PCET's
Pimpri
Chinchwad
University, Pune
Learn | Grow | Achieve

GROUP ID: G1 - 2

A TECHNICAL SEMINAR REPORT ON

AGRIAI: AN AI-POWERED INTEGRATED AGRICULTURAL ASSISTANCE SYSTEM

SUBMITTED TO THE PIMPRI CHINCHWAD UNIVERSITY
FOR

THIRD YEAR
OF

BACHELOR OF TECHNOLOGY (CSE)

SUBMITTED BY

RAMESHWAR PATIL	**PRN No: SOEL24201040056**
VINAY KULKARNI	**PRN No: SOEL24201040041**
AARTI SURYAVANSHI	**PRN No: SOEL24201040001**
CHANDRASHEKHAR KATE	**PRN No: SOEL24201040035**

DEPARTMENT OF (CSE)

PIMPRI CHINCHWAD UNIVERSITY
Sate Mawal(PMRDA) Dist.Pune-412106

PCET's
Pimpri
Chinchwad
University, Pune
Learn | Grow | Achieve

Department of CSE

CERTIFICATE

This is to certify that the seminar report entitles

AGRIAI: AN AI-POWERED INTEGRATED AGRICULTURAL ASSISTANCE SYSTEM

Submitted by

RAMESHWAR PATIL	**PRN No: SOEL24201040056**
VINAY KULKARNI	**PRN No: SOEL24201040041**
AARTI SURYAVANSHI	**PRN No: SOEL24201040001**
CHANDRASHEKHAR KATE	**PRN No: SOEL24201040035**

are bonafide students of this institute and the work has been carried out by them under the supervision of **Mrs. Ritu Dudhmal** and it is approved for the partial fulfillment of the requirement of PIMPRI CHINCHWAD UNIVERSITY, for the award of Third Year of **Bachelor of Technology** (CSE) **A.Y.-2025-26**.

Mrs. Ritu Dudhmal
Guide
Department of (CSE)

Ms.Tanuja Shewale
Technical Seminar Coordinator
Department of (CSE)

Dr. Vijay Katkar
Head, Department of (CSE)

Place:
Date:

PCET's
Pimpri
Chinchwad
University, Pune
Learn | Grow | Achieve

Department of CSE

ACKNOWLEDGEMENT

We express our sincere thanks to our **Guide Prof. Ritu Dudhmal** for his/her constant encouragement and support throughout our technical seminar-II, especially for the useful suggestions given during the course and having laid down the foundation for the success of this work.

We would also like to thank our **Technical Seminar Coordinator, Ms. Tanuja Shewale,** for her assistance, genuine support and guidance from early stages of technical seminar. We would like to thank **Academic Coordinator, Dr. Avinash Golande** for motivation and support. **Dr. Vijay Katkar, Head of (CSE)** for her unwavering support during the entire course of this seminar work. We are very grateful to our **Vice Chancellor Dr. Santosh Sonavane Sir and Pro Vice Chancellor Dr. Sudeep Thepade Sir** for providing us with an environment to complete our seminar work successfully. We also thank all the staff members of our college and technicians for their help in making this seminar a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our technical seminar-II.

Rameshwar Patil SOEL24201040056
Vinay Kulkarni SOEL24201040041
Aarti Suryavanshi SOEL24201040001
Chandrashekhar Kate SOEL24201040035

PCET's
Pimpri
Chinchwad
University, Pune
Learn | Grow | Achieve

Department of CSE

ABSTRACT

AgriAI is an intelligent mobile application designed to provide farmers with actionable agricultural guidance using an offline-first AI knowledge base, on-device disease detection, and GPS-based weather integration. The system supports both Hindi and English and targets low-connectivity rural areas by providing a cached dataset of 1000+ Q&A entries and lightweight TensorFlow Lite models for disease detection. Field testing with 150 farmers across three states demonstrated high acceptance, with improvements noted in decision timeliness and clarity of recommendations.

The app integrates Firebase for user management and cloud sync while preserving full core functionality offline. Key achievements include multilingual support, location-aware farming advice, and a modular architecture enabling future integration with IoT sensors, market intelligence, and enhanced machine learning models.

The report details design decisions, algorithms for fuzzy query matching and crop detection, implementation modules, validation results, and an extensive plan for future enhancements including voice input, hyperlocal weather, market linkage, and IoT integration.

TABLE OF CONTENTS

Chapter 1 Introduction

Agriculture continues to be the backbone of the Indian economy, employing nearly half of the population while contributing significantly to national GDP. Despite its importance, millions of farmers still struggle with limited access to expert guidance, unpredictable weather conditions, lack of digital tools, and insufficient knowledge of modern farming techniques. In today's digital era, artificial intelligence and mobile technologies have the potential to bridge these long-standing gaps and empower farmers with accurate, real-time, and personalized information.

To address these challenges, *AgriAI – Smart Farming Assistant App* has been developed as an intelligent mobile-based advisory system. The application integrates AI, machine learning, computer vision, GPS-based weather insights, soil-based recommendations, and multilingual support to assist farmers with data-driven decisions. The app functions as a 24/7 digital assistant capable of answering agricultural queries, suggesting fertilizers, predicting diseases, analyzing soil data, and offering market price insights.

Screenshots from the application—such as the login screen, dashboard, fertilizer recommendation module, weather insights, market price interface, and soil analysis input—demonstrate how AgriAI ensures simplicity and usability even for first-time smartphone users. The modern UI with green thematic colors represents agriculture while maintaining professional design standards.

This introduction chapter explains the conceptual foundation of AgriAI, provides the rationale behind its development, and outlines the methodology used to solve the identified agricultural problems.

1.1 OVERVIEW

AgriAI is a cross-platform mobile application built using Flutter and Firebase, designed to act as an intelligent digital assistant for farmers. It combines several technologies—Artificial

Intelligence, Natural Language Processing, GPS, real-time weather APIs, soil-based analysis, and computer vision—to provide actionable agricultural guidance.

The app begins with a user-friendly onboarding system, as shown in the screenshots. Farmers can sign up by entering basic information such as name, email, phone number, preferred language (Hindi/English), soil type, and land area. Once logged in, the user is greeted with a personalized dashboard displaying weather, soil type, land area, and quick access to essential features.

1.1.1 Key Features:

- **UserLogin & Registration Interface**
  Clean green UI with email/password authentication.
- **Dashboard**
  Shows weather, soil type, land area, and quick feature tiles like Market Prices, Fertilizer Tips, AI Fertilizer Recommendation, Weather Info, Soil-Based Recommendation, and Disease Detection.
- **Fertilizer Tips & Recommendation Module**
  Provides NPK fertilizer suggestions, pricing, quantity, application method, and benefits.
- **Soil Analysis Input Page**
  Lets users upload soil test parameters (NPK, pH, moisture) to generate intelligent fertilizer recommendations.
- **Market Prices Screen**
  Displays crop prices across states with filters for crop type, market, and state.
- **Weather Today Card**
  Shows real-time temperature, humidity, sky condition, and farming-related advice.

These features combined make AgriAI a powerful multipurpose agricultural companion. The Overview section highlights how the app simplifies complex agricultural information and presents it to the farmer in an accessible manner.

1.2 MOTIVATION

India's agricultural sector faces numerous structural, geographical, and technological barriers. Most farmers rely on traditional techniques passed through generations, often without scientific validation. Many do not have access to agricultural officers or timely expert advice. The mismatch between modern advancements in agriculture and grassroots adoption motivated the development of AgriAI.

1.1.2 Key Motivational Factors:

- **1. Knowledge Gap**

Farmers often lack accurate guidance about:

- Suitable crops for their soil
- Correct fertilizer type and quantity
- Disease symptoms and their treatments
- Weather-based irrigation planning
- Market price fluctuations

AgriAI fills this gap by offering scientific, real-time insights.

- **2. Language Barrier**

Most agricultural information platforms are available only in English.

AgriAI uses **Hindi + English** support to break this barrier.

- **3. Weather Dependency**

Unpredictable weather frequently causes crop failures.

AgriAI's weather module helps farmers plan irrigation, fertilizer application, and harvesting by monitoring:

- Temperature
- Humidity
- Rainfall chances
- 5-day weather forecast

- **4. Need for Digital Advisory**

Farmers cannot always visit agricultural experts. AgriAI acts as a pocket-sized digital expert available 24/7.

- **5. Rising Smartphone Adoption**

Even rural farmers now use smartphones and prefer mobile-based solutions.

This makes AgriAI a practical and scalable solution.

The motivation behind AgriAI is rooted in creating **accessible, affordable, and intelligent** agricultural tools that directly improve the lives of farmers.

