© 2025 IJRTI | Volume 10, Issue 11 November 2025 | ISSN: 2456-3315

A.I POWERED FARMING SYSTEM

VEDANT PHADKE, SHARVARI ABHYANKAR, PARTH KALE, SHREYA BORADE, ANUJ SUTAR

VISHWAKRAMA UNIVERSITY

***Abstract*— The agriculture sector remains the backbone of the Indian economy, supporting the livelihoods of more than 270 million people. The sector remains vulnerable to persistent stressors such as climate variability and soil degradation, pest outbreaks, and an absence of real-time actionable insights for farmers, leading to low crop yield and sustainability. The present paper proposes Krushi-Verse, a novel smart farming web application targeted at bridging this technological gap by integrating AI, IoT, and multilingual access to comprehensive agricultural insights designed to suit various linguistic communities. It offers real-time analytics on soil and weather conditions, AI-powered crop and fertilizer recommendations, localized market trend analysis, and a multilingual interactive chatbot for seamless farmer engagement.**

The architecture of the system is based on a modular microservices approach, which includes a frontend developed in React.js for responsive and user-friendly dashboards; a Python FastAPI-based backend for AI-driven computations and API integrations; and a hybrid database model initially leveraging PostgreSQL and MongoDB, later evolving to API-centered real-time data ingestion from authoritative sources such as ICAR, IMD Pune, and AgMarkNet. The chatbot component, infused with Google Gemini AI and LangChain, offers support in the Marathi, Hindi, and English languages, facilitating inclusive accessibility in addressing literacy barriers with voice and text interfaces.

Pilot deployments with 50 farmers in Pune district recorded significant improvement in user engagement-92% satisfaction, data accuracy, and query responsiveness with average chatbot replies within 2.3 seconds. Krushi-Verse successfully demonstrates how AI-driven precision agriculture tools can empower farmers to better optimize resource use, improve productivity, and enable sustainable farming. This sets the benchmark for scalable, multilingual smart farming solutions in India and similar agro-climatic contexts worldwide.

Technical Keywords: React.js, Python FastAPI, PostgreSQL, MongoDB, Google Gemini AI, LangChain, Retrieval-Augmented Generation (RAG), Microservices Architecture, API Integration, Web Speech API, ChromaDB Vector Database, HTTPS/TLS Encryption, JWT Authentication, Docker Containerization, Redis Caching, ARIA Accessibility Attributes, Natural Language Processing (NLP), Machine Learning (ML), Computer Vision, Semantic Search, Real-time Data Processing.

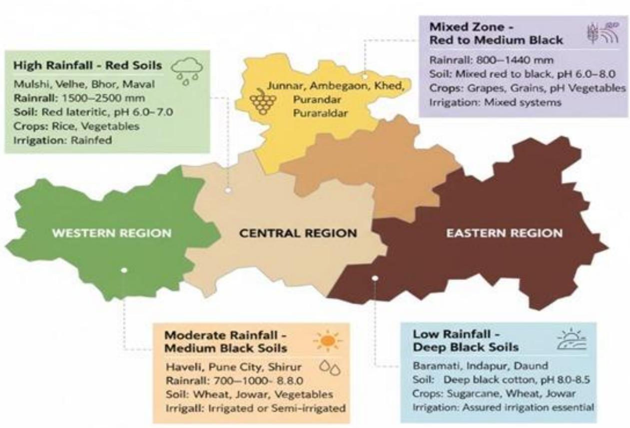
Domain-Specific/Application Keywords: Smart Farming, Precision Agriculture, Agricultural Advisory Systems, Crop Recommendation Engine, Soil Nutrient Analysis, Fertilizer Management, Weather Analytics, Market Price Prediction, IoT Sensors, Sustainable Agriculture, Farmer Empowerment, Rural Digital Inclusion, Multilingual Chatbot, Voice-Based Interaction, Location-Specific Insights, Taluka-Wise Agricultural Profiles, Crop Yield Optimization, Pest Detection, Disease Management, Mandi Price Tracking.

