VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka - 590018

A Project Report on

AGRO VISION: Agricultural Intelligence System for Predictive Price Analytics

Submitted in partial fulfilment of the requirements for the conferment of degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

Rakshitha L N (1BY22CS142)
Reyyan Aleem Janbaz (1BY22CS146)
Sarika S Sura (1BY22CS162)
Swithin Fernandes (1BY22CS184)

Under the Guidance of

Prof. Tanishq Nanda
Assistant Professor, Department of CSE

Department of Computer Science and Engineering

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Autonomous Institute under VTU, Belagavi, Karnataka - 590 018)

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Autonomous Institute under VTU, Belagavi, Karnataka - 590 018)

Yelahanka, Bengaluru, Karnataka - 560119

CERTIFICATE

This is to certify that the project entitled **"AGRO VISION: Agricultural Intelligence System for Predictive Price Analytics"** is a Bonafide work carried out by **Rakshitha L N (1BY22CS142), Reyyan Aleem Janbaz (1BY22CS146), Sarika S Sura (1BY22CS162)** and **Swithin Fernandes (1BY22CS184)** in partial fulfilment for the award of "BACHELOR OF ENGINEERING" in "Computer Science and Engineering" of the Visvesvaraya Technological University, Belagavi, during the year 2025-26. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect to work for the BE degree.

Signature of the Guide
Prof. Tanishq Nanda
Assistant Professor, Department of CSE
Department of CSE

Signature of the Cluster Head
Dr. Radhika R
Associate Professor & Associate Head
Department of CSE

Signature of the HoD
Dr. Satish Kumar T
Professor and HoD
Department of CSE

Signature of the Principal
Dr.SANJAY H A
Principal
BMSITM, Bengaluru

Name of the Examiners	**Signature with Date**
1.	.....
2.	.....

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to everyone who has contributed to make this project a memorable experience and has inspired this work in some way.

Let me begin by expressing my gratitude to the Almighty God for the numerous blessings bestowed upon me. We are happy to present this project after completing it successfully.

This project would not have been possible without the guidance, assistance, and suggestions of many individuals. We express our deep sense of gratitude and indebtedness to each and every one who has helped us make this project a success.

We heartily thank **Dr. Sanjay H A**, Principal, BMS Institute of Technology & Management for constant encouragement and inspiration in taking up this project.

We heartily thank **Dr. Satish Kumar T**, HoD, Department of Computer Science and Engineering, BMS Institute of Technology & Management for constant encouragement and inspiration in taking up this project.

We heartily thank **Dr. Radhika R**, Cluster Head, BMS Institute of Technology & Management for constant encouragement and inspiration in taking up this project.

We gratefully thank our project guide, **Prof. Tanishq Nanda**, for guidance and support throughout the course of the project work.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation. Lastly, we thank our parents and friends for their encouragement and support in helping us complete this work.

Rakshitha L N (1BY22CS142)
Reyyan Aleem Janbaz (1BY22CS146)
Sarika S Sura (1BY22CS162)
Swithin Fernandes (1BY22CS184)

DECLARATION

We, hereby declare that the project titled **"AGRO VISION: Agricultural Intelligence System for Predictive Price Analytics"** is a record of original project work under the guidance of **Prof. Tanishq Nanda, Assistant Professor, Department of CSE**, Department of Computer Science and Engineering, BMS Institute of Technology & Management, Autonomous Institute under Visvesvaraya Technological University, Belagavi during the Academic Year 2025-26.

I also declare that this project report has not been submitted for the award of any degree, diploma, associateship, fellowship or other title anywhere else.

Name of the Student	USN	Signature
Rakshitha L N	1BY22CS142	
Reyyan Aleem Janbaz	1BY22CS146	
Sarika S Sura	1BY22CS162	
Swithin Fernandes	1BY22CS184	

ABSTRACT

Agro Vision is an agricultural intelligence platform designed to provide real-time crop price insights, historical trend analysis, and AI-driven price predictions for farmers, merchants, and consumers. The agricultural sector in India, despite employing a significant portion of the workforce, continues to face challenges related to information asymmetry and market unpredictability. Farmers often lack access to timely and accurate price information, leading to suboptimal selling decisions, while merchants struggle to identify regional arbitrage opportunities and consumers remain unaware of the factors driving retail price fluctuations.

This project addresses these challenges by developing a comprehensive web-based platform that transforms raw agricultural market data into actionable insights. The system integrates interactive dashboards, region-based filtering, and a lightweight prediction model to simplify decision-making in a highly fluctuating market. The dashboard presents crop prices in a visually intuitive card-based layout inspired by financial trading platforms, enabling quick assessment of market conditions. Detailed analytics pages provide interactive historical charts with selectable time frames, regional comparisons, and trend visualizations.

The AI-powered prediction engine employs a hybrid approach combining linear regression for trend analysis, seasonal adjustment factors, and real-time weather and supply-demand indicators to generate price forecasts with confidence scoring. Users can explore factors influencing price changes, access crop-specific news, and receive guidance through an integrated context-aware chatbot that understands the current page context and provides relevant explanations.

The platform implements a multi-role architecture that tailors information presentation to the specific needs of farmers (selling timing and price optimization), merchants (regional price comparison and procurement planning), and consumers (seasonal availability and retail price understanding). By combining data visualization with intelligent analytics, Agro Vision offers a unified, user-friendly solution that enhances transparency, supports better market decisions, and bridges the information gap in agricultural pricing. The system is built using modern web technologies including React JS, Node.js, and PostgreSQL, ensuring scalability and maintainability for future enhancements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Agriculture remains one of the most essential sectors in developing economies, providing livelihood, food security, and raw materials for industries. However, despite its importance, the agricultural market is largely unorganized and unpredictable. Price variations, climatic uncertainties, and market discrepancies often result in financial losses for farmers and inefficiencies in the supply chain. With the rise of digital technologies, data analytics, and artificial intelligence, there exists an opportunity to build systems that enable informed decision-making for all agricultural stakeholders—farmers, merchants, and consumers.

The project titled "Agricultural Intelligence System for Predictive Price Analytics" aims to bridge this gap by building a unified digital platform—AgroVision—that transforms traditional, intuition-based market participation into data-driven decision-making. The system provides real-time price updates, historical data visualizations, AI-generated predictions, and intelligent assistance through an integrated chatbot. By combining multiple modern technologies, AgroVision seeks to make agriculture more transparent, predictable, and accessible.

1.1 Background

Agricultural markets are influenced by multiple interdependent factors, including seasonal variations, weather patterns, pest outbreaks, transportation costs, government policies, import-export regulations, and global commodity prices. This complexity makes it difficult for stakeholders to accurately estimate the value of a crop at any given time. Traditionally, farmers access market price information through local traders, informal networks, or outdated reports. This lack of reliable, real-time data often leads to poor pricing decisions, forcing farmers to sell below the fair market rate. Merchants face a different challenge—they need to understand price variations across regions to optimize procurement and distribution strategies. Consumers, meanwhile, experience fluctuating retail prices and have limited knowledge of the reasons behind these changes.

1.1.1 The Indian Agricultural Context

India's agricultural sector employs approximately 42% of the country's workforce and contributes around 18% to the national GDP. Despite its significance, the sector continues to face structural challenges that limit the economic potential of farming communities. The agricultural supply chain involves multiple intermediaries between the

farm gate and the consumer, with each layer adding costs and reducing transparency. Price discovery mechanisms in traditional mandis (wholesale markets) often disadvantage farmers who lack access to information about prevailing rates in distant markets.

The volatility of agricultural prices poses significant risks to all stakeholders. Farmers who plant crops based on high prices observed during the previous season may face losses if prices crash by the time of harvest. Similarly, merchants who procure large quantities anticipating price increases may suffer if market conditions change unexpectedly. This uncertainty discourages investment in agricultural infrastructure and technology, perpetuating a cycle of low productivity and income instability.

1.1.2 Information Asymmetry in Agricultural Markets

One of the fundamental problems in agricultural commerce is information asymmetry—the unequal distribution of market knowledge among participants. Large traders and institutional buyers often have access to comprehensive market intelligence, including data on production estimates, weather forecasts, import-export policies, and inventory levels. In contrast, small-scale farmers typically rely on word-of-mouth information or advice from local traders who may have conflicting interests.

This asymmetry manifests in several ways. Farmers may sell their produce immediately after harvest when prices are at their lowest, simply because they lack storage facilities or information about expected price recovery. They may be unaware of better prices available in nearby markets due to the absence of real-time price comparison tools. The inability to forecast prices also limits their capacity to negotiate effectively with buyers or plan their cropping patterns strategically.