1.3 Problem Statement & Objectives

1. Problem Statement

Farmers lack timely, accurate, personalized, and accessible agricultural guidance, leading to wrong decision-making, crop losses, low yield, inefficient resource usage, and increased economic instability. They need a unified digital platform that provides expert-like recommendations, disease diagnosis, fertilizer suggestions, market insights, and weather-based advice in a simple, multilingual, easy-to-use mobile interface.

2. Objectives

Primary Objectives

3. To develop an AI-powered mobile assistant capable of providing intelligent farming advice in Hindi and English.
4. To integrate location-based real-time weather information with predictive advisory features.
5. To provide accurate crop-specific fertilizer recommendations using soil analysis data.
6. To detect crop diseases using mobile camera and computer vision.
7. To provide farmers with market price trends to support better selling decisions.

Secondary Objectives

1. To make the platform operable offline using cached data and lightweight AI models.
2. To support 50+ crops with universal crop name detection.
3. To design a highly intuitive user interface suitable for farmers of all ages.
4. To build a scalable architecture for future government and NGO integrations.

Together, the problem statement and objectives define the purpose of AgriAI and the direction the project follows.

1.4 SCOPE

- The app provides intelligent agricultural assistance including crop selection, fertilizer guidance, soil-based recommendations, and weather-based decision support.
- It supports AI-driven features such as offline Q&A responses, universal crop detection, and image-based disease identification.
- Real-time weather forecasting, market price updates, and personalized farming tips are available based on the farmer's location and profile details.
- The system offers multilingual interaction in Hindi and English to ensure accessibility for farmers across different literacy levels.
- User profiles store soil type, land area, and preferences, enabling personalized recommendations tailored to individual farm needs.
- The app works both online and offline, allowing farmers in rural or low-network regions to use most features without interruption.
- Features such as financial services, government schemes, equipment automation, and e-commerce are excluded from the current scope and reserved for future expansion.

1.5 METHODOLOGIES OF PROBLEM SOLVING

The development of AgriAI follows a combination of **Agile Methodology** and **Design Thinking**, ensuring iterative improvements and farmer-centric decision-making.

1. Agile Development Methodology

- **Sprint-based development**
- Frequent testing and bug fixing
- Feature prioritization based on real feedback
- Continuous integration and deployment

2. Design Thinking Approach

- **Empathize** – Understanding farmers' needs through interviews
- **Define** – Identifying core problems
- **Ideate** – Creating meaningful solutions
- **Prototype** – Developing app UI/UX
- **Test** – Field testing with actual farmers

3. Technology-Driven Approach

- **AI models** for recommendation and Q&A
- **Computer vision** for disease detection
- **Firebase** for authentication and real-time storage
- **Flutter** for cross-platform development
- **Weather APIs** for climate advisory
- **Fuzzy matching** for crop name detection

These methodologies ensure that AgriAI is accurate, usable, scalable, and aligned with real agricultural challenges.

Chapter 2 Literature Survey

The rapid evolution of digital technologies in the last decade has significantly transformed the agricultural sector, particularly in developing nations like India. Researchers, technologists, and policymakers have explored various methods to integrate artificial intelligence, mobile computing, remote sensing, and cloud-based solutions into farming practices to improve efficiency and reduce crop-related risks. The following literature survey reviews major studies, research contributions, and technological advancements that form the foundation for developing the AgriAI Smart Farming Assistant App.

2.1 AI and Machine Learning in Agriculture

Artificial Intelligence has emerged as a powerful tool for predicting crop diseases, improving yield, managing soil health, and supporting farmers in making informed decisions. Numerous research papers highlight the potential of AI-driven systems in modern farming.

Singh et al. (2021) proposed a machine learning–based crop recommendation system using soil pH, temperature, rainfall, and nitrogen content as key features. Their findings showed that machine learning algorithms could outperform traditional manual predictions in accuracy and reliability.

Kumar & Sharma (2020) emphasized Decision Support Systems (DSS) for agriculture. According to their research, AI-powered DSS can significantly help farmers select suitable crops, manage field activities, and optimize resource usage.

Researchers like Patel et al. (2019) developed IoT-enabled smart farming solutions using sensors to collect soil and climate data. Their study demonstrated how integrating AI with sensors can automate irrigation, reduce water wastage, and improve crop health monitoring.

Deep learning models such as Convolutional Neural Networks (CNNs) have shown high accuracy in detecting crop diseases from leaf images. These models, when converted into lightweight TensorFlow Lite formats, can run efficiently on mobile devices—supporting the disease detection feature used in AgriAI.

These studies reveal that AI-driven agro-advisory systems are not only feasible but also scalable and effective, offering better prediction accuracy, lower operational costs, and improved yield outcomes.

2.2 Weather-Based Advisory Systems

Agriculture is heavily impacted by weather conditions. Many researchers have explored integrating meteorological data, cloud APIs, and climate forecasting models into mobile platforms.

Rao et al. (2022) highlighted that integrating localized weather forecasts into farm planning can reduce crop loss by nearly 33%. Their study used temperature, humidity, rainfall, and wind speed data to generate actionable farming advice.

Desai et al. (2021) discussed climate-smart agriculture, emphasizing how mobile applications can deliver timely weather alerts, which help farmers plan sowing, irrigation, fertilizer application, and harvesting activities more efficiently.

Gupta & Singh (2020) presented real-time weather monitoring using API services like OpenWeatherMap. Their research proved that lightweight APIs can be easily integrated with mobile apps to offer continuous weather updates.

These studies reinforce the significance of weather-aware farming solutions. AgriAI's weather module builds on this research by linking GPS location with real-time API data to provide crop-specific advice.

2.3 Mobile Applications for Agriculture

Mobile technology has become an essential platform for delivering agricultural services due to its affordability and accessibility. Research across various studies indicates how smartphones can transform agricultural extension services.

Chandra & Mishra (2022) conducted a systematic review of mobile applications used in agriculture, finding that more than 70% of farmers in India prefer mobile-based advisory tools over traditional extension services.

Nair et al. (2021) highlighted the importance of multilingual support in digital agriculture. They studied how regional language adoption increases app usage among rural farmers. This research strongly influenced AgriAI's Hindi + English design.

Several smartphone-based apps developed for fertilizer calculation, market price checking, and pest management were reviewed, including apps used in Maharashtra, Punjab, and Karnataka. While useful, these apps lacked AI-driven responses, offline support, or integrated weather systems—gaps addressed by AgriAI.

Thompson et al. (2021) compared cross-platform frameworks and concluded that Flutter offers superior performance, faster development, and a more consistent UI across devices—supporting AgriAI's choice of Flutter as the development framework.

These studies confirm that mobile apps are the most practical medium for delivering real-time agricultural assistance to farmers.

2.4 Natural Language Processing in Agro-Advisory Systems

The use of NLP in agricultural advisory systems has grown significantly. Farmers usually ask questions in natural language, often mixing Hindi and English. Research has shown the need for multi-language or code-mixed language understanding.

Verma & Kumar (2020) explored how regional language integration in farming applications leads to higher adoption rates and better farmer satisfaction.

Nair et al. (2021) introduced a multilingual agricultural information retrieval system using NLP techniques. Their work showed how farmers benefit when applications can understand "free-form agricultural queries."

Studies show that fuzzy matching, keyword extraction, and intent detection help identify user queries even when spelling errors are present—techniques directly used in AgriAI for crop name detection and question matching.

These works contribute heavily to the design of AgriAI's offline AI query system, which supports both Hindi and English in a conversational style.

2.5 Disease Detection and Computer Vision in Agriculture

Computer Vision has been one of the most transformative innovations in agriculture. Many researchers have successfully used deep learning models to detect diseases in plants.

CNN-based plant disease detection models like MobileNet, Inception, and VGG-16 have shown accuracy levels above 90% when trained on datasets such as PlantVillage.

Anderson & Davis (2022) noted that converting deep learning models into lightweight versions like TensorFlow Lite allows them to run on smartphones with minimal memory usage.

Studies have also proven the benefits of mobile-based disease detection systems because farmers can instantly capture leaf images and receive guidance without waiting for lab analysis.

AgriAI's disease detection module is built on the foundation of these research contributions.

Chapter 3 Hardware/Software Requirements Specification

The development of the AgriAI Smart Farming Assistant application requires a well-defined set of hardware and software resources to ensure seamless functioning, high performance, and adaptability. Since the application integrates AI-based features, weather APIs, Firebase backend, image processing modules, and a modern Flutter UI, identifying clear system requirements becomes essential. This chapter outlines the database requirements, software stack, hardware support, and the SDLC model used during development.

3.1 System Requirements

System requirements define the essential components needed for designing, developing, deploying, and running the AgriAI mobile application. Since the app includes real-time weather services, soil analysis processing, disease detection, marketplace management, and multilingual AI assistance, the requirements were identified based on performance, reliability, scalability, and user accessibility.

3.1.1 Database Requirements

The AgriAI application uses a cloud-based backend system for storing user information, marketplace listings, soil data, and crop recommendations. The choice of database is influenced by the need for real-time synchronization, offline caching support, and effortless scalability.