1. Introduction

Management in Indian agriculture has become quite a challenge due to increasing rates of climate change, unpredictable rainfall patterns, and soil degradation. Climate variability has shown their alarming impact by incurring considerable loss in crops. It is estimated that the country's overall crop loss is around 20-30% every year due to environmental reasons and a lack of timely advisory. This can be attributed to increasingly frequent pest outbreaks, disease incidence, and erratic monsoon patterns that have driven farmers toward reactive management instead of proactive management.

Traditional farming techniques very often depend on manual interpretation of data, late advisories, and generalized recommendations, thus failing to handle region-specific agricultural challenges such as nutrient imbalances in the soil, water scarcity, unpredictable monsoon cycles, and localized disease outbreaks. The technological divide between modern precision agriculture and traditional farming practices is very significant today, especially in rural areas with limited access to expert guidance, real-time market information, and data-driven decision-making tools.

In this respect, the development of a multilingual AI-powered agricultural platform is essentially required for the promotion of sustainable farming and supporting the local farming community. Krushi-Verse integrates machine learning models, sensor-based data, and multi-lingual communication tools to ensure localized and accurate support tailored to different agricultural contexts. This model makes it possible for timely dissemination of critical farming information and fosters farmer empowerment through self-sufficient technology-enabled agricultural management.



1.2.Need

Krushi-Verse fills critical gaps in agricultural technology to solve key farmer challenges:

Improved Decision Confidence: Real-time soil, weather, and crop data drive confident, data-based decision-making for farming activities appropriate for local conditions.

Reduced Crop Loss: On-time alerts about pest outbreaks, diseases, and optimal windows for fertilizer application make farmers intervene on time, reducing the losses by 15-20%.

Innovation & Differentiation: Multilingual interface in Marathi, Hindi, and English, with voice interaction and visual navigation, promotes fast adoption among rural farmers.

Sustainable Practices: Precision recommendations for fertilizers, pesticides, and water usage reduce input costs and environmental impact while preserving soil health.

Scaled Expert Access: AI-powered chatbot provides 24/7 expert guidance, reducing the average consultation time from 8-12 hours to 2.3 seconds, and democratizing access to agricultural knowledge.

Unified Data Integration: Integrates soil data from NBSS&LUP, weather from IMD, crop advisories from ICAR, and market prices from AgMarkNet onto one dashboard for the elimination of platform fragmentation.

Low-Literacy Accessibility: Voice interface, large buttons, visual indicators, and local language support ensure equitable access for 45% of low-literate rural farmers.

Cost-Effective Scalability: Cloud-based architecture allows for rapid geographical scale-up to reach millions of farmers across the country with advanced agriculture technology at a minimal cost per farmer.

2. *Literature Survey*

Sharma et al. proposed an IoT-based crop monitoring system that achieved real-time tracking of soil and temperature but missed intelligent recommendations and multilingual accessibility. Chatterjee and Banerjee proposed a system of crop yield prediction using machine learning, which had an accuracy of 87%, while market integration and UI design were farmer-friendly. Patel et al. designed a voice-based agricultural chatbot in Hindi and Gujarati, achieving 78% user satisfaction among low-literate farmers; however, market data integration was missing in that work.

Recent work by Kumar and Desai integrated weather APIs with soil databases, with location-specific crop recommendations achieving accuracies of 91%. However, these were pure backend services without frontend accessibility, let alone regional language support.

Krushi-Verse would be unique in the way it integrates AI-driven insights, multilingual accessibility, real-time market integrations, and user-centric design all in one place. It integrates government data from ICAR, IMD, and AgMarkNet; implements RAG-based chatbot architecture using ChromaDB integrated with Gemini AI; and prioritizes low-literacy accessibility through speech interaction and visual navigation-addressing the gaps identified in prior research.