1.1.3 The Role of Technology in Agricultural Transformation

With the evolution of cloud computing platforms, mobile applications, and AI-based forecasting models, it has now become possible to centralize large amounts of agricultural data and present them in user-friendly formats. Modern technologies make it feasible to show market behavior in the same way financial platforms visualize stock price trends. The widespread adoption of smartphones, even in rural areas, has created an opportunity to deliver sophisticated market intelligence directly to farmers and other stakeholders.

Artificial intelligence and machine learning techniques have demonstrated significant potential in pattern recognition and predictive analytics. When applied to agricultural price data, these technologies can identify trends, detect anomalies, and generate forecasts that would be impossible through manual analysis. Natural language processing enables the development of conversational interfaces that can explain complex data in simple terms, making technology accessible to users with limited digital literacy.

AgroVision uses this technological shift to bring sophistication and intelligence to

agricultural commerce, democratizing access to market insights that were previously available only to well-resourced market participants.

1.2 Literature Survey

Numerous studies and systems emphasize the role of digitalization in agriculture. Government portals report daily agricultural prices, but they lack real-time updates, predictive analytics, and role-based customization. Research papers highlight the use of machine learning algorithms like ARIMA, LSTM, and regression models for agricultural price forecasting. Studies on precision agriculture explore the use of satellite imagery, weather models, and IoT data for predicting crop health and yields.

1.2.1 Government and Institutional Platforms

The Government of India has launched several digital initiatives to improve agricultural market transparency. The electronic National Agriculture Market (e-NAM) platform, introduced in 2016, aims to create a unified national market for agricultural commodities by networking existing APMC mandis. While e-NAM has improved price discovery to some extent, it primarily focuses on facilitating online trading rather than providing analytical insights or predictive capabilities. The platform lacks interactive visualizations that could help users understand price trends over time.

Agmarknet, maintained by the Directorate of Marketing and Inspection, provides daily wholesale price data from regulated markets across India. However, the interface is largely text-based and does not offer graphical representations of historical trends. Users must manually compare prices across dates and regions, making it difficult to identify patterns or make informed decisions quickly.

Kisan Suvidha, a mobile application developed by the Ministry of Agriculture, provides information on weather forecasts, market prices, plant protection, and expert advisories. While comprehensive in scope, the application treats these features as separate modules without integrating them into a unified decision-support framework. The lack of AI-driven insights limits its utility for users seeking predictive guidance.

1.2.2 Academic Research on Price Prediction

Academic literature on agricultural price forecasting has explored various statistical and machine learning approaches. Time series models such as ARIMA (Autoregressive Integrated Moving Average) have been widely applied due to their effectiveness in capturing temporal dependencies in price data. Research by Karthick and Kannan (2022) demonstrated that ARIMA models could achieve reasonable accuracy for short-term price forecasting of commodities like rice and wheat.

More recent studies have explored deep learning architectures for agricultural price prediction. Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have shown promise in capturing complex nonlinear patterns in price movements. Research indicates that LSTM models outperform traditional statistical methods when sufficient historical data is available and when prices exhibit strong seasonal components.

Hybrid approaches combining multiple techniques have also gained attention. Ensemble methods that integrate predictions from ARIMA, neural networks, and regression models can provide more robust forecasts by leveraging the strengths of each approach. However, most academic implementations remain confined to research environments and have not been translated into user-friendly applications accessible to farmers and traders.

1.2.3 Gaps in Existing Solutions

Existing applications such as e-NAM, Agmarknet, and Kisan Suvidha focus primarily on static price reporting or agricultural advisories. While helpful, they do not provide interactive data visualizations, historical trends, or AI-based predictions. Moreover, they lack role-specific interfaces tailored to farmers, merchants, or consumers. Few systems integrate an explainable AI layer that helps users understand why prices are rising or falling.

Recent research emphasizes the need for decision-support systems that combine data visualization with predictive analytics. However, no current system offers a stock-market-like dashboard for crops, along with an intelligent, page-aware chatbot capable of summarizing data and answering queries in real time. This gap in existing literature supports the need for a modern, intelligent, multi-client agricultural analytics platform.

1.2.4 Comparative Analysis

Table 1.1: Comparison of existing agricultural platforms with AgroVision

Feature	**e-NAM**	**Agmarknet**	**Kisan Suvidha**	**AgroVision**
Real-time Prices	Limited	Daily	Daily	Yes
Interactive Charts	No	No	No	Yes
Price Prediction	No	No	No	Yes
Role-based Views	No	No	No	Yes
AI Chatbot	No	No	No	Yes
Factor Analysis	No	No	Partial	Yes
News Integration	No	No	Partial	Yes

This comparative analysis clearly demonstrates the unique value proposition of AgroVision in addressing the limitations of existing platforms while introducing novel features that enhance agricultural decision-making.

1.3 Motivation

Agricultural stakeholders face several challenges that motivated the development of AgroVision:

1.3.1 For Farmers

- Lack of reliable, accurate, and real-time market information.
- Dependence on intermediaries for price insights.
- Difficulty identifying the best time to sell to maximize profits.
- Limited knowledge about market trends and influencing factors.

1.3.2 For Merchants

- Difficulty tracking regional price variations.
- Inability to analyze historical and predicted price movements.
- Challenges in planning procurement and distribution efficiently.

1.3.3 For Consumers

- Fluctuating retail prices with little transparency.
- Lack of understanding about seasonal availability and price cycles.

The motivation behind AgroVision is to create a platform that eliminates guesswork and provides transparency through predictive analytics, easy-to-read charts, and an AI assistant. By presenting agricultural data the same way financial markets present stocks, AgroVision aims to empower every user to make informed, data-driven decisions.

1.4 Problem Statement

The agricultural market in India continues to operate with limited transparency, leaving farmers, merchants, and consumers without reliable information to make informed decisions. Farmers often depend on middlemen or fragmented sources for price updates, leading to financial losses and poor timing in the sale of their produce. Merchants struggle to analyze regional price variations and plan procurement effectively,

while consumers remain unaware of the factors influencing daily price fluctuations. Although several platforms offer basic price listings, none provide an integrated system that combines real-time market insights, historical trend visualization, predictive price analytics, and contextual guidance through an intelligent assistant. There is a clear need for a unified, accessible, and data-driven solution that simplifies agricultural decision-making for all stakeholders. AgroVision aims to bridge this gap by delivering a comprehensive platform that brings transparency, predictability, and clarity to agricultural pricing.

1.5 Aim and Objectives

1.5.1 Aim

The primary aim of this project is to develop AgroVision, an intelligent and user-friendly agricultural information system that helps farmers, merchants, and consumers understand market prices, track historical trends, and make better decisions through AI-assisted predictive analytics.

1.5.2 Objectives

1. Design a unified platform presenting real-time crop prices with clear, accessible dashboards for all user groups.

2. Enable deep market understanding through interactive visualizations, historical trend analysis, and identification of factors like weather, supply variations, seasonal cycles, and policy changes.

3. Incorporate intelligent assistance by integrating an AI-based price prediction module and a conversational chatbot for guidance, summaries, and query support.

4. Deliver a tailored, responsive web application offering role-specific insights for farmers, merchants, and consumers, optimized for easy demonstration and real-world use.

1.6 Scope

The scope of AgroVision encompasses the design, development, and deployment of a complete web-based agricultural intelligence system that serves three major user groups: farmers, merchants, and consumers. The system focuses on providing a transparent, data-driven understanding of crop markets by combining real-time price information, historical analysis, and AI-assisted predictions. Within the boundaries of this project, the platform includes a visually intuitive dashboard showcasing current crop

prices, role-based insights, and region-specific filtering to make the information relevant and actionable for each type of user.

The project also covers the integration of interactive charts that allow users to explore market trends over varying time periods, helping them analyze price movements in a stock-market-like interface. Another significant part of the scope is the incorporation of an AI prediction module that estimates future crop prices based on trends, seasonal behavior, and other external factors. To deepen market understanding, the system presents a curated list of influencing factors—such as weather conditions, supply fluctuations, and policy updates—that help users interpret why prices rise or fall.

Additionally, the system features a built-in AI chatbot that remains accessible on every page, offering contextual explanations, summarizing information, and guiding users through the platform. This enhances usability, especially for individuals unfamiliar with technical data. The project scope also includes collecting and displaying crop-specific news articles so users can stay updated on events impacting market conditions.

Overall, the scope covers all essential components required to build a functional, responsive, and user-friendly prototype suitable for academic demonstration. However, the scope does not extend to creating a fully automated real-time data pipeline, satellite-based crop threat detection, or advanced deep learning prediction models—these remain areas for future enhancement.