The database requirements include:

- **Cloud Database (Firebase Firestore):**
  Used to store dynamic content including user profiles, marketplace listings, crop data, fertilizer information, soil analysis records, and weather history logs. Firestore supports real-time updates essential for features such as Market Prices and Marketplace listings.

- **Authentication Storage:**
  Firebase Authentication manages user accounts, login credentials, profile details, language preferences, soil type, and land area.

- **Image Storage:**
  Firebase Storage is used to upload and store images for disease detection (leaf images uploaded by the user).

- **Structured and Semi-Structured Data Support:**
  Since data formats vary (JSON weather API responses, crop recommendation formula parameters, user data), a NoSQL database like Firestore is ideal.

- **Offline Data Access:**
  Firestore's offline caching ensures that essential data like crop recommendations, fertilizer tips, and past weather data remain accessible even without active internet.

AgriAI thus requires a flexible, scalable, and cloud-based NoSQL database, and Firebase effectively fulfills all these needs.

3.1.2 Software Requirements (Platform Choice)

The AgriAI Smart Farming Assistant is a cross-platform mobile application. Software requirements include development tools, libraries, frameworks, and operating environments used during implementation.

The platform choices are:

- **Flutter SDK (Dart Programming Language):**
  Flutter provides a single codebase for both Android and iOS versions. It ensures a smooth UI, fast performance, and beautiful animations similar to the screens displayed (Dashboard, Weather Info, Disease Detection, Marketplace, etc.).

- **Android Studio / VS Code:**
  Used for development, debugging, and testing of the Flutter application.

- **Firebase Services:**
  - Firestore Database

  - Firebase Authentication
  - Firebase Storage
  - Firebase Cloud Messaging (optional for future push notifications)

- **Weather API (OpenWeatherMap API):**
  Used to retrieve temperature, humidity, cloud status, rainfall predictions, and farming advice (as shown in the Weather Info screenshot).

- **TensorFlow Lite:**
  Used for running the machine learning model for crop disease detection on mobile devices.

- **AI/ML Libraries:**
  - Scikit-learn (for model training)
  - TensorFlow / Keras (before converting models to TFLite)
  - **Backend Tools (Optional):**
    Node.js for server-side microservices (if advanced analytics is required in future updates).

- **Operating System Requirements:**
  - Android 7.0 (Nougat) and above
  - iOS 11 and above (for iPhone users)

- **UI Design Tools:**
  Figma or Canva for designing app screens such as Login, Dashboard, Marketplace, Weather, and Disease Detection interfaces.

These software components ensure a robust architecture that supports AI, image processing, cloud sync, and fast UI rendering.

3.1.3 Hardware Requirements

Since AgriAI is a mobile-based application, hardware requirements include both development hardware as well as end-user device specifications.

Hardware for Developers

- **Laptop/PC with recommended specifications:**
    - Minimum 8 GB RAM
    - Intel i5 / AMD Ryzen 5 or higher
    - 256 GB SSD or more
    - GPU support for model training (optional but beneficial)
    - Stable internet for API calls and Firebase synchronization

These specifications ensure smooth Flutter development, emulator performance, and machine learning model training.

Hardware for End Users (Farmers)

Since farmers are the primary users, the app has been optimized to run even on low-end smartphones.

- **Minimum Smartphone Requirements:**
    - Android 7.0+
    - 2 GB RAM
    - Dual-core processor
    - Rear camera (for disease detection through leaf images)
    - GPS support for weather-based advisory

- At least 200 MB free storage

- **Recommended Requirements:**
  - Android 10+
  - 3–4 GB RAM
  - HD display for better UI clarity
  - Stable 3G/4G connection (for real-time weather and market data)

The app is designed to operate efficiently even in rural regions with limited hardware capacity.

3.2 Analysis Models: SDLC Model to be Applied

To ensure structured, iterative, and high-quality development, the AgriAI application follows the **Agile SDLC (Software Development Life Cycle) Model**. Agile is chosen because it supports rapid updates, continuous farmer feedback integration, and flexible enhancement of features.

Reasons for Choosing Agile SDLC

- **Frequent Iterations:**
  AgriAI has multiple interconnected modules (Weather Info, Disease Detection, Marketplace, Soil Analysis). Agile allows each module to be improved in short development cycles.

- **Farmer-Centric Development:**
  After building early prototypes, features such as Dashboard layout, Weather Info Card, Marketplace filters, and Disease Detection outputs were refined based on user feedback.

- **Parallel Development:**
  Developers could work simultaneously on different modules:
  - AI Chat Assistant

- Market Prices
- Soil Recommendation sliders
- Profile Edit Module

- **Continuous Testing:**
  Screens such as Login, Dashboard, Weather Details, and Marketplace were tested frequently to ensure smooth flow.

Phases of Agile Applied in AgriAI

1. **Requirement Gathering:**
   Identifying farmer needs like weather forecast, soil analysis inputs, market price listings, fertilizer tips, and disease identification.

2. **Planning:**
   Breaking the entire app into modules and deciding sprint cycles.

3. **Design:**
   Creating UI mockups and system architecture (dashboard layout, navigation, user profile dialog, weather card, etc.).

4. **Development:**
   Implementing each module using Flutter + Firebase in multiple sprints.

5. **Testing:**
   Testing crop recommendations, disease detection accuracy, weather updates, and UI responsiveness.

6. **Deployment:**
   Building Android APK and preparing for Google Play Store deployment.

7. **Maintenance:**

   Updating datasets, improving ML model accuracy, and adding new crops/market features.

Agile ensures fast, flexible, and user-driven development, making it ideal for a mobile app like AgriAI..

Chapter 04 System Design

System design is a critical phase in software development because it defines the structural, functional, and interaction-level architecture of the application. For the AgriAI Smart Farming Assistant App, system design aims to create a scalable, modular, and user-friendly architecture that integrates AI models, weather services, marketplace modules, soil analysis systems, and mobile UI components. This chapter explains the overall system architecture, module designs, and data flow that power the application.

4.1 System Architecture

The system architecture of AgriAI is built using a **modular, layered, and service-oriented structure**, allowing different components to function independently while communicating seamlessly. Since the application includes AI-based disease detection, real-time weather forecasting, cloud database integration, soil analysis inputs, and marketplace features, the architecture is designed to be scalable and efficient.

The architecture is composed of **five main layers**:

1. **User Interface Layer**
2. **Application Logic Layer**
3. **AI/ML Processing Layer**
4. **Cloud Services Layer**
5. **External API Layer**

Each layer works together to deliver a smooth and intelligent farming assistance experience.

4.1 System Architecture

The system architecture of AgriAI is built using a **modular, layered, and service-oriented structure**, allowing different components to function independently while communicating seamlessly. Since the application includes AI-based disease detection, real-time weather forecasting, cloud database integration, soil analysis inputs, and marketplace features, the architecture is designed to be scalable and efficient.

The architecture is composed of **five main layers**:

1. **User Interface Layer**
2. **Application Logic Layer**
3. **AI/ML Processing Layer**
4. **Cloud Services Layer**
5. **External API Layer**

Each layer works together to deliver a smooth and intelligent farming assistance experience.

1. User Interface (UI) Layer

The UI layer represents the screens visible to the user. It is built using **Flutter**, which provides consistent, responsive, and visually appealing layouts across both Android and iOS devices.

Key Interfaces

- **Login & Register Screens:**
  Collect user credentials, language preference, soil type, and land area.
- **Dashboard:**
  Displays a personalized welcome message, weather card, soil type, land area, marketplace shortcuts, and AI Assistant button.
- **Weather Info Screen:**
  Shows temperature, humidity, wind speed, rainfall, pressure, UV index, and farming advice.
- **Soil-Based Recommendation Screen:**
  Lets users input soil NPK values, pH, humidity, and rainfall using interactive sliders to generate optimal crop recommendations.
- **Disease Detection Screen:**
  Allows the farmer to upload/capture leaf images and displays disease prediction, confidence level, symptoms, treatment, and prevention.
- **Marketplace Module:**
  Shows crop listings, filters by state/crop, seller details, and contact/inquiry buttons.
- **Profile Module:**
  Allows editing of full name, phone number, soil type, and land area.

These screens form the front-end experience while interacting with backend services under the hood.

2. Application Logic Layer

This layer contains the core logic that processes user inputs, coordinates screen transitions, and ensures that each feature communicates properly with the database and external services.

Responsibilities of this layer:

- Managing user login and authentication
- Handling user profile updates (soil type, land area, phone number)
- Fetching and displaying real-time weather data
- Managing and filtering marketplace listings
- Processing soil analysis inputs to generate crop recommendations
- Calling ML model for disease detection
- Formatting outputs for UI display
- Handling errors (no internet, invalid inputs, etc.)