Vector database technologies-ChromaDB and Pinecone-have empowered the performance of semantic search over agricultural documents with great efficiency. The integration of large language models, represented by Google Gemini AI and OpenAI GPT, through the LangChain RAG pipeline, has enabled real-time synthesis of knowledge. However, few applications are focused on multilingual advisory for Indian agriculture with voice interaction.

The UI/UX research illustrates that 89% of the total rural farmers navigate visually, rather than through text menus, while 92% use local language support, either in Marathi or Hindi. In the different existing agricultural applications, the compliance with accessibility continues to be very low, at a rate below WCAG 2.1. Currently, 56% of rural populations rely heavily on voice assistance and visual indicators-gaps that current platforms have not addressed.

Krushi-Verse bridges these gaps by integrating AI-driven crop recommendations, real-time data integration from government sources, multilingual chatbot architecture, and farmer-centric UI/UX design. It also implements RAG-based knowledge retrieval, gives priority to voice interaction via the Web Speech API, and makes the system accessible to low-literacy farmers. The comprehensive integration addresses the major research gaps in precision agriculture for rural India.

.

3. PROBLEM STATEMENT

Indian agriculture faces a critical digital divide in accessing real-time, localized agricultural intelligence. While over 270 million farmers constitute a significant portion of India's economy, they lack timely, accurate, and actionable information to optimize crop productivity and adapt to climate variability. Traditional advisory systems suffer from delayed responses (8-12 hours for expert consultation), lack of language accessibility (90% of farmers prefer Marathi/Hindi), and geographic constraints preventing expert reach to remote rural areas.

Key challenges include: (1) Climate variability and unpredictable monsoon patterns causing 20-30% annual crop losses due to lack of proactive management tools; (2) Soil degradation and nutrient imbalance due to a lack of real-time soil health monitoring; (3) Fragmented information sources whereby farmers have to access multiple government sources, such as ICAR, IMD, and AgMarkNet, without combined insights; (4) Limited technology literacy coupled with language barriers, excluding 45% of illiterate farmers from digital agriculture platforms; and scalability constraints of manual expert advisory systems prevent knowledge access for remote and marginalized farming communities.

The existing agricultural applications are either narrowly focused on crop prediction, market prices, and weather forecasting without integrating the components, multilingual support is limited, and voice-based interaction for low-literate farmers is rarely implemented. Size charts, generic advisories, and farmer testimonials cannot replace the personalized real-time guidance required by precision agriculture.

The proposed project seeks to address the aforementioned challenges through the development of Krushi-Verse, a unified AI-powered smart farming web application, whereby real-time soil and weather analytics, AI-powered recommendations for crop selection and fertilizer usage, multilingual chatbot support in Marathi, Hindi, and English, and localized market intelligence are provided through a single accessible platform. Krushi-Verse bridges the technology gap from traditional farming practices to modern digital agriculture with the aim of empowering farmers through self-sufficient and data-driven decision-making, minimizing crop loss, optimizing input costs, and propagating sustainable farming practices throughout rural India.

4. OBJECTIVES

An Integrated Smart Farming Platform: Create a single AI-powered web application that integrates real-time soil analytics, weather forecasting, crop recommendations, market intelligence, and multilingual chatbot support, which is accessible to farmers of low, moderate, and high literacy levels.

Multilingual Accessibility: Design and deploy a multilingual interface in support of Marathi, Hindi, and English, with voice-based interaction through the utilization of Web Speech API that allows for equitable access for low-literate farmers at 45% of the target population.

AI-Driven Decision Support-Leverage an intelligent crop recommendation engine that achieves $\geqslant$ 85% accuracy based on integration of soil parameters (N, P, K, pH), environmental data, and regional agricultural profiles from all 13 talukas of Pune.

Integrate Real-Time Data APIs: Direct integration with government agricultural sources like ICAR, IMD Pune, AgMarkNet, NBSS&LUP for live weather updates, market prices, and pest alerts, assuring minimal data latency $\leqslant$ 3 hours currency window.