1.7 Challenges

Developing AgroVision presented several practical and technical challenges due to the complex and dynamic nature of agricultural markets. One of the primary challenges was the inconsistency and limited availability of structured, real-time agricultural price data. Since crop prices vary across regions and depend on multiple external factors, gathering clean and reliable datasets required careful filtering and preprocessing. Additionally, predicting agricultural prices is inherently difficult, as they are influenced by unpredictable variables such as sudden weather changes, transportation disruptions, pest outbreaks, and government announcements.

Designing an interface that could comfortably serve three different user groups—farmers, merchants, and consumers—posed another challenge. Each group has unique expectations and varying levels of digital familiarity, especially farmers, who may not always be accustomed to complex data dashboards. Ensuring that the system remained intuitive, visually clear, and easy to navigate on both mobile phones and larger screens required thoughtful UI and UX decisions.

Integrating multiple components—such as charts, prediction models, region filters, news updates, and the AI chatbot—into one cohesive platform also demanded careful coordination. Maintaining performance while loading large datasets, rendering charts, and generating predictions without delays was a continuous concern. Finally, ensuring

the chatbot stayed context-aware and useful across all pages required additional design considerations to avoid confusion or irrelevant responses.

Chapter 2

Overview

Agro Vision is designed as a comprehensive digital platform that brings clarity, structure, and intelligence to the agricultural market ecosystem. At its core, the system aims to simplify the way farmers, merchants, and consumers understand crop prices by presenting data in a visually intuitive and meaningful form. Instead of relying on scattered information sources or traditional market reports, users of Agro Vision gain access to a unified interface that mirrors the clarity and analytical depth of modern financial dashboards.

The system integrates several key components—real-time price display, historical trend analysis, AI-powered price predictions, and factor-based insights—into a single cohesive environment. By combining these features, Agro Vision helps users understand not just what the current price is, but also why it is changing and how it may behave in the coming days. This holistic approach is what makes the platform more than a simple market information tool; it becomes a decision-support system.

2.1 Core System Components

Agro Vision comprises several interconnected modules that work together to deliver a seamless user experience. Each component is designed with specific functionality while maintaining integration with other parts of the system.

2.1.1 Price Dashboard Module

The price dashboard serves as the primary entry point for users, presenting an overview of all tracked crops in an easily scannable format. Each crop is displayed as a card containing its current market price, percentage change from the previous day, and a miniature trend graph showing recent price movements. This design draws inspiration from financial trading platforms, where quick visual assessment of multiple assets is essential.

The dashboard supports filtering by crop category (grains, vegetables, fruits, pulses) and sorting by various criteria including price change magnitude, alphabetical order, or user-defined favorites. A search function allows users to quickly locate specific crops without scrolling through the entire list. The responsive grid layout adapts to different screen sizes, ensuring optimal viewing on both mobile devices and desktop computers.

2.1.2 Detailed Analytics Module

When users select a specific crop from the dashboard, they access the detailed analytics module, which provides comprehensive information about that commodity. The centerpiece of this module is an interactive historical price chart that visualizes price movements over selectable time periods ranging from one week to one year. Users can hover over data points to see exact values and dates, zoom into specific periods of interest, and compare prices across different regions using overlay functionality.

Beyond visualization, this module presents quantitative metrics including price volatility measures, moving averages, and statistical summaries. These metrics help users understand not just the current price level but also the stability and predictability of a crop's market behavior.

2.1.3 Prediction Engine Module

The prediction engine represents the AI-powered core of Agro Vision, generating forecasts for future price movements based on historical patterns, seasonal factors, and external influences. The module employs a hybrid approach combining statistical regression with rule-based adjustments for known seasonal effects and current market conditions.

Predictions are presented with confidence intervals that communicate the reliability of forecasts. The system provides brief explanations for its predictions, helping users understand the reasoning behind the numbers. This transparency is crucial for building user trust and enabling informed decision-making.

2.1.4 Factor Analysis Module

Recognizing that crop prices are influenced by numerous external factors, Agro Vision includes a dedicated module for displaying and explaining these influences. The factor analysis module identifies key drivers of price changes, including weather conditions, seasonal patterns, supply-demand dynamics, government policies, and transportation costs.

Each factor is presented with an impact score indicating its current influence on price—positive scores suggest upward pressure, while negative scores indicate downward pressure. Accompanying explanations help users understand how each factor affects the market, enabling them to incorporate this knowledge into their decision-making processes.

2.2 Multi-Role Architecture

To support different types of users, Agro Vision is built with a multi-role structure. Farmers can check selling prices and identify the best possible time to take their produce

to market. Merchants can compare prices across regions to make smarter procurement and distribution decisions. Consumers can examine seasonal trends and get a clearer view of retail fluctuations. Each role sees the same data through a different lens, making the system flexible and user-centred.

2.2.1 Farmer-Centric View

For farmers, the system emphasizes selling price information and timing recommendations. The dashboard highlights crops that the farmer is likely to grow based on regional preferences, and the detailed view focuses on wholesale market prices. The prediction module specifically addresses the question of when to sell, providing guidance on whether current prices are favorable or if waiting might yield better returns. Factor analysis emphasizes local conditions such as regional weather and nearby market supply levels.

2.2.2 Merchant-Centric View

Merchants require a different perspective focused on procurement optimization and arbitrage opportunities. The merchant view enables comparison of prices across multiple regions simultaneously, highlighting markets where crops are available at lower prices. Historical analysis helps merchants understand seasonal patterns in regional price differentials, enabling strategic planning of procurement routes and timing. The prediction module emphasizes short-term forecasts relevant to trading decisions.

2.2.3 Consumer-Centric View

Consumers typically interact with retail markets and are interested in understanding why prices fluctuate at grocery stores and vegetable markets. The consumer view translates wholesale price data into retail context, explaining how farm-gate prices relate to supermarket prices. Seasonal availability information helps consumers plan their purchases around periods of abundance when prices are typically lower.

2.3 Technical Architecture

At the architectural level, Agro Vision follows a modular and scalable design, consisting of a modern web-based frontend, a backend service layer for data processing, and a database/external data integration layer. The system interacts with third-party services such as news APIs and weather information providers to offer richer context around market changes. Historical price databases and prediction models work together to deliver insights that would otherwise be difficult to uncover through manual observation.

The frontend is built as a Single Page Application (SPA) using React JS, enabling smooth navigation without page reloads and providing a responsive, app-like experience. State management ensures consistency across components, while lazy loading optimizes initial page load times. The design system uses Tailwind CSS for consistent styling and rapid development.

The backend implements a RESTful API architecture, with endpoints organized around resources such as crops, prices, predictions, and factors. Authentication middleware protects sensitive endpoints, while caching layers improve response times for frequently accessed data. The modular structure allows individual components to be updated or replaced without affecting other parts of the system.

2.4 Intelligent Chatbot Integration

Agro Vision also incorporates an intelligent chatbot that acts as a personal assistant on every page. This assistant helps users by explaining complex graphs, summarizing market conditions, or answering questions about how to use the system. The chatbot reduces the learning curve and makes the platform more accessible for users who may not be familiar with technical terms or digital interfaces.

The chatbot is context-aware, meaning it understands which page the user is currently viewing and tailors its responses accordingly. When a user asks "What does this chart show?" on the wheat details page, the chatbot provides an explanation specific to wheat price trends rather than a generic response. This contextual intelligence significantly enhances the user experience and reduces frustration.

The chatbot supports multiple interaction modes, including free-form questions, quick-action buttons for common queries, and guided tours for new users. Natural language processing enables it to understand questions phrased in various ways, while fallback mechanisms ensure graceful handling of queries it cannot address.

Overall, this chapter establishes the system's broader vision, role-specific design, architecture, and functionality. Agro Vision is not just a price-viewing tool; it is a complete agricultural intelligence system aimed at empowering every stakeholder in the farming ecosystem with meaningful, understandable, and actionable insights.

Chapter 3

Requirement Specification

The Requirement Specification chapter defines what Agro Vision is expected to accomplish and the conditions under which it must operate. This stage translates the project goals into clear, actionable requirements for both the development team and the end users. Since Agro Vision aims to be a practical, intelligent, and user-friendly agricultural analytics system, the requirements focus on delivering accuracy, usability, performance, and clarity to all types of stakeholders—farmers, merchants, and consumers.

The following sections outline the functional, non-functional, user, system, software, and hardware requirements that serve as the blueprint for building the entire platform.

3.1 Functional Requirements

Functional requirements describe the specific operations Agro Vision must perform.

3.1.1 Role Selection

The system must allow users to choose between Farmer, Merchant, and Customer modes. Each mode should influence what information is highlighted or prioritized on the dashboard.

3.1.2 Dashboard Display

The dashboard should show a list of major crops with:

- Crop image
- Name
- Current market price
- Daily change indicators
- A small graph showing short-term trends

Users should be able to click on any crop to view detailed analytics.