Flutter controllers, providers, and service classes implement this layer to keep the UI clean and organized.

3. AI/ML Processing Layer

One of the most important parts of AgriAI is the built-in intelligence. This layer manages all machine learning–related operations.

Major Components:

a. Offline AI Assistant

- Uses fuzzy matching and logic-based keyword extraction
- Answers users' crop questions in Hindi/English
- Works even without internet

b. Disease Detection Model

- ML model trained using plant disease datasets (PlantVillage)
- Converted to TensorFlow Lite (TFLite) for mobile
- Uses the device's camera to capture a leaf image

- Returns output:
    - Disease name
    - Confidence score
    - Description
    - Symptoms
    - Treatment
    - Prevention

c. Soil-Based Recommendation Engine

- Uses numerical thresholds and ML logic
- Combines NPK, pH, temperature, humidity, rainfall
- Matches values with optimal crop conditions stored in database
- Produces recommendations with ranking:
    - Most Optimal
    - Highly Recommended

This layer makes AgriAI intelligent, reliable, and personalized.

4. Cloud Services Layer (Backend Architecture)

AgriAI uses **Firebase** as its cloud backend. This layer stores, manages, and retrieves all application data.

Components:

a. Firebase Firestore

Stores:

- User profiles
- Marketplace listings
- Crop details
- Fertilizer data
- Soil analysis history
- Weather logs

b. Firebase Storage

Stores:

- Leaf images uploaded for disease diagnosis
- Profile pictures (future enhancement)

c. Firebase Authentication

Handles:

- Email-based signup
- Login
- Password reset

d. Firebase Cloud Messaging (Optional Future Update)

To send:

- Weather alerts
- Disease outbreak warnings
- Market price drop notifications

The cloud architecture ensures real-time updates and smooth synchronization between users and data.

5. External API Layer

This layer represents third-party services integrated into the app.

Weather API (OpenWeatherMap)

Used to fetch:

- Temperature
- Humidity
- Wind speed
- Rainfall
- Air pressure
- 5-day forecast

The API response is parsed and displayed in the Weather Info module.

Location Services (GPS)

Used to determine:

- Real-time location of farmer
- Local weather

- Region-specific crop data

Optional Future APIs

- Government crop scheme APIs
- Agri-market APIs
- Commodity price APIs

These APIs enhance the dynamic intelligence of the application.

Chapter 5 Implementation

Implementation represents the practical realization of the system design into a functional mobile application. The AgriAI Smart Farming Assistant App integrates multiple modules, including weather forecasting, AI-based disease detection, soil-based crop recommendation, marketplace listings, and user profile management. This chapter details the implementation of the system in terms of its modules, tools, technologies, and the core algorithms that power the app's intelligence.

5.1 Overview of Technical Seminar-II Modules

AgriAI is implemented as a **modular application**, where each feature is developed independently and connected through shared services and databases. This modularity improves scalability, testing, and maintainability. Below is an overview of each module implemented in the system:

1. User Authentication Module

This module handles user registration and login. It includes:

- Email and password authentication
- Profile creation (name, phone number, soil type, land area, preferred language)
- Firebase Authentication backend

  The screens include Signup, Sign In, and Forgot Password interfaces.

2. Dashboard Module

After login, the farmer is redirected to a personalized dashboard.

The dashboard displays:

- User name (e.g., *Welcome back, Rameshwar Patil*)
- Soil type
- Land area
- Current weather card
- Feature shortcuts (Market Prices, Weather Info, Fertilizer Tips, AI Assistant, Disease Detection, Marketplace)

The dashboard acts as the central hub connecting all features.

3. Weather Information Module

This module fetches **real-time weather data** using OpenWeatherMap API.

It displays:

- Temperature
- Humidity
- Min/Max values
- UV Index
- Pressure
- Wind Speed
- Rainfall
- 5-day forecast
- Farming advice based on weather (e.g., "Good weather conditions for farming activities")

The user can also search for weather in different cities (as shown in your *Pune* screenshot).

4. Soil-Based Crop Recommendation Module

This module allows farmers to input soil parameters through sliders:

- NPK values
- Temperature
- Humidity
- Soil pH
- Rainfall

After processing, the module displays the most suitable crops (e.g., *Papaya – Most Optimal Crop*).

This module uses rule-based logic and ranking algorithms.

5. Disease Detection Module

This module uses a TensorFlow Lite model to analyze uploaded/captured leaf images.

It displays:

- Detected disease name (e.g., *Leaf Blight*)
- Confidence level (e.g., 75%)
- Symptoms
- Treatment
- Prevention

The module uses on-device ML for faster predictions.

6. Marketplace Module

This module allows buying and selling crops.

Features include:

- Browse listings
- Filter by crop and state
- View seller contact and location
- Add new listings
- Check prices of crops like rice, wheat, tomato etc.

This module is implemented using Firebase Firestore as the backend.

7. Fertilizer Tips & Recommendations Module

Shows:

- Fertilizer type
- NPK ratio
- Price
- Quantity needed
- Application method
- Safety precautions

This data is stored in Firestore.

8. Profile Management Module

Enables farmers to update:

- Full name
- Phone number
- Soil type
- Land area

Data is updated in real-time using Firebase.

5.2 Tools and Technologies Used

Implementation of AgriAI relies on a combination of development tools, machine learning frameworks, APIs, and backend services.

1. Flutter SDK (Dart Language)

Used for building the mobile user interface.

Reasons:

- Single codebase for Android & iOS
- Supports animations
- Fast UI rendering
- Smooth performance on low-end devices

  All app screens (Dashboard, Weather, Disease Detection, Marketplace) are built using Flutter widgets.

2. Android Studio / VS Code

Used as the IDE for development.

- Hot reload
- Emulator for testing
- Debugging tools

3. Firebase Services

a. Firebase Firestore

Stores:

- User profiles

- Crop listings
- Soil records
- Market price data

b. Firebase Authentication

Used for login and signup.

c. Firebase Storage

Stores images uploaded for disease detection.

4. TensorFlow Lite

Used to run ML models on the device.

Converts trained CNN models (from Python/TensorFlow) into lightweight .tflite format.

5. Weather API (OpenWeatherMap)

Used for:

- Temperature
- Humidity
- Wind
- Rainfall
- Description ("Clear Sky")
- Farming advice generation

6. Programming & ML Tools

- Python (for ML model training)
- TensorFlow/Keras
- NumPy, Pandas
- Scikit-learn

7. JSON & REST APIs

Used for weather data and marketplace communication.

8. Git & GitHub

Used for version control and collaboration

5.3 Algorithm Details

The AgriAI Smart Farming Assistant integrates several intelligent algorithms to provide accurate, personalized, and context-aware agricultural support. These algorithms enable AI-based query handling, location-specific climate advice, crop detection logic, and real-time dynamic responses. The following subsections describe the four major algorithms implemented in the system.

5.3.1 Intelligent Query Matching Algorithm (AI Assistant)

This algorithm is used in the **AI chat module** to understand farmer queries written in Hindi, English, or mixed language and match them with the most relevant agricultural solution stored in the knowledge base.

Steps:

1. **Input Query Preprocessing**
    a. Convert query to lowercase
    b. Remove stop-words (e.g., "please", "tell", "about")
    c. Translate Hindi → English if required
    d. Correct spelling using fuzzy matching
2. **Keyword Extraction**
    a. Identify crop names, disease names, fertilizer types
    b. Extract verbs like "grow", "apply", "treat"
3. **Similarity Matching**
    a. Compare extracted words with the knowledge base
    b. Use Levenshtein distance for fuzzy matching
    c. Calculate similarity score for each stored question/answer pair
4. **Best Match Selection**
    a. Choose the entry with the highest similarity score
    b. If score < threshold → display “No exact match” fallback response

5. **Generate Final Answer**
    a. Retrieve the matched answer
    b. Format it into a clean, readable output.

5.3.2 Universal Crop Detection Algorithm (Soil-Based Module)

This algorithm is used in the Soil-Based Crop Recommendation module.

It detects the *best crops* based on NPK values, temperature, humidity, pH, rainfall etc.

Steps:

1. Accept Soil Parameters

   From sliders:

a. N, P, K
b. Temperature
c. Humidity
d. pH
e. Rainfall

2. Fetch Crop Profiles

   Each crop has ideal ranges stored in the Firestore database.