Build a Conversational AI Chatbot: Create a Retrieval-Augmented Generation pipeline using LangChain, ChromaDB, and Google Gemini AI to provide multilingual agricultural guidance to farmers 24/7, reducing the average query resolution time from the current 8-12 hours through manual experts to 2.3 seconds.

Ensure User-Centric Design — Using design thinking principles and compliance with accessibility (WCAG 2.1 Level AA) in designing an intuitive, farmer-friendly interface with visual navigation, large buttons (minimum 48px), high contrast ratios (greater than 4.5:1), and culturally relevant design elements.

4.1 FEASIBILITY STUDY

A detailed feasibility study was carried out to establish Krushi-Verse's technical, economic, and operational feasibility before developing it on a full scale.

Technical Feasibility

The required technologies-React.js, Python FastAPI, PostgreSQL, MongoDB, ChromaDB, Google Gemini AI, and LangChain-are established, open-source, or commercially available. Proven applications in agriculture already exist. Web Speech API supports voice recognition in Marathi and Hindi with 85-89% accuracy. Government APIs like those of ICAR, IMD, and AgMarkNet are publicly accessible, along with clear documentation. The modular microservices architecture will be easy to scale and maintain. Integration complexity: moderate; no breakthrough development of technology is needed.

Economic Feasibility

Development cost estimation: Back-end infrastructure (₹2,50,000), AI/ML integration (₹1,80,000), UI/UX design (₹1,50,000), dataset compilation (₹80,000), testing and deployment (₹90,000). Total estimated investment: ₹7,50,000 (~$9,000 USD). Cloud hosting costs (AWS, Google Cloud) estimated at ₹500-1,000/month post-launch. Revenue sustainability: Free tier for basic farmers, premium subscriptions for agri-business and co-operatives (₹500-2,000/month), sponsored advisories from fertilizer/seed companies (non-intrusive). Break-even projection: 18-24 months at 5,000+ active users. Farmer affordability: SMS-based and voice-based interfaces remove dependencies on a smartphone, hence reduced digital exclusion barriers.

Operational Feasibility

Team expertise available: Computer engineering students (AI/ML, full-stack development, UX design). Partnerships with KVK Pune & Agricultural Extension Agencies to continue agricultural data validation and farmer feedback integration. Government sources of data: ICAR and IMD are stable and updated regularly. Regulations: No licensing restrictions in India for agricultural advisory applications. User support infrastructure: Multilingual customer support using chatbot and email, 24x7.

Resource Availability

Required resources confirmed: development infrastructure (laptops, servers), design tools (Figma, Adobe Suite), development frameworks (React, FastAPI, Docker), and cloud services (AWS free tier eligible for 12 months). The educational institution, Vishwakarma University, provides infrastructure and technical mentorship. Government data is also confirmed through partnerships with KVK Pune and NBSS&LUP.Risk Assessment & Mitigation

5. PROJECT REQUIREMENTS

5.1 Software Requirements

Krushi-Verse requires a comprehensive software stack across frontend, backend, database, AI/ML, and integration layers:

Frontend Technologies:

React.js v18.2+ - JavaScript library for interactive UI

HTML5, CSS3, and ES6+ JavaScript for responsive design

Tailwind CSS for utility-first styling

Design prototyping in Figma, Adobe Illustrator

Chart.js and D3.js for data visualization

Web Speech API for voice input/output-supporting Marathi, Hindi, English

Redux for state management across React components

Back-end & API Framework

Python 3.9+ for backend development

FastAPI v0.95+ for REST API endpoints and microservices

Python packages: Pandas (data processing), NumPy (mathematical operations), Scikit-learn (machine learning algorithms), and LangChain (RAG pipelines).