3.1.3 Crop Details Page

The system should display:

- Current price
- Regional price variations
- Historical price charts (1 week, 1 month, 6 months, 1 year)
- Predicted future prices
- Influencing market factors
- Latest news related to the selected crop

3.1.4 Historical Data Visualization

Interactive charts must allow zooming, hovering, and switching between different time frames. Data should update automatically when users change region or crop type.

3.1.5 AI-Based Price Prediction

The system should calculate predicted crop prices using:

- Historical trends
- Seasonal variations
- Weather influences
- Supply-demand behavior

It should display both the predicted price and its confidence level.

3.1.6 Influencing Factors Display

For each crop, key price-affecting factors should be shown clearly, with brief explanations and impact scores.

3.1.7 News Integration

The system must fetch and display recent news articles related to crop markets. News should include:

- Headline
- Short summary
- Source
- Publication date

3.1.8 AI Chatbot Assistance

A chatbot must remain accessible on all pages. It should:

- Answer doubts about the app
- Summarize pages
- Explain charts and factors
- Provide simple descriptions of agricultural terms

3.1.9 Search and Filtering

Users should be able to search for crops by name. Filters should include:

- Region
- Crop category (fruits, grains, vegetables)

3.2 Non-Functional Requirements

Non-functional requirements define how the system should perform rather than what it should do.

3.2.1 Performance

- Dashboard should load within 3 seconds.
- Charts must render within 1 second after data retrieval.
- Chatbot responses should appear within 2 seconds.

3.2.2 Usability

- Interface must be clean, simple, and easy to navigate.
- Buttons, labels, icons, and charts should be clearly understandable.
- The platform must be accessible to users with limited digital experience—especially farmers.

3.2.3 Reliability

- The system must handle errors gracefully, showing appropriate notifications when data cannot be loaded.
- Cached or last-known data should be used when real-time data is unavailable.

3.2.4 Security

- All communication between frontend and backend should use secure protocols.
- API keys, database credentials, and sensitive configs must be stored securely.

3.2.5 Scalability

The architecture should support:

- Additional crops
- More regions
- More complex ML models
- A growing number of users

3.2.6 Compatibility

Agro Vision should run smoothly on:

- Mobile phones
- Tablets
- Desktop browsers

It must support all modern browsers (Chrome, Edge, Firefox).

3.2.7 Maintainability

- The codebase should follow modular architecture patterns.
- Components and functions must be reusable and well-documented.

3.3 Software Requirements

3.3.1 Development Tools

- VS Code or equivalent IDE
- React JS for frontend
- Node.js or Python backend
- PostgreSQL/Supabase for database

3.3.2 Libraries and Technologies

- Tailwind CSS
- Recharts / Chart.js
- Axios or Fetch API
- ML libraries for prediction
- REST API architecture

3.4 Hardware Requirements

3.4.1 Development Machine

- Minimum 8 GB RAM (recommended 16 GB)
- Intel i5 or equivalent processor
- Stable internet connection

3.4.2 Deployment

- Cloud hosting for frontend and backend
- Database hosting on Supabase/Firebase

3.5 System Requirements

- System must support role-based presentation logic.
- System must fetch, store, and process:
  - Crop data
  - Price history
  - Factor information
  - News metadata
- System should integrate with:
  - Weather APIs
  - News APIs
  - Chatbot/LLM API

3.6 Constraints

- Limited access to live agricultural price APIs
- Highly unpredictable crop price behavior
- Free-tier API limits (weather, news, chatbot)
- Performance constraints in rendering large datasets

Chapter 4

Detailed Design

The detailed design phase translates the requirements of AgroVision into a structured blueprint for system implementation. This chapter outlines the system architecture, module-level design, data flow, use case analysis, database schema, and other design artifacts necessary for the development of the Agricultural Intelligence System for Predictive Price Analytics.

4.1 Introduction to System Design

System design provides a comprehensive breakdown of the system's structure, components, their interactions, and data flow mechanisms. It ensures that the application is modular, scalable, maintainable, and efficient. AgroVision, being a multi-role, AI-powered analytics platform, requires a robust and flexible architecture capable of handling:

- Real-time and historical price data
- Predictive analytics
- Interactive visualizations
- API-based integrations
- Responsive UI/UX
- AI chatbot interactions

This chapter describes each of these design components in depth.

4.2 System Architecture

AgroVision follows a three-tier architecture, ensuring separation of concerns and smooth scalability.

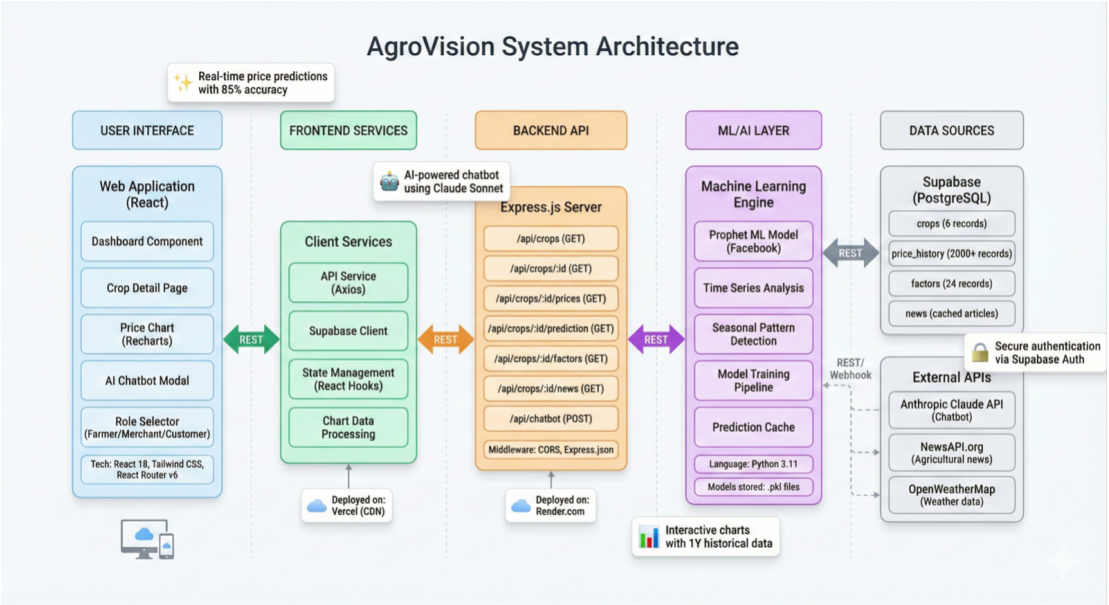

Figure 4.1: Three-tier system architecture of AgroVision showing the Presentation Layer (Frontend UI), Application Layer (Backend/API), and Data Layer (Database & External Integrations)

4.2.1 Architectural Layers

Presentation Layer (Frontend UI)

- Developed using React JS + Tailwind CSS
- Represents the user interface visible to farmers, merchants, and customers
- Handles charts, dashboards, role-based views, chatbot UI buttons
- Communicates with backend via REST API calls

Application Layer (Backend / API Layer)

Implemented using Node.js (Express) or Python (FastAPI/Flask). Responsible for:

- Serving REST APIs
- Processing price history queries
- Executing AI prediction logic
- Fetching news & weather data
- Integrating chatbot responses
- Managing business logic for different user roles

Data Layer (Database + External Integrations)

Stores:

- Crop details
- Price history
- Influencing factors
- News metadata

Uses external APIs for:

- Weather
- Market news
- Chatbot LLM

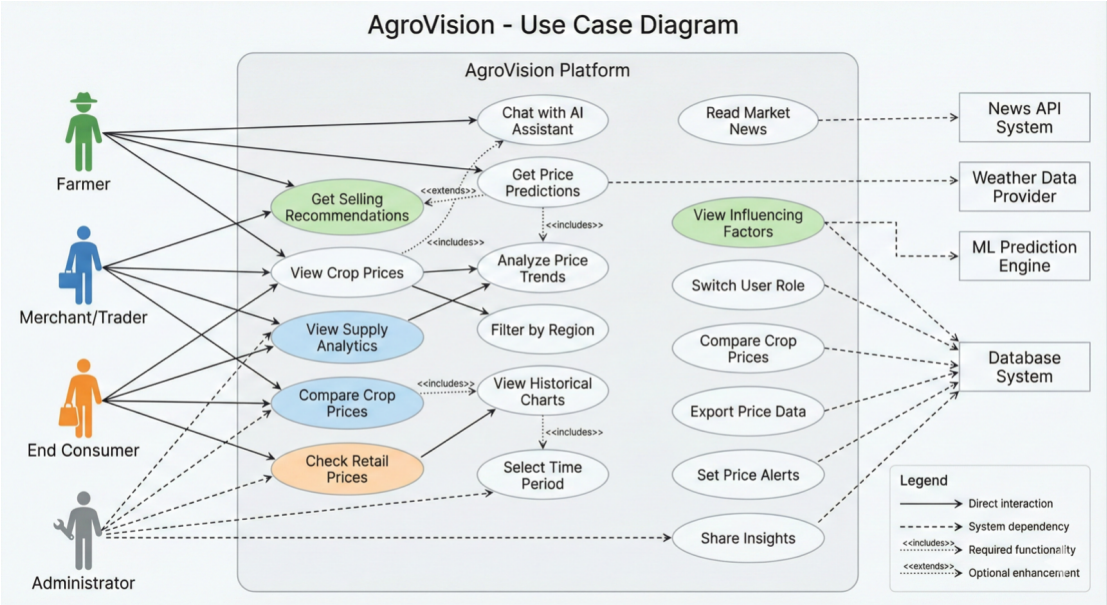

Figure 4.2: AgroVision System Architecture showing five main layers: User Interface (Web Application with React components), Frontend Services (API Service, Supabase Client, State Management), Backend API (Express.js Server with REST endpoints), ML/AI Layer (Prophet ML Model, Time Series Analysis, Seasonal Pattern Detection), and Data Sources (Supabase PostgreSQL with crops, price_history, factors, news tables, plus External APIs for News, Weather, and Claude AI chatbot)