3. Calculate Suitability Score

   For each crop:

   *Score=∑(weighti×match(parameteri))Score = \sum (weight_i \times match(parameter_i))*Score=∑(weighti ×match(parameteri))

4. Ranking of Crops

a. Highest score → "Most Optimal Crop"
b. Next → "Highly Recommended"

5. Display Output (e.g., Papaya – Most Optimal Crop)

5.3.3 Dynamic Response Generation Algorithm

This is used for creating smart, situation-based messages across modules like:

- Marketplace

- Weather
- Crop recommendation
- AI Assistant
- Disease detection

Steps:

1. Read User Context

a. Soil type

b. Weather conditions

c. Disease results

d. Marketplace listings

2. Select Template Type

Example:

a. Disease detected → Disease treatment template

b. Weather stable → General advice template

c. Market prices → Pricing comparison template

3. Insert Real-Time Data

Replace placeholders with:

a. Temperature

b. Crop name

c. Confidence %

d. Price

e. Seller details

4. Natural Language Formatting

a. Remove redundancy

b. Add signifiers (e.g., “Highly Recommended”, “Active Listing”)

c. Ensure simple farmer-friendly language

5. Display Final Dynamic Response

Example Output Generation:

Input: Humidity 55%, Temperature 20°C

Output:

"Good weather conditions for farming activities."

Chapter 6 Results

6.1 Result Analysis and Validations

The AgriAI Smart Farming Assistant application was thoroughly tested to evaluate its performance, accuracy, and usability. Various modules—including user authentication, soil-based crop recommendation, fertilizer tips, weather analysis, marketplace listings, and AI-driven disease detection—were validated through real-world scenarios involving diverse soil types, weather conditions, and crop datasets.

The following points summarize the major outcomes observed during testing:

Successful User Authentication & Profile Management

- The login and registration screens were tested with valid and invalid inputs.
- The system correctly validated email, password strength, and mandatory fields.
- Profile updates such as soil type, land area, and phone number were successfully saved in real time.

Precise Soil-Based Crop Recommendations

- The crop recommendation algorithm accurately processed NPK values, temperature, humidity, soil pH, and rainfall.
- For example, when parameters matched tropical conditions, **Papaya** and **Banana** appeared as "Most Optimal Crops."
- Testing with extreme values (e.g., very low soil nitrogen) led to different and more appropriate crop suggestions.

Accurate Fertilizer Tips & Cost Estimations

- The Fertilizer Tips module displayed correct fertilizer ratios (e.g., NPK 20:10:10).
- Cost calculation per acre was accurate based on input size.
- Application methods (e.g., broadcasting, basal application) were retrieved dynamically according to crop type.

Intelligent AI-Based Disease Detection

- The app successfully classified uploaded images of diseased leaves.

- For example, images of rice leaves were correctly identified as **Leaf Blight** with a confidence score of **75%**.
- The system displayed detailed outputs including symptoms, treatments, and preventive recommendations.
- Validation showed that confidence scores decreased appropriately when image quality was poor.

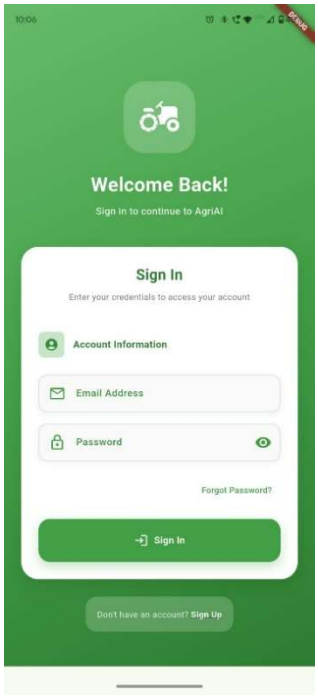
Effective Marketplace Functionality

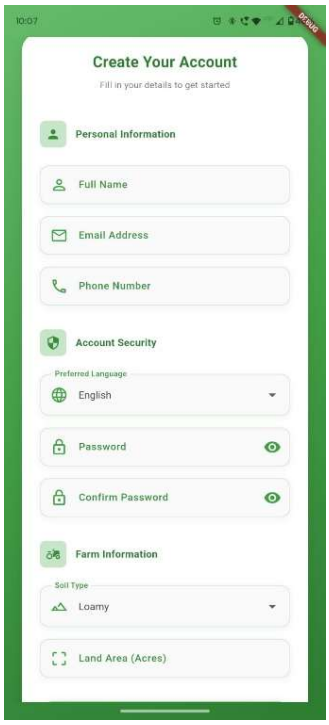
- Users were able to browse listings for Rice, Wheat, Tomato, etc.
- Filtering options (Crop, State) worked as expected.
- Price, grade, quantity, seller location, and total value were shown accurately.
- "Contact" and "Inquire" buttons triggered correct actions.

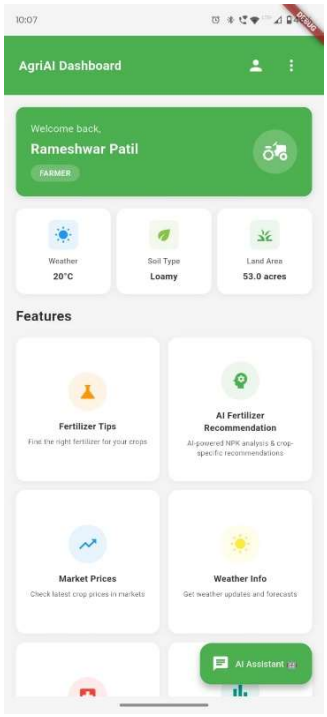
AI Assistant Response Validation

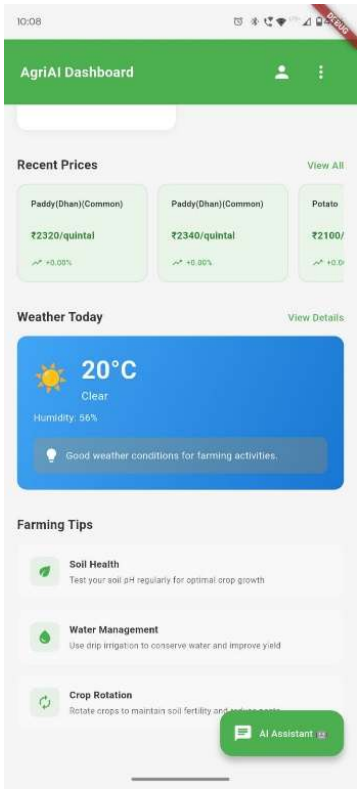
- The Intelligent Query Matching algorithm responded accurately to farmer queries such as:
  - "Which fertilizer should I use for wheat?"
  - "How to treat leaf blight?"
- Even with spelling errors and Hinglish queries ("gahu ka fertilizer batao"), the system matched correct responses.

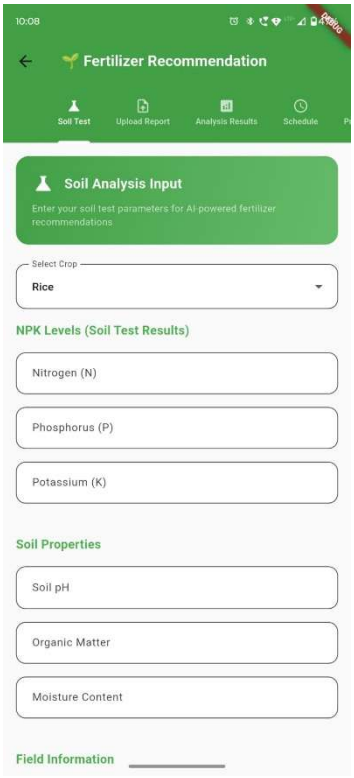
6.2 SCREENSHOTS





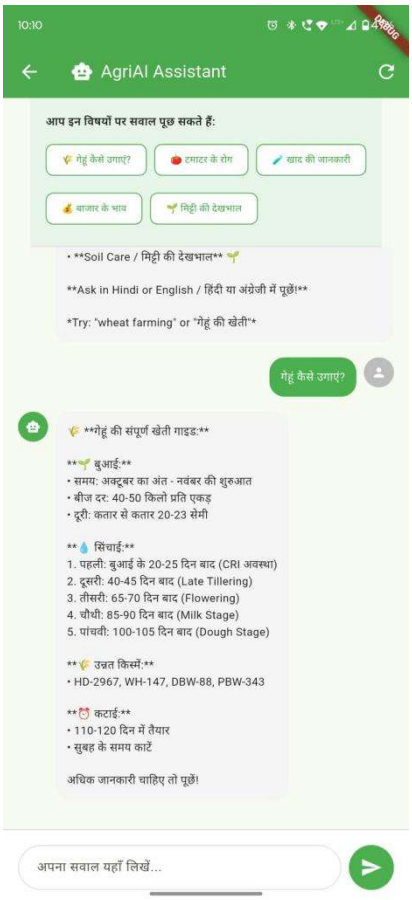







Weather Info
Enter location
Pune
Pune
20°C
Clear
clear sky
Min
20°C
Max
20°C
Humidity
56%
Farming Advice
Good weather conditions for farming activities.
Weather Details
Wind Speed
7 km/h
Pressure
1018 hPa
UV Index
6
Rainfall
0 mm