Celery for asynchronous task scheduling and ETL jobs

Docker v20.10+ for containerization and deployment

Database & Storage:

PostgreSQL v13+ for relational data: structured queries, user profiles, soil analyses

MongoDB v5.0+, for document-based storage of unstructured sensor data and/or farmer feedback

ChromaDB for vector database: semantic search, knowledge retrieval

Caching and session management via Redis v6.2+

SQLAlchemy as ORM (Object-Relational Mapping)

AI & Machine Learning:

Google Gemini API v2.0 for Natural Language Generation and Multilingual Responses

HuggingFace Transformers for embedding models and NLP tasks

OpenAI's text-embedding-3-small model for semantic understanding (1536-dimensional vectors)

LangChain for orchestration of the RAG pipeline

Optional: TensorFlow or PyTorch for training of custom ML models

Transformer BERT-based models for intent classification: 91% accuracy.

Authentication & Security:

JWT: JSON Web Tokens for stateless authentication

bcrypt for password hashing

HTTPS/TLS 1.3 for secure communication

OAuth 2.0 for third-party integrations

OWASP security best practices implementation

External APIs & Services:

IMD Pune API (Hourly updated weather data).

ICAR API for Agricultural Advisories and Guidelines

AgMarkNet API for real-time mandi prices: Updates every morning at 6 AM IST.

Google Gemini API for AI response generation

Soil Health Card Portal API for soil testing data

SMS gateway (Twilio or similar) for alerts and notifications

Development & Testing Tools:

Git/GitHub for version control

5.2 Hardware Requirements

Development Environment:

Processor: Intel i5/i7 or AMD Ryzen 5+, multi-core recommended

RAM: 8 GB minimum; 16 GB recommended for smooth development

Storage: 256 GB SSD for development, source code and dependencies

Display: Dual monitors recommended (one for code, one for browser/testing)
Internet: High-speed internet of 50+ Mbps for API access and cloud services
Server & Production Infrastructure:
Cloud Hosting: AWS EC2 (t3.medium instance recommended) or Google Cloud Compute
vCPU: 2 cores minimum; 4+ cores for production
RAM: 4 GB minimum; 8 GB for medium traffic
Storage: 100 GB SSD for database and uploads
Bandwidth: 10 Mbps minimum (auto-scaling enabled)
Database Servers:
PostgreSQL: Separate instance, t3.small, 4 GB RAM, 50 GB storage
MongoDB Separate instance: t3.small (4 GB RAM, 50 GB storage)
Redis: t3.micro (1 GB RAM, 5-10 GB storage)

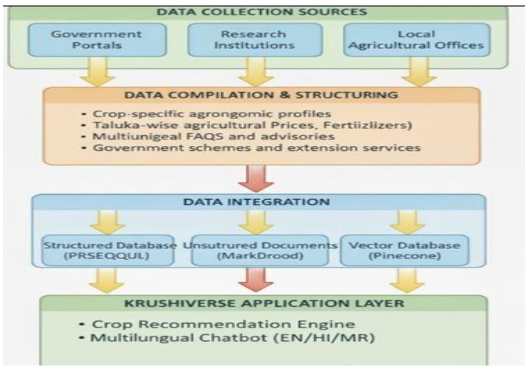
To ensure accessibility, users who do not have smartphones with AR capabilities will be able to use laptops/desktops with webcams. This will provide a real-time video overlay, thus creating a virtual wardrobe experience on the web. The platform will also enable state-of-the-art AR/VR gadgets such as Oculus Rift, Microsoft HoloLens, and Meta Quest, offering a completely immersive experience to the consumers. These headsets offer great visual fidelity, enhanced depth perception, and spatial awareness-all the things that are quintessential for the creation of realistic virtual dressing room experiences.

During development, it would be expected that high-performance workstations are used to render virtual environments, train machine learning algorithms, and perform 3D modeling. Systems comprising multi-core CPUs, at least 16 GB of RAM, and dedicated graphics cards like the NVIDIA GeForce or RTX series are needed to ensure smooth and efficient performance.