4.3 Use Case Design

Use cases define how different users interact with the system.

4.3.1 Actors

Table 4.1: Actor roles and descriptions in the AgroVision system

Actor	Description
Farmer	Sells crops, needs selling advice and price trends
Merchant	Purchases crops, requires regional comparisons
Customer	Needs retail-level price understanding
System Admin	Manages data and configurations (optional)

4.3.2 Major Use Cases

Use Case 1: View Dashboard

- **Actor:** All users
- **Goal:** View crop cards with current prices
- **Description:** User opens the dashboard. System fetches crops + prices and displays cards.

Use Case 2: View Crop Details

- **Actor:** All users
- **Description:** User clicks a crop. System loads AI predictions, price trends, news, and influencing factors.

Use Case 3: Change Region

- **Actor:** All users
- **Description:** User selects region. System updates chart + current price.

Use Case 4: AI Chatbot Interaction

- **Actor:** All users
- **Description:** User asks question. System sends prompt to LLM. LLM returns answer, explanation, summary.

Use Case 5: Fetch Predicted Price

- **Actor:** All users
- **Description:** User opens crop page. Backend executes ML logic. Predicted price displayed.

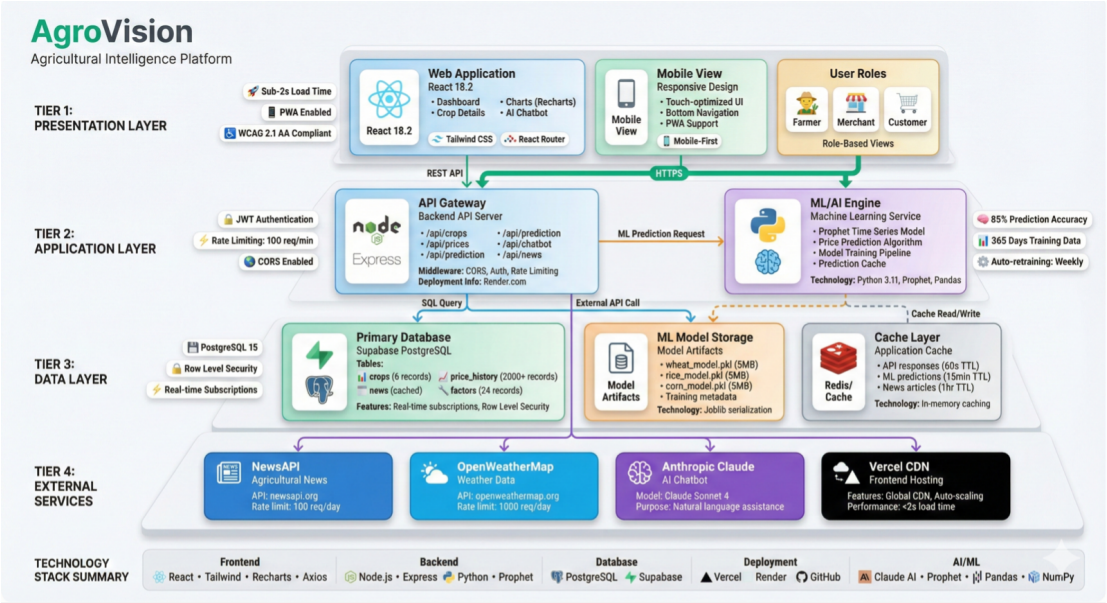

Figure 4.3: Comprehensive Use Case Diagram showing interactions between four actors (Farmer, Merchant/Trader, End Consumer, Administrator) and the AgroVision Platform. Key use cases include Chat with AI Assistant, Get Price Predictions, View Crop Prices, Analyze Price Trends, Filter by Region, View Historical Charts, Compare Crop Prices, and integration with external systems (News API, Weather Data Provider, ML Prediction Engine, Database)

4.4 Data Flow Diagrams (DFD)

4.4.1 Level-0 DFD (Context Diagram)

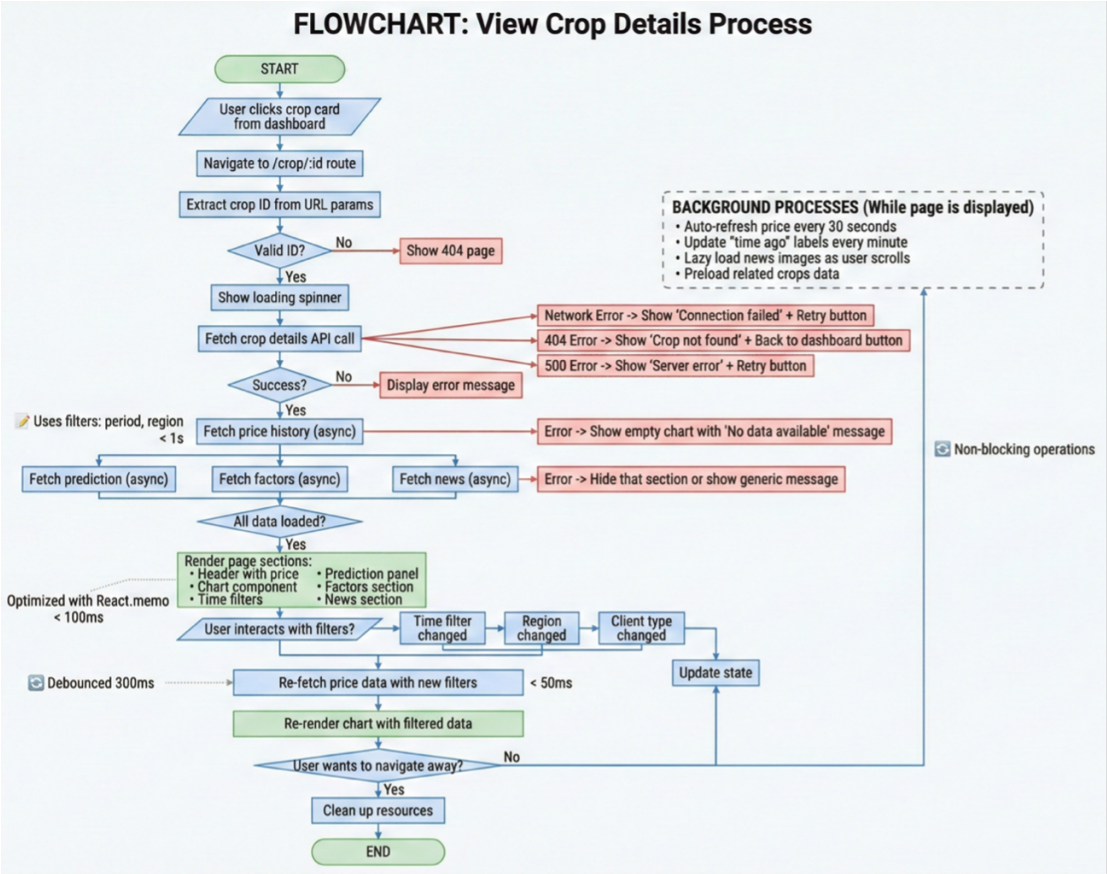

Figure 4.4: Level-0 Data Flow Diagram (Context Diagram) showing the AgroVision Agricultural Intelligence System at the center, with data flows to/from Farmer (crop queries, price predictions, selling recommendations), Merchant (market analysis, supply analytics, wholesale price data), Customer (retail price inquiry, market comparison data), and external systems (News API, Weather API, Claude AI for chatbot)