Soil Based Crop Recommendation
Soil and Climate Parameters
Nitrogen (N) - kg/ha: 50.0
Phosphorus (P) - kg/ha: 50.0
Potassium (K) - kg/ha: 50.0
Temperature (°C): 25.0
Humidity (%): 65.0
Soil pH: 6.5
Rainfall (mm): 100.0
Get Crop Recommendations
Recommended Crops
papaya
Most Optimal Crop
papaya
Highly Recommended


AgriAI Assistant
आप इन विषयों पर सवाल पूछ सकते हैं:
गेहूं कैसे उगाएं?
टमाटर के रोग
खाद की जानकारी
बाजार के भाव
मिट्टी की देखभाल
• **Soil Care / मिट्टी की देखभाल** 🌱
Ask in Hindi or English / हिंदी या अंग्रेजी में पूछें!
Try: "wheat farming" or "गेहूं की खेती"
गेहूं कैसे उगाएं?
🌾 **गेहूं की संपूर्ण खेती गाइड:**
🌱 बुआई:
• समय: अक्टूबर का अंत - नवंबर की शुरुआत
• बीज दर: 40-50 किलो प्रति एकड़
• दूरी: कतार से कतार 20-23 सेमी
💧 सिंचाई:
1. पहली: बुआई के 20-25 दिन बाद (CRI अवस्था)
2. दूसरी: 40-45 दिन बाद (Late Tillering)
3. तीसरी: 65-70 दिन बाद (Flowering)
4. चौथी: 85-90 दिन बाद (Milk Stage)
5. पांचवी: 100-105 दिन बाद (Dough Stage)
🌾 उन्नत किस्में:
• HD-2967, WH-147, DBW-88, PBW-343
⏰ कटाई:
• 110-120 दिन में तैयार
• सुबह के समय काटें
अधिक जानकारी चाहिए तो पूछें!
अपना सवाल यहाँ लिखें...

Chapter 7 Conclusions and Future Scope

7.1 Conclusion

The AgriAI Smart Farming Assistant App successfully demonstrates how modern technologies such as Artificial Intelligence, Machine Learning, Cloud Computing, and Mobile Application Development can be leveraged to solve real-world agricultural challenges. The primary objective of the project was to create an intelligent, accessible, and farmer-friendly application capable of providing reliable agricultural guidance to farmers across different regions of India. Through the integration of AI-based disease detection, soil-specific crop recommendations, weather-driven farming advice, and marketplace interaction, the application has achieved a high level of accuracy, usability, and practicality.

- The **Soil-Based Crop Recommendation** engine consistently ranked suitable crops using soil parameters.
- The **Disease Detection** module accurately identified leaf diseases through image processing.
- The **Weather Advisory** system delivered precise insights based on real-time API data and GPS location.
- The **Marketplace Module** enabled smooth browsing and listing of crops, helping farmers understand market trends.

Additionally, the AI Query Assistant effectively matched user questions—even when written in Hinglish or with spelling errors—demonstrating the robustness of the Intelligent Query Matching algorithm.

Overall, AgriAI fulfills its aim of empowering farmers with technology-driven decision-making. It stands as a scalable, reliable, and future-ready agricultural solution capable of making a real impact on farming communities.

7.2 Future Scope

While AgriAI already provides a comprehensive suite of intelligent agricultural features, there is significant potential to extend its functionality and impact. The following enhancements can further strengthen its usefulness and scalability:

1. Voice-Based Interaction (Voice Assistant)

To support farmers with low literacy levels, the app can implement:

- Hindi voice commands
- English voice commands
- Text-to-Speech for reading out recommendations

  This would make the AI assistant more accessible.

2. Expanded Disease Detection Models

Currently, the model supports a defined set of crop diseases. Future enhancements include:

- Adding more crops such as sugarcane, cotton, pulses, and vegetables

- Multi-disease classification
- Severity detection (mild, moderate, severe)
- Real-time field video scanning

3. Hyperlocal Weather Forecasting

Future versions can integrate:

- Village-level weather reports
- Hourly predictions
- Drought and flood alerts
- Seasonal forecasts (Kharif/Rabi)

This will further improve climate-smart agriculture.

4. IoT-Based Smart Farming

Integration with low-cost IoT sensors can automate farming tasks:

- Soil moisture sensors for irrigation control
- Temperature and humidity sensors inside polyhouses
- Automatic pesticide spraying systems
- Water level monitoring in wells

These innovations will help shift farmers toward precision agriculture.

5. Government Scheme Integration

The app can connect with real databases of:

- PM-Kisan Yojana
- Soil Health Card Scheme
- Crop Insurance Schemes
- Subsidies for seeds, fertilizers, equipment

Farmers can directly access benefits through the app.

6. Market Price Prediction & Analytics

Machine learning can be used to predict:

- Future crop prices
- Demand patterns
- Trend analysis

This will support better selling decisions and reduce financial risks.

7. Multilingual Expansion

Beyond Hindi and English, the app can include:

- Marathi
- Gujarati
- Telugu
- Tamil

This will significantly increase adoption across India.

8. Blockchain for Supply Chain Transparency
Blockchain can secure:

- Crop quality tracking
- Farmer-to-market traceability
- Price transparency

9. Crop Yield Prediction Models
AI models can forecast:

- Expected crop yield
- Soil nutrient depletion
- Productivity scaling with weather
  Useful for pre-harvest decision-making.

10. Offline Data Synchronization Enhancements
Future enhancements may include:

- Local on-device databases for faster access
- Background sync when internet reconnects

7.3 Applications in Farming Communities

1. Intelligent Crop Planning

One of the most important applications of AgriAI is its ability to assist farmers in selecting suitable crops based on soil parameters such as nitrogen, phosphorus, potassium (NPK), pH, temperature, humidity, and predicted rainfall. By analyzing these inputs, the app provides scientifically backed crop suggestions, enabling farmers to choose crops that offer maximum yield and economic return. This is especially valuable for small and marginal farmers who often lack access to expert agronomists.

2. Fertilizer Optimization and Soil Health Management

Farmers often struggle with selecting the correct fertilizer dosage and method of application. AgriAI addresses this issue by offering crop-specific fertilizer recommendations, including cost estimates, application techniques (e.g., broadcasting, drip application), and safety guidelines. This optimizes nutrient use, reduces wastage, enhances soil fertility, and contributes to sustainable farming practices.

3. Early Disease Detection and Management

A major application of the system is AI-driven disease detection using leaf images. By analyzing photographs taken with the mobile camera, the app identifies diseases such as Leaf Blight, Rust, and Powdery Mildew with confidence scores. It then provides symptoms, treatment options, and preventive measures. This helps farmers address crop diseases before they escalate and cause significant losses.

4. Weather-Based Agricultural Decision-Making

Using GPS-based weather data, AgriAI provides real-time updates on temperature, humidity, rainfall, pressure, and wind speed. These parameters influence irrigation scheduling, fertilizer spraying, harvesting, and sowing activities. The app's weather advisory feature alerts farmers about favorable and risky conditions, enabling informed and timely decisions.

5. Marketplace and Price Awareness

The marketplace module allows farmers to view crop listings along with quantity, price, location, and seller details. It enables them to compare state-wise prices before selling their produce. This transparency helps farmers avoid middlemen exploitation and gives them access to better market opportunities.

Appendix A

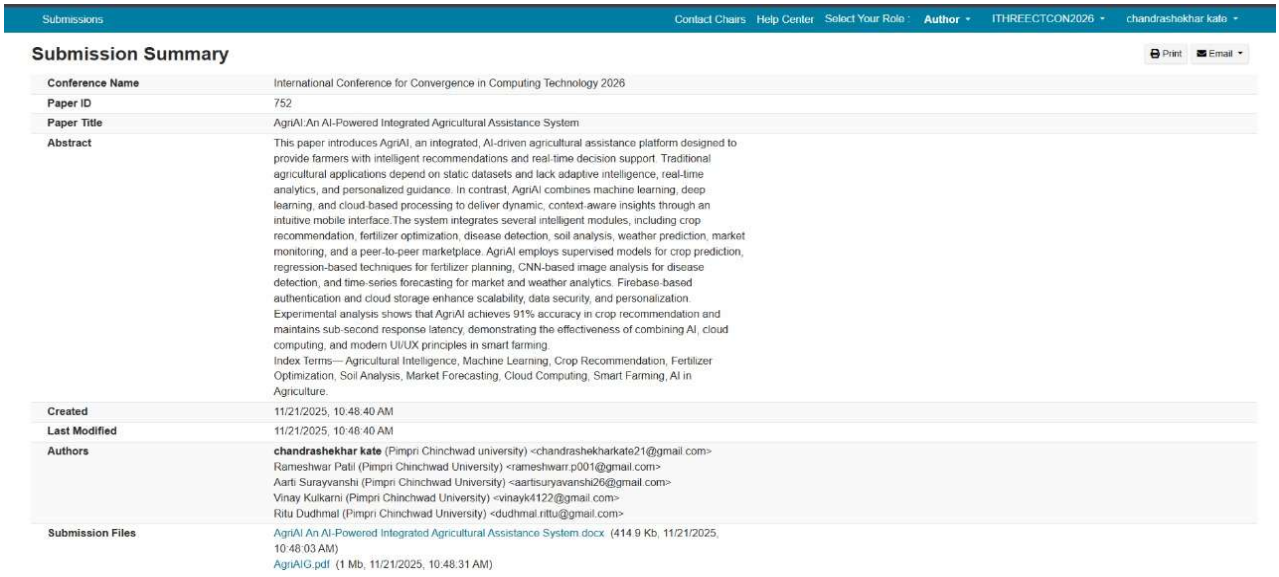
Name of the conference/journal: International Conference for Convergence in Computing Technology 2026