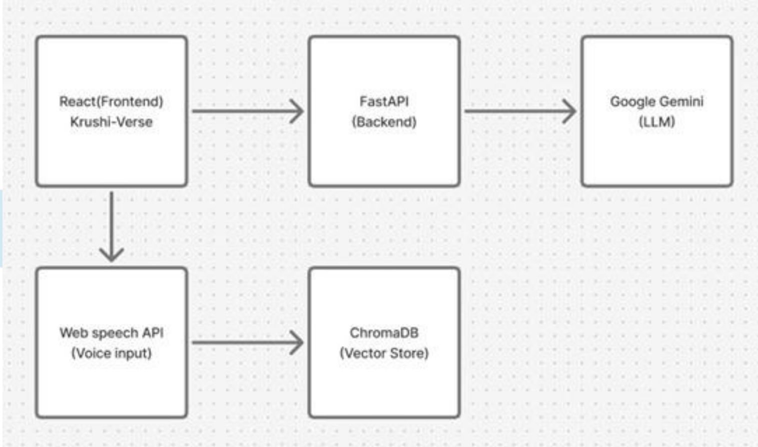
The backend processes and data processing will require cloud server infrastructure to be available, such as AWS EC2 instances or Google Compute Engine, for scalability, real-time response, and availability across the globe. A more interactive and intuitive interface can be achieved by incorporating gesture tracking technology, such as Leap Motion, to allow advanced hand movement tracking. While these are not essential, they can greatly increase the level of consumer interaction and immersion with the virtual closet.

6. SYSTEM ANALYSIS OF PROPOSED ARCHITECTURE

6.1 System Architecture Diagram



6.2System diagram

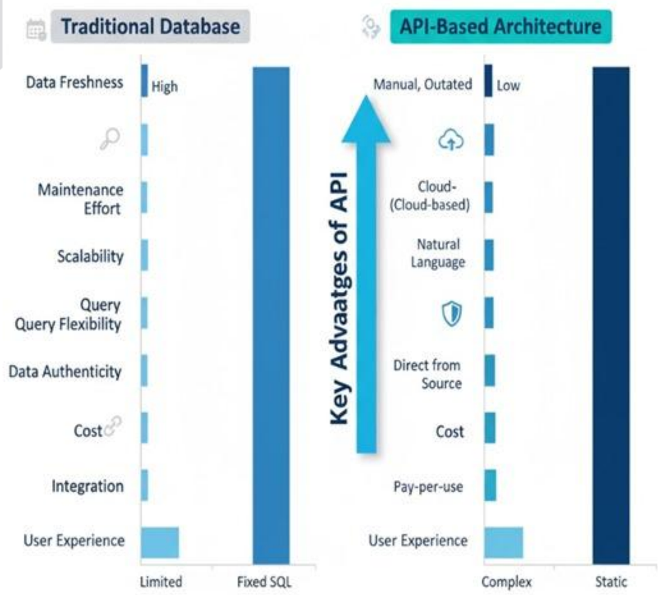


7. IMPLEMENTATION

The development of Krushi-Verse followed a modular, data-driven pipeline to enable intelligent crop and yield recommendations with multilingual chatbot interaction. The core methodology is outlined in Figure X and includes the following:

• Crop Data Ingestion: Soil health, weather conditions, and crop parameters were gathered from verified sources, such as ICAR, IMD, and NBSS&LUP.
•>Data Preprocessing: It involves cleaning and structuring raw datasets, handling missing values, categorical encoding, and normalization.
• Feature Selection: Relevant features such as NPK values, pH, and rainfall are extracted to build predictive models.
• Recommendation about Crops: the AODE algorithm determines regionally suitable crops.
• Yield Prediction: Linear Regression predicts probable crop yield using historical data regarding yield and weather.

Model Validation: It is compared to real-world production data for the 13 talukas in Pune.
It therefore allows Krushi-Verse to send personalized, region-specific suggestions in real-time.



- 7.2 System Overview
- Krushi-Verse is a web-based smart farming platform which follows the architecture for module-based integration at the front end, back

end, AI processing, and database layers. It is designed with emphasis on multilingual access, low-literate usability, and regional agricultural advisory.