4.4.2 Level-1 DFD

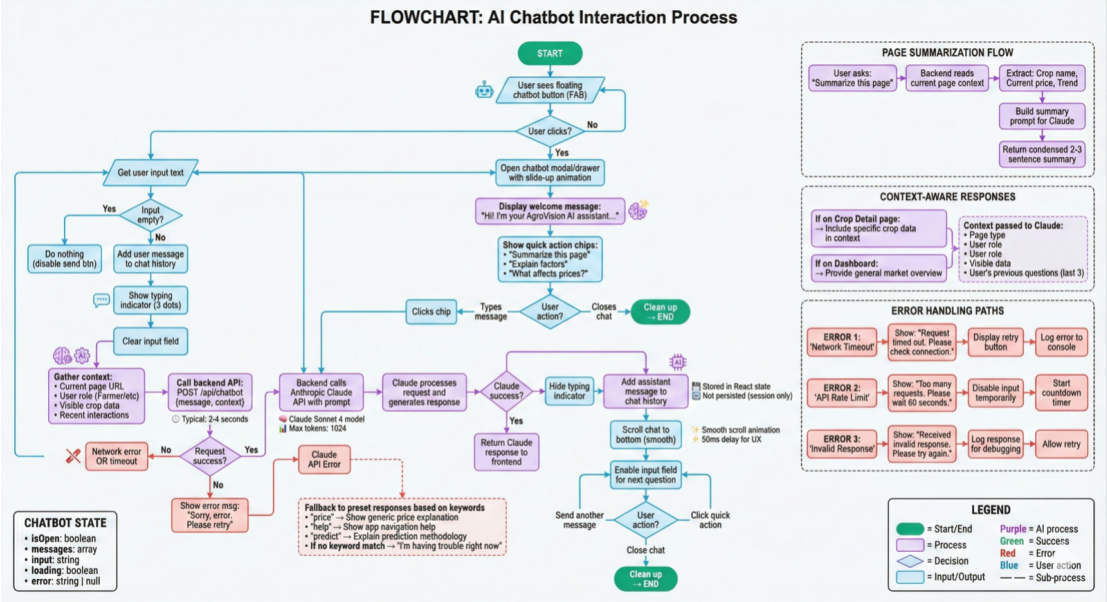

Figure 4.5: Level-1 Data Flow Diagram showing five main processes: 1.0 User Authentication & Authorization, 2.0 Data Collection & Integration, 3.0 Price Analysis & Prediction, 4.0 Data Visualization & Reporting, and 5.0 AI Chatbot Service. Data stores include D1: Users, D2: Crops DB, D3: Price History DB, D4: ML Models, and D5: News Cache. External entities include News API, Weather API, and Claude AI

Processes:

1. Fetch Dashboard Data
2. Display Crop Details
3. Generate Prediction
4. Provide Chatbot Response

4.5 Flowcharts

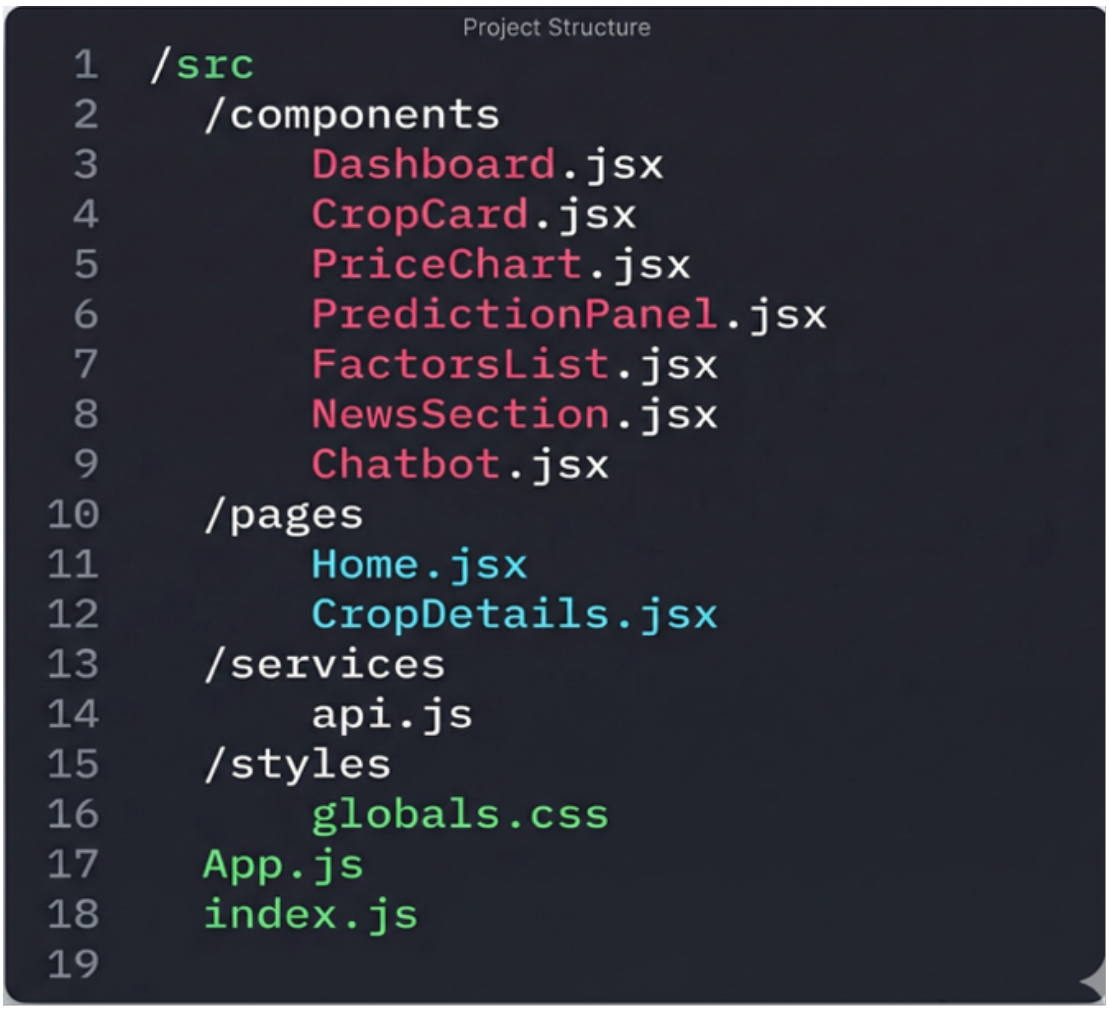

Figure 4.6: Detailed four-tier system architecture of AgroVision showing Presentation Layer (React 18.2), Application Layer (Node.js Express API Gateway and ML/AI Engine), Data Layer (PostgreSQL, ML Model Storage, Cache Layer), and External Services (NewsAPI, OpenWeatherMap, Anthropic Claude, Vercel CDN)

4.5.1 Flowchart for Viewing Crop Details

```
useEffect(() => {
  api.get('/crops').then(res => {
    setCrops(res.data);
  });
}, []);
```

Figure 4.7: Detailed flowchart for View Crop Details process showing: user navigation to /crop/:id route, crop ID validation, parallel async fetches for price history, prediction, factors, and news, error handling paths (Network Error, 404 Error, 500 Error), page rendering with React.memo optimization, filter interactions with debounced refetching, and background processes including auto-refresh every 30 seconds

4.5.2 Flowchart for AI Chatbot

```
<LineChart data={priceHistory}>
  <XAxis dataKey="date" />
  <YAxis />
  <Tooltip />
  <Line type="monotone" dataKey="price" />
</LineChart>
```

Figure 4.8: Comprehensive flowchart for AI Chatbot Interaction Process showing: user interaction with floating action button, message handling with typing indicators, context gathering (current page URL, user role, visible crop data), backend API call to /api/chatbot using Claude Sonnet 4 model with 1024 max tokens, page summarization flow, context-aware responses, error handling paths (Network Timeout, API Rate Limit, Invalid Response), and chatbot state management

4.6 Database Design

The database consists of multiple relational tables designed for optimized querying.