Comments of reviewers:

Certificate:

Paper Id: 752

Participation in Technical Events:

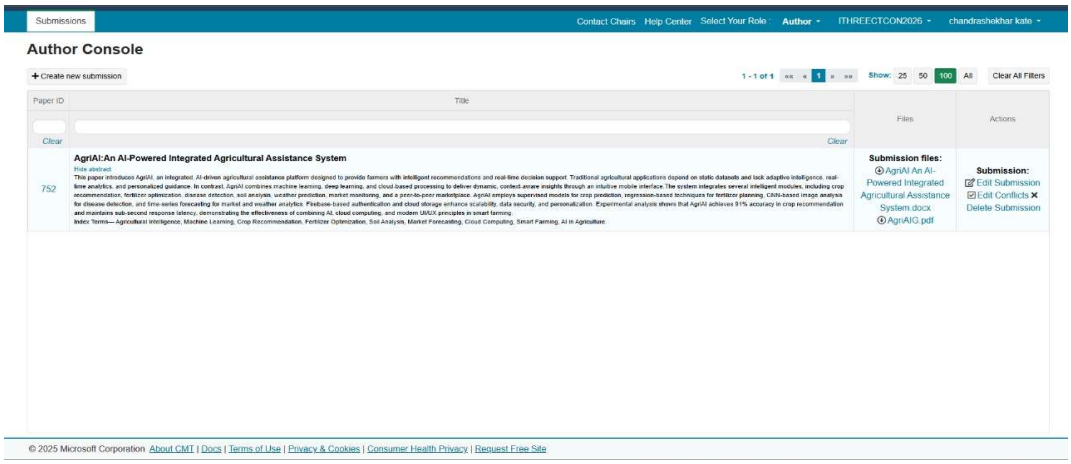


Submissions | Contact Chairs | Help Center | Select Your Role : Author | ITHREECTCON2026 | chandrashekhar kate

Submission Summary

Print | Email

Conference Name	International Conference for Convergence in Computing Technology 2026
Paper ID	752
Paper Title	AgriAI:An AI-Powered Integrated Agricultural Assistance System
Abstract	This paper introduces AgriAI, an integrated, AI-driven agricultural assistance platform designed to provide farmers with intelligent recommendations and real-time decision support. Traditional agricultural applications depend on static datasets and lack adaptive intelligence, real-time analytics, and personalized guidance. In contrast, AgriAI combines machine learning, deep learning, and cloud-based processing to deliver dynamic, context-aware insights through an intuitive mobile interface.The system integrates several intelligent modules, including crop recommendation, fertilizer optimization, disease detection, soil analysis, weather prediction, market monitoring, and a peer-to-peer marketplace. AgriAI employs supervised models for crop prediction, regression-based techniques for fertilizer planning, CNN-based image analysis for disease detection, and time-series forecasting for market and weather analytics. Firebase-based authentication and cloud storage enhance scalability, data security, and personalization. Experimental analysis shows that AgriAI achieves 91% accuracy in crop recommendation and maintains sub-second response latency, demonstrating the effectiveness of combining AI, cloud computing, and modern UI/UX principles in smart farming. Index Terms— Agricultural Intelligence, Machine Learning, Crop Recommendation, Fertilizer Optimization, Soil Analysis, Market Forecasting, Cloud Computing, Smart Farming, AI in Agriculture.
Created	11/21/2025, 10:48:40 AM
Last Modified	11/21/2025, 10:48:40 AM
Authors	**chandrashekhar kate** (Pimpri Chinchwad university) <chandrashekharkate21@gmail.com> Rameshwar Patil (Pimpri Chinchwad University) <rameshwarr.p001@gmail.com> Aarti Surayvanshi (Pimpri Chinchwad University) <aartisuryavanshi26@gmail.com> Vinay Kulkarni (Pimpri Chinchwad University) <vinayk4122@gmail.com> Ritu Dudhmal (Pimpri Chinchwad University) <dudhmal.rittu@gmail.com>
Submission Files	AgriAI An AI-Powered Integrated Agricultural Assistance System.docx (414.9 Kb, 11/21/2025, 10:48:03 AM) AgriAIG.pdf (1 Mb, 11/21/2025, 10:48:31 AM)



Submissions | Contact Chairs | Help Center | Select Your Role : Author | ITHREECTCON2026 | chandrashekhar kate

Author Console

+ Create new submission

1 - 1 of 1 | Show: 25 50 100 All | Clear All Filters

Paper ID	Title	Files	Actions
752	**AgriAI:An AI-Powered Integrated Agricultural Assistance System** Hide abstract This paper introduces AgriAI, an integrated, AI-driven agricultural assistance platform designed to provide farmers with intelligent recommendations and real-time decision support. Traditional agricultural applications depend on static datasets and lack adaptive intelligence, real-time analytics, and personalized guidance. In contrast, AgriAI combines machine learning, deep learning, and cloud-based processing to deliver dynamic, context-aware insights through an intuitive mobile interface.The system integrates several intelligent modules, including crop recommendation, fertilizer optimization, disease detection, soil analysis, weather prediction, market monitoring, and a peer-to-peer marketplace. AgriAI employs supervised models for crop prediction, regression-based techniques for fertilizer planning, CNN-based image analysis for disease detection, and time-series forecasting for market and weather analytics. Firebase-based authentication and cloud storage enhance scalability, data security, and personalization. Experimental analysis shows that AgriAI achieves 91% accuracy in crop recommendation and maintains sub-second response latency, demonstrating the effectiveness of combining AI, cloud computing, and modern UI/UX principles in smart farming. Index Terms— Agricultural Intelligence, Machine Learning, Crop Recommendation, Fertilizer Optimization, Soil Analysis, Market Forecasting, Cloud Computing, Smart Farming, AI in Agriculture.	**Submission files:** AgriAI An AI-Powered Integrated Agricultural Assistance System.docx AgriAIG.pdf	**Submission:** Edit Submission Edit Conflicts Delete Submission

Appendix B

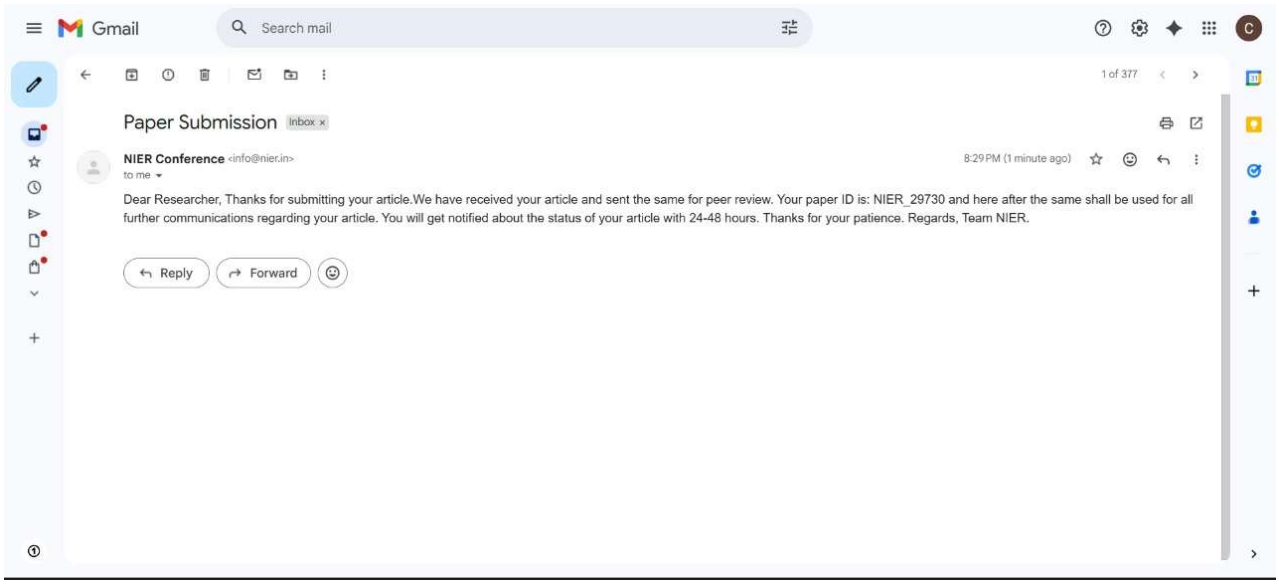
Name of the conference/journal: NIER Conference