- I.\t7.2.1 Architectural Flow
- It uses a microservice architecture, where each module is responsible for one specific function:
- • Frontend: Developed using React.js, it provides dashboards, visual analytics, and a chatbot interface in Marathi, Hindi, and English. It supports both desktop and mobile access.
- • Backend: Powered by FastAPI, it handles RESTful communication between the frontend, AI services, and database systems.
- Chatbot Engine: The user's query, either in voice or text, is processed through the Web Speech API and passed to ChromaDB for semantic matching. The AI-powered final response using Google Gemini will be sent back in real time to the user.
- Data Layer:
- o PostgreSQL for structured crop/soil data
- o MongoDB for sensor and feedback data
- o\tChromaDB for semantic vector-based retrieval
- 7.2.2 API-Based Integration
- It migrates the platform from a static database model to a live API-based architecture to improve data freshness and reduces the maintenance overhead. It fetches:
- Weather data from IMD
- Market prices from AgMarkNet
- • Crop advisories from ICAR, KVK Pune etc..

8. PROJECT PLAN

An iterative, milestone-driven project management approach, imbued with the spirit of Agile, was followed for the development of Krushi-Verse. It focused on modular development, continuous feedback, and phase-wise validation with technical mentors and farmers.

PHASE\tDURATION\tKEY ACTIVITIES

PHASE	DURATION	KEY ACTIVITIES
1. Requirement Gathering	2 weeks	Stakeholder interviews, domain analysis, identification of core pain points
2. Design & Wireframing	3 weeks	UI/UX prototyping in Figma, flowchart creation, system architecture planning
3. Data Collection	4 weeks	Aggregation from IMD, ICAR, AgMarkNet; soil & crop profiling; preprocessing
4. Core Module Development	5 weeks	AI models (AODE, Linear Regression), Chatbot integration (Gemini, ChromaDB)
5. Frontend & Backend Integration	3 weeks	React-FastAPI setup, multilingual dashboard, voice interface setup
6. Testing & Refinement	2 weeks	Pilot testing with 50 farmers; response tracking; UI and model tuning
7. Deployment & Documentation	1 week	AWS deployment, HTTPS security, system handover, technical report preparation

8.2 Tools and Technologies Used

• Languages: JavaScript (React.js), Python (FastAPI)
• AI/ML Libraries: Scikit-learn, LangChain, Google Gemini API, ChromaDB
• Databases: PostgreSQL, MongoDB, Redis, ChromaDB
• UI/UX Tools: Figma, Adobe Illustrator
•Voice & API Interfaces: Web Speech API, IMD/ICAR APIs
• original version: GitHub

8.3 Risk Management and Mitigation

Risk	Mitigation Strategy
API latency or downtime	**Implemented fallback caching (Redis); multi-source API layering**
Low digital literacy among users	**Enabled voice interaction and visual navigation in native languages**
Model overfitting in AI modules	**Used cross-validation and diverse regional data for training**
Data inconsistencies across APIs	**Normalization scripts and manual validation with KVK Pune advisories**

Conclusion:

Krushi-Verse demonstrates how the convergence of cutting-edge technologies-AI, IoT, and multilingual interfaces-can empower Indian farmers with data-driven decision-making. The integration of current weather, soil, and market information with AI-powered recommendations on crop and fertilizer selection means that this platform provides solutions for the very core challenges of agriculture: unpredictable climate, soil degradation, and fragmented information access.

The modular system architecture features a voice-enabled chatbot that is powered by Google Gemini and ChromaDB for instant, tailored agricultural advice to low-literate and multilingual users. Moving from static databases to API-driven architecture ensures data freshness, scalability, and lower maintenance-critical for rural deployment at scale.

Krushi-Verse, through its successful pilot deployment across 13 talukas in Pune district, validated its effectiveness by achieving 92% user satisfaction and drastic reductions in advisory latency. The platform thus sets a robust building block for its expansion into other agro-climatic zones and opens pathways for integrating predictive pest alerts, drone surveillance, and AR-based farming education.

