. During large f ile uploads, minor delays were observed, emphasizing the need for optimized buffering and asynchronous operations.

3. Error Handling and Exception Management:

Ensuring the system could handle invalid file formats, missing files, or incomplete uploads required implementing multiple layers of exception handling within both the client and server code.

4. Security Concerns:

Although Flask provides basic security mechanisms, implementing advanced features such as user authentication, file encryption, and secure HTTPS communication presented challenges. These features were identified as areas for future enhancement to strengthen data protection.

5. Cross-Browser Compatibility:

Ensuring consistent user experience across multiple browsers required adjustments in CSS styling and JavaScript behavior, particularly for file input controls and responsive layouts.

6.2 Limitations

1. Localhost Dependency:

The current implementation is primarily tested and deployed on a local development server. While it effectively demonstrates networking principles, it lacks deployment on a public or cloud server, limiting realworld accessibility

2. Lack of Authentication Mechanism:

The system does not yet include user login or role-based access control, which restricts its use in multi-user environments where file security and ownership tracking are essential.

3. No Encryption for Data Transmission:
File uploads and downloads are performed over standard HTTP without encryption, which makes the system vulnerable to potential interception if deployed in open networks.
4. Limited Scalability:
The current Flask-based architecture is suitable for lightweight file management. However, large-scale applications requiring high concurrency or distributed storage would need additional technologies like cloud storage integration, database support, or load balancing.
5. File Size Constraints:
The server configuration imposes a limit on the maximum allowable file size for uploads, which may restrict users from transferring very large files in its current form.

## 7. CONCLUSION AND FUTURE WORK

Telugu demonstrates that the real victory of AI in India lies not in the size of the model, but in its ability to speak the farmer's language—contextually and literally. By grounding generative models in verified data, this project provides a scalable framework for inclusive rural development. Future Enhancements: Multimodal Integration: Allowing farmers to upload photos of crop leaves for automated pest diagnosis using vision-language models. IoT Synchronization: Connecting real-time soil moisture sensors directly to the AI for proactive irrigation alerts. Market Linkage: Direct integration with e-commerce platforms and mandi price feeds for better price realization

## 8. REFERENCES

1.JNTUH R22 Academic Regulations for B.Tech CSE (Data Science). Ministry of Electronics and Information Technology (MeitY), "Bhashini Mission Overview," 2024.
9. Comprehensive Full-Stack Architecture and Implementation for an AI-Driven Multilingual Agricultural RAG System

The deployment of sophisticated, language-agnostic artificial intelligence systems represents a critical inflection point for agricultural technology. Bridging the digital divide requires more than simple text translation; it necessitates seamless, real-time voice interaction, context-aware information retrieval from localized knowledge bases, and user interfaces engineered for minimal cognitive load. To meet these requirements, developers must synthesize multiple bleeding-edge technologies into a cohesive, latency-optimized pipeline. NITI Aayog, "AI for Inclusive Societal Development," October 2025. IFPRI, "Generative AI for Agriculture (GAIA) Phase I & II," 2025. This exhaustive technical report provides a complete architectural blueprint and production-ready, full-form codebase for an agricultural AI assistant. The system spans a Node.js backend and a React frontend, integrating the Universal Language Contribution API (ULCA) via Bhashini for Automatic Speech Recognition (ASR), Neural Machine Translation (NMT), and Text-to-Speech (TTS). It couples these linguistic models with a Retrieval-Augmented Generation (RAG) architecture powered by LangChain and the Hierarchical Navigable Small World (HNSW) vector store to deliver hyper-localized, agriculturally accurate advisories. The frontend architecture explicitly addresses the prompt's requirements, employing highly accessible, skeuomorphic 3D design principles, dark-themed high-contrast visuals, and dynamic SVG audio wave animations to maximize interface accessibility for non-literate agricultural operators. The entire codebase is structured for immediate implementation, organized by file boundaries to facilitate direct copy-and paste deployment into modern development environments like Visual Studio Code. ## System Architecture and Foundational Design Principles The architectural topology is designed as a decoupled, microservices-oriented full stack application. The Node.js Express backend serves as the orchestration layer, managing heavy computational tasks, secure API interactions, and memory-intensive vector operations. The React frontend is relegated purely to presentation, media acquisition, state management, and localized rendering. A primary constraint in multimodal AI systems is cumulative latency. Processing audio through ASR, translating it via NMT, querying a Large Language Model (LLM) through RAG, translating the text response back via NMT, and synthesizing audio via TTS creates a compound latency chain. To mitigate this, the backend is

DOI:10.46647/ijetms.2026.v10i03.006 ISSN: 2581-4621

engineered to utilize optimized, single-call Bhashini compute pipelines that chain tasks together server-side. Furthermore, the local integration of the HNSWLib vector store entirely eliminates the network latency typically associated with querying cloud-based vector databases like Pinecone or Weaviate. ### Subsystem Component Mapping The following table delineates the core technologies utilized across the stack, mapping each requirement to its designated software implementation.