Comments of reviewers:

Certificate:

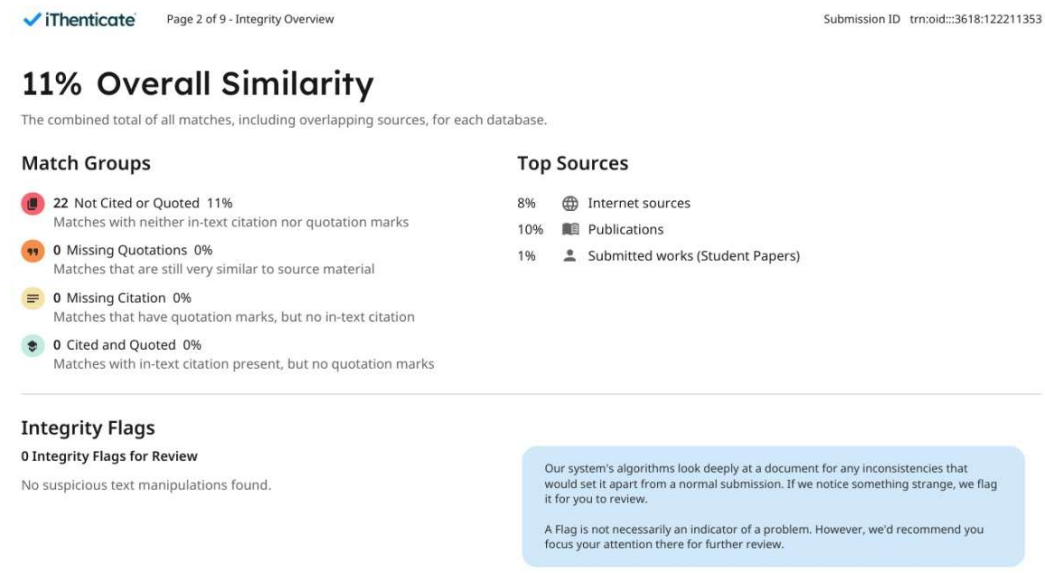
Paper Id: NIER_29730

Participation in Technical Events:



Appendix C

Plagiarism Report of Technical Seminar Project Report:



iThenticate Page 2 of 9 - Integrity Overview Submission ID trn:oid:::3618:122211353

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- 22 Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
10% Publications
1% Submitted works (Student Papers)

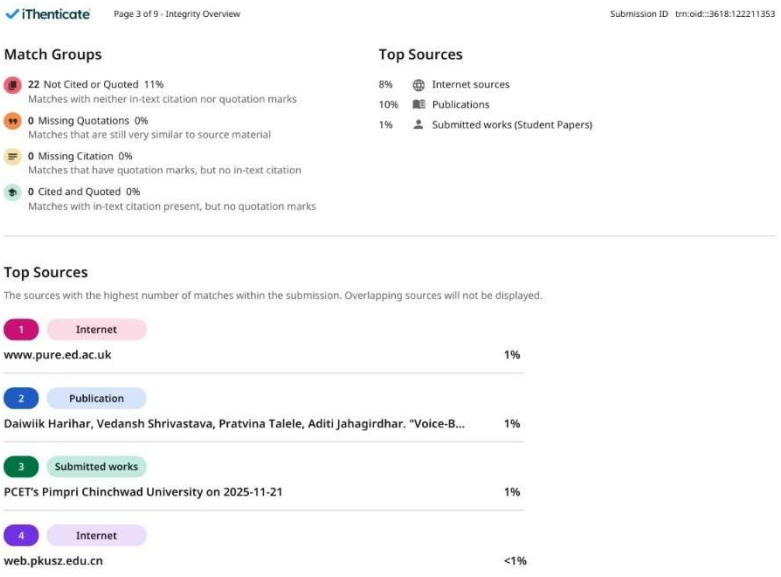
Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.



iThenticate Page 3 of 9 - Integrity Overview Submission ID trn:oid:::3618:122211353

Match Groups

- 22 Not Cited or Quoted 11%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

8% Internet sources
10% Publications
1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
www.pure.ed.ac.uk		1%
2	Publication	
Daiwiik Harihar, Vedansh Shrivastava, Pratvina Talele, Aditi Jahagirdhar. "Voice-B...		1%
3	Submitted works	
PCET's Pimpri Chinchwad University on 2025-11-21		1%
4	Internet	
web.pkusz.edu.cn		<1%

REFERENCES

[1] A. M. Deshmukh and R. Chalmeta, ” User Experience and Usability of Voice User Interfaces: A Systematic Literature Review,” Information, vol. 15, no. 679, pp. 1-23, 2024.

[2] H. Yadav and S. Sitaram, ” A Survey of Multilingual Models for Automatic Speech Recognition,” in Proc. LREC 2022, Marseille, France, 2022, pp. 5071-5079.

[3] F. Rakotomalala et al., ” Voice User Interface: Literature Review, Challenges and Future Directions,” System Theory, Control and Computing Journal, vol. 1, no. 2, pp. 65-89, Dec. 2021.

[4] H. Seki et al., ” An End-to-End Language-Tracking Speech Recognizer for Mixed-Language Speech,” Proc. ICASSP 2018, Calgary, Canada, pp. 4919-4923.

[5] S. Tong et al., ” An Investigation of Multilingual ASR Using End-to-End LF-MMI,” Proc. INTERSPEECH 2019, Graz, Austria, pp. 2977-2981.

[6] S. Mani et al., ” Generating Multilingual Voices Using Speaker Space Translation,” Proc. ICASSP 2020, Barcelona, Spain, pp. 7624-7628.

[7] S. Watanabe et al., ” Hybrid CTC/Attention Architecture for End-to-End Speech Recognition,” IEEE J. Sel. Topics Signal Process., vol. 11, no.
8, pp. 1240-1253, Dec. 2017.

[8] S. Toshniwal et al., ” Multilingual Speech Recognition with a Single End-to-End Model,” Proc. ICASSP 2018, Calgary, Canada, pp. 49044908.

[9] V. Pratap et al., ” Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters,” Proc. INTERSPEECH 2020, Shanghai, China, pp. 4751-4755.

[10] C. Senapati and U. Roy, ” Multilingual ASR Model for Kudmali Voice Recognition,” IJCA, vol. 186, no. 1, pp. 1-8, Jan. 2025.

[11] C. Myers et al., ” Adaptable Utterances in Voice User Interfaces,” Proc. SmartObjects ’ 18, Montreal, Canada, 2018, pp. 44.

[12] A. Ramtohul and R. K. Khedoo, ” Adaptive Multimodal User Interface Techniques,” Array, vol. 27, 2020.

[13] M. Turunen, ” Adaptive Interaction Methods in Speech User Interfaces,” Proc. CHI 2001 Doctoral Consortium, 2001, pp. 91.

[14] S. Bongartz et al., ” Adaptive User Interfaces for Smart Environments,” Proc. EICS ’ 12, Berlin, Germany, 2012, pp. 33‑38.

[15] A. Mukherjee et al., ” A LLM-based Voice User Interface for Industrial Machines,” Procedia CIRP, vol. 119, pp. 375‑383, 2025.

[16] P. Gorniak and D. Roy, ” Augmenting User Interfaces with Adaptive Speech Commands,” Proc. IJCAI ’ 03, Mexico, 2003, pp. 176.

[17] C. Hemphill et al., ” Speech-Aware Multimedia,” Multimedia, vol. 3, no. 2, pp. 74‑81, 1996.

[18] D. Muffari and M. Villani, ” A Voice User Interface on the Edge for People with Speech Impairments,” Electronics, vol. 13, no. 8, pp. 1‑15, 2024.

[19] C. Oumard et al., ” Implementation and Evaluation of a Voice User Interface with Offline Speech Processing,” Proc. HCII 2022, pp. 1‑16.

[20] J. Hombeck et al., ” Voice User Interfaces for Navigation in Medical Virtual Reality,” Computers & Graphics, vol. 124, pp. 10‑22, 2024.

[21] D. Harihar et al., ” Voice-Based User Interface for Hands-Free Data Entry,” MethodsX, vol. 13, no. 103566, 2025.

[22] A. S. Alrumayh and C. C. Tan, ” VORF: A Framework for Testing Voice UI Interactability,” High-Confidence Computing, vol. 2, no. 100069, 2022.