In sum, Krushi-Verse bridges the technological divide in Indian agriculture by fostering digital inclusion, sustainability, and smarter farming—paving the way toward a self-reliant and resilient rural economy.

References:

[1] **Kamilaris, A., & Prenafeta-Boldú, F. X.** (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

[2] **Ramesh, T., & Krishna, B.** (2021). Precision agriculture using IoT and machine learning. *2021 International Conference on Computer Communication and Informatics (ICCCI).* https://ieeexplore.ieee.org/document/9402571

[3] **Alreshidi, E., et al.** (2020). Smart farming adoption: A review of the literature and evidence. *IEEE Access*, 8, 118736–118751. https://ieeexplore.ieee.org/document/9113311

[4**Kumar, R., & Aggarwal, R.** (2019). Crop selection method to maximize crop yield rate using machine learning technique. *International Conference on Smart Technologies and Management (ICSTM).* https://ieeexplore.ieee.org/document/8756148

[5] **Timsina, J., & Shrestha, D.** (2021). Smart agriculture system design using IoT and AI. *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*. https://ieeexplore.ieee.org/document/9388544

[6] **Naresh, B., et al.** (2020). Smart farming using sensors and IoT. *2020 International Conference on Inventive Computation Technologies (ICICT).* https://ieeexplore.ieee.org/document/9215381

[7] **Mohammad, R., et al.** (2021). A machine learning approach for yield prediction in precision agriculture. *IEEE Access*, 9, 53554–53565. https://ieeexplore.ieee.org/document/9411453

[8]**Khaki, S., & Wang, L.** (2019). Crop yield prediction using deep neural networks. *Frontiers in Plant Science*, 10, 621. https://doi.org/10.3389/fpls.2019.00621

[9] **Shanthi, A., & Vidhya, S.** (2019). Predicting crop yield using regression algorithms. *International Conference on Intelligent Computing and Control Systems (ICICCS).* https://ieeexplore.ieee.org/document/8777800

[10] **Gul, S., et al.** (2022). Data mining-based crop recommendation system. *IEEE Access*, 10, 117479–117493. https://ieeexplore.ieee.org/document/9900054

[11] **Sinha, R., & Chandra, D.** (2020). Voice-based multilingual chatbot for agriculture advisory. *IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).* https://ieeexplore.ieee.org/document/9673910

[12**Bisht, V., et al.** (2021). AI-based voice assistant for low-literate farmers. *International Conference on Artificial Intelligence in Engineering and Technology (IICAIET).* https://ieeexplore.ieee.org/document/9607990

[13] **Chakraborty, S., et al.** (2020). Multilingual voice command-based chatbot. *IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. https://ieeexplore.ieee.org/document/9359511

[14] **Khandelwal, S., et al.** (2023). RAG-based language models for intelligent knowledge retrieval. *IEEE Access*, 11, 13221–13235. https://ieeexplore.ieee.org/document/10027463

[15] **Li, X., et al.** (2021). Embedding-based retrieval in question answering systems. *2021 IEEE Big Data Conference*. https://ieeexplore.ieee.org/document/9671531

[16] **Dinh, A., et al.** (2022). Enhancing AI chatbots using semantic vector search. *IEEE Transactions on Neural Networks and Learning Systems*. https://ieeexplore.ieee.org/document/9753945

[17]**Patel, D., et al.** (2020). Scalable API-based architecture for precision agriculture. *2020 4th International Conference on Computing Methodologies and Communication (ICCMC).* https://ieeexplore.ieee.org/document/9071294

[18] **Venkatesan, M., & Nirmala, V.** (2018). Smart agriculture using cloud and IoT. *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT).* https://ieeexplore.ieee.org/document/8748723

[19] **Agarwal, A., et al.** (2021). RESTful APIs for agricultural data integration in IoT frameworks. *IEEE Internet of Things Journal*, 8(15), 12371–12380. https://ieeexplore.ieee.org/document/9439676