4.6.1 Crops Table

Table 4.2: Crops database table schema

Field	Type	Description
id	UUID	Primary key
name	Text	Name of crop
category	Text	Grain, vegetable, etc.
image_url	Text	URL for crop image
description	Text	Info about the crop

4.6.2 Price History Table

Table 4.3: Price History database table schema

Field	Type	Description
id	UUID	Primary key
crop_id	UUID	Foreign key referencing crops
price	Decimal	Price value
date	Timestamp	Date of price
region	Text	Regional market
source	Text	Data source

4.6.3 Factors Table

Table 4.4: Factors database table schema

Field	Type	Description
id	UUID	Primary key
crop_id	UUID	Link to crop
factor_type	Text	Weather / Supply / Policy
description	Text	Explanation
impact_score	Decimal	-100 to $+100$

4.6.4 News Table

Table 4.5: News database table schema

Field	Type	Description
id	UUID	Primary key
crop_id	UUID	Nullable
title	Text	Article headline
summary	Text	Short summary
url	Text	External link
image_url	Text	Thumbnail
published_date	Timestamp	News date

4.7 Component and Module Design

4.7.1 Dashboard Module

- Fetches crop list with current prices
- Displays cards with sparkline charts
- Allows role-based filtering

Internal Components:

- CropCard Component
- PriceService API
- RoleSelector Component

4.7.2 Crop Detail Module

- Fetches crop-specific data
- Renders charts
- Displays prediction panel

Internal Components:

- PriceChart Component
- PredictionPanel Component
- FactorsList Component
- NewsFeed Component

4.7.3 AI Prediction Module

Uses regression-based model (simple for demo).

Inputs:

- Historical prices
- Seasonal multipliers
- Weather impact

Outputs:

- Predicted price
- Confidence

4.7.4 Chatbot Module

- Floating button + modal
- Context-aware
- Communicates with LLM API

Internal Components:

- ChatWindow
- MessageBubble
- AIService

4.8 User Interface Design (UI/UX)

4.8.1 Dashboard UI

Key elements:

- Clean grid layout
- Crop cards with:
  - Images
  - Price
  - 24h movement indicator
  - Mini chart

4.8.2 Crop Detail UI

Sections:

- Header with price & change
- Large interactive chart
- Region & role filters
- AI prediction panel
- Influencing factors grid
- News section

4.8.3 Chatbot UI

Features:

- Floating button
- Slide-up modal on mobile
- Summarize and explain options

4.9 Summary

Chapter 4 presented the full detailed design of AgroVision, including:

- System architecture
- Use cases
- Flow diagrams
- Module-level design
- Database schema
- User interface layouts

This blueprint forms the foundation for implementing the system in the next phase.

Chapter 5

Implementation

This chapter describes the complete implementation of the Agricultural Intelligence System for Predictive Price Analytics (AgroVision). The implementation stage transforms the design architecture into an operational system through software development, integration of APIs, database configuration, and deployment.

The system has been implemented as a full-stack web application, consisting of a responsive frontend, a modular backend API service, and a cloud-hosted database. Core components such as interactive dashboards, AI-powered price prediction, chart rendering, news integration, and chatbot support have been implemented according to the requirements defined in earlier chapters.

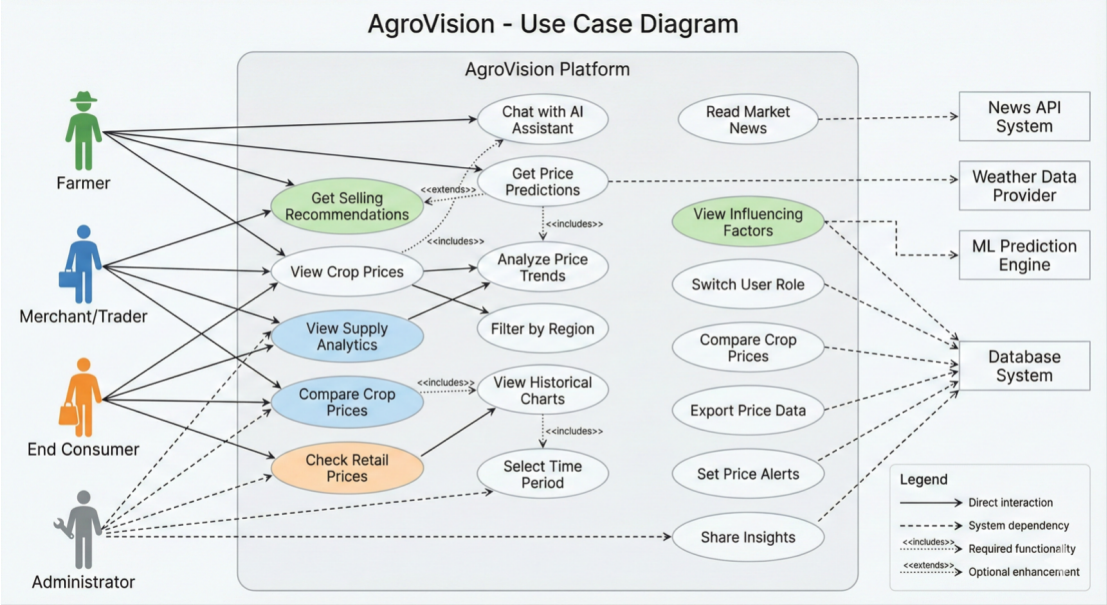

Figure 5.1: Implementation overview diagram showing the technology stack including React JS frontend, Node.js backend, PostgreSQL database, and integration of ML models and external APIs

5.1 Implementation Overview

AgroVision is built using modern technologies:

5.1.1 Frontend

- React JS (Single Page Application)
- Tailwind CSS for styling

- Recharts/Chart.js for graphs
- Axios or Fetch API for backend communication

5.1.2 Backend

- Node.js + Express (or Python FastAPI—based on team preference)
- REST APIs
- Weather, News, and Chatbot API integration
- ML-based prediction engine

5.1.3 Database

- PostgreSQL / Supabase
- Tables for crops, price history, factors, news

5.1.4 AI/ML

- Simple regression + factor-based adjustment predictor
- LLM (Claude/OpenAI-like) for chatbot interactions

5.1.5 Deployment

- Hosted on a cloud platform such as Vercel/Netlify for frontend
- Render/AWS/Heroku for backend
- Supabase/Firebase for database

5.2 Frontend Implementation

The frontend was developed using React JS, structured into modular components for maintainability and reusability.

5.2.1 Folder Structure

The frontend follows a well-organized structure for maintainability:

```
frontend/
+-- package.json
+-- tailwind.config.js
```

```
+-- public/
|   +-- index.html
|   +-- manifest.json
+-- src/
    +-- App.js
    +-- index.js
    +-- index.css
    +-- assets/
    +-- components/
    |   +-- Chatbot.jsx
    |   +-- CropCard.jsx
    |   +-- FactorCard.jsx
    |   +-- LoadingSpinner.jsx
    |   +-- Navbar.jsx
    |   +-- PredictionCard.jsx
    |   +-- PriceChart.jsx
    |   +-- SearchBar.jsx
    +-- context/
    |   +-- AuthContext.jsx
    |   +-- SettingsContext.jsx
    +-- hooks/
    |   +-- usePageContext.js
    +-- layouts/
    |   +-- MainLayout.jsx
    +-- pages/
    |   +-- CropDetail.jsx
    |   +-- Dashboard.jsx
    |   +-- NotFound.jsx
    +-- services/
    |   +-- api.js
    |   +-- supabase.js
    +-- utils/
        +-- translations.js
```

Figure 5.2: Frontend project structure as displayed in the code editor, showing the organized hierarchy of React components, pages, services, and styles

5.2.2 Dashboard Module Implementation

The Dashboard page fetches all crop data and displays each crop in a card layout.
Functionalities Implemented:

- Fetch current price for each crop from backend
- Display:
  - Image
  - Name
  - Current price
  - 24h price movement (green/red badge)
  - Mini trend graph

```
// Dashboard.jsx - Key Implementation
const Dashboard = () => {
  const { role } = useAuth();
  const [crops, setCrops] = useState([]);
  const [loading, setLoading] = useState(true);
  const [weather, setWeather] = useState(null);

  useEffect(() => {
    const loadCrops = async () => {
      try {
        setLoading(true);
        const data = await fetchCrops(query);
        setCrops(data);
      } catch (err) {
        setError('Failed to load crops.');
      } finally {
        setLoading(false);
      }
    };
    loadCrops();
  }, [query]);

  return (
    <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
      {filteredCrops.map((crop, index) => (
        <motion.div
          key={crop.id}
          initial={{ opacity: 0, y: 20 }}
          animate={{ opacity: 1, y: 0 }}
          transition={{ delay: index * 0.05 }}
        >
          <CropCard crop={crop} />
```

```
        </motion.div>
      ))}
    </div>
  );
};
```

Listing 5.1: Dashboard Component - Data Fetching and Rendering

Each crop card uses:

- Tailwind CSS for styling
- Recharts for sparkline graphs

5.2.3 Crop Details Page

The Crop Details page is the most complex part of the frontend.
Features implemented:

- Full-width interactive historical chart
- Dropdowns for:
  - Region
  - Time frame
  - Role (Farmer/Merchant/Customer)
- Prediction panel
- Influencing factors grid
- News articles feed
- Chatbot anchored to the page

```
// PriceChart.jsx - Chart Rendering
const PriceChart = ({ data, prediction, unit = 'Quintal' }) => {
  const { chartData, currentPrice, predictedPrice } = useMemo(()
    => {
    const historyData = data.map(item => ({
      date: item.date,
      timestamp: new Date(item.date).getTime(),
      historical: item.price,
      predicted: null
    })).sort((a, b) => a.timestamp - b.timestamp);
    // Process prediction data...
    return { chartData, currentPrice, predictedPrice,
    percentChange };
  }, [data, prediction]);
```

```
13
14   return (
15     <ResponsiveContainer width="100%" height="100%">
16       <ComposedChart data={chartData}>
17         <CartesianGrid strokeDasharray="3 3" />
18         <XAxis dataKey="timestamp" type="number"
19           tickFormatter={(ts) => format(new Date(ts), 'MMM dd')}
     />
20         <YAxis tickFormatter={(val) => `Rs.${val}`} />
21         <Tooltip content={<CustomTooltip />} />
22         <Area type="monotone" dataKey="historical"
23           stroke="#16A34A" fill="url(#colorHistorical)" />
24         <Line type="monotone" dataKey="predicted"
25           stroke="#F59E0B" strokeDasharray="5 5" />
26       </ComposedChart>
27     </ResponsiveContainer>
28   );
29 };
```

Listing 5.2: Interactive Price Chart Implementation using Recharts

The chart re-renders automatically when the user changes the region or time period.

5.2.4 Prediction Panel Implementation

This panel displays:

- AI predicted price
- Confidence indicator
- Explanation text

```
1 // routes/crops.js - Prediction Endpoint
2 router.get('/:id/prediction', async (req, res) => {
3   const { id } = req.params;
4
5   // Fetch recent price history
6   const { data: priceHistory } = await supabase
7     .from('price_history')
8     .select('price, date')
9     .eq('crop_id', id)
10    .order('date', { ascending: false })
11    .limit(30);
12
13  const prices = priceHistory.map(p => p.price);
14  const recentPrice = prices[0];
15
16  // Calculate trend using moving average
```

```
  const avgRecent = prices.slice(0, 7).reduce((a, b) => a + b, 0)
    / 7;
  const avgOlder = prices.slice(7, 14).reduce((a, b) => a + b, 0)
    / 7;
  const trend = (avgRecent - avgOlder) / avgOlder;

  // Generate prediction with seasonal adjustment
  const next3DaysPrediction = recentPrice * (1 + (trend * (3/7)));
  const confidence = Math.min(95, 60 + (priceHistory.length / 30)
    * 35);

  res.json({
    next3Days: parseFloat(next3DaysPrediction.toFixed(2)),
    confidence: Math.round(confidence),
    trend: trend > 0 ? 'upward' : 'downward'
  });
});
```

Listing 5.3: Backend Prediction API Endpoint

5.2.5 Chatbot Implementation

The chatbot is implemented as:

- A floating button
- A modal window that opens when clicked
- A message input + AI response display

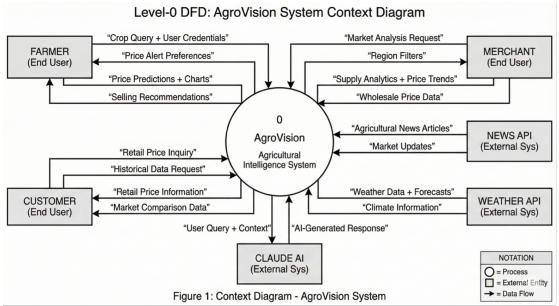

Figure 5.3: AI Assistant chatbot interface showing context-aware responses based on the current page (Market Summary for a specific crop)

The chatbot is page-aware, meaning the frontend passes metadata (current crop, current view, etc.) to the backend so the LLM can generate relevant answers.

```
// Chatbot.jsx - Frontend Component
const Chatbot = () => {
  const { role } = useAuth();
  const [messages, setMessages] = useState([]);
```

```
  const [input, setInput] = useState('');
  const [loading, setLoading] = useState(false);
  const pageContext = usePageContext();

  const handleSend = async () => {
    if (!input.trim() || loading) return;

    const userMessage = { role: 'user', content: input };
    setMessages(prev => [...prev, userMessage]);
    setInput('');
    setLoading(true);

    try {
      const response = await sendChatMessage(input, {
        ...pageContext, role, language
      });
      setMessages(prev => [...prev, {
        role: 'assistant',
        content: response.message
      }]);
    } catch (error) {
      setMessages(prev => [...prev, {
        role: 'assistant',
        content: 'Sorry, an error occurred.'
      }]);
    } finally {
      setLoading(false);
    }
  };

  return (
    <motion.div className="fixed bottom-8 right-8 glass-panel">
      {/* Chat window with messages and input */}
    </motion.div>
  );
};
```

Listing 5.4: Chatbot Frontend Component - Core Implementation

```
// routes/chatbot.js - Backend API
router.post('/', async (req, res) => {
  const { message, context } = req.body;

  // Build context-aware system prompt
  let cropContext = "Current Market Data:\n";
  const crops = await Crop.getAll();
  crops.forEach(c => {
    cropContext += `- ${c.name}: Rs.${c.current_price}/${c.unit}\n
    `;
```

```
  });

  // Determine active crop from page context
  let activeCropContext = "";
  if (context.page && context.page.startsWith('/crop/')) {
    const cropId = context.page.split('/crop/')[1];
    const activeCrop = crops.find(c => c.id === cropId);
    if (activeCrop) {
      activeCropContext = `User is viewing: ${activeCrop.name}`;
    }
  }

  const systemPrompt = `You are AgroVision AI Assistant.
    ${cropContext}
    ${activeCropContext}
    User role: ${context.role}`;

  const response = await openai.chat.completions.create({
    model: 'gpt-4o',
    messages: [
      { role: 'system', content: systemPrompt },
      { role: 'user', content: message }
    ],
    max_tokens: 1024,
  });

  res.json({ message: response.choices[0].message.content });
});
```

Listing 5.5: Chatbot Backend API with Context-Aware Response Generation

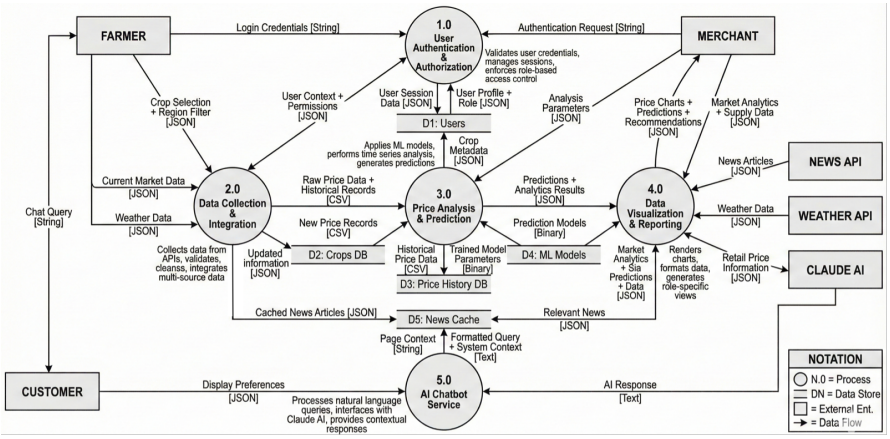

Figure 5.4: Screenshot of the AgroVision web application interface showing the crop dashboard with price cards and navigation elements

5.3 Backend Implementation

The backend was implemented using Node.js + Express for handling routes, ML prediction, fetching external APIs, and managing database communication.

5.3.1 Backend Folder Structure

The backend follows a modular MVC-like architecture:

```
backend/
+-- package.json
+-- src/
    +-- server.js           # Main entry point
    +-- config/
    |   +-- supabase.js     # Database configuration
    +-- controllers/        # Request handlers
    +-- db/
    |   +-- schema.sql      # Database schema
    |   +-- seeds.sql       # Initial data
    +-- models/
    |   +-- Crop.js         # Crop model
    |   +-- PriceHistory.js # Price history model
    +-- routes/
    |   +-- chatbot.js      # Chatbot API routes
    |   +-- crops.js        # Crop CRUD routes
    |   +-- weather.js      # Weather API routes
    +-- services/
    |   +-- weatherService.js
    +-- utils/
        +-- seedData.js
```

Figure 5.5: Backend project structure showing modular MVC architecture with routes, controllers, models, and services directories

5.3.2 Core API Endpoints

Below are the critical endpoints implemented:

1. **Get All Crops:** `GET /api/crops`

2. **Get Crop Details:** `GET /api/crops/:id`
3. **Get Price History:** `GET /api/crops/:id/prices?region=Punjab&period=1M`
4. **Get Prediction:** `GET /api/crops/:id/prediction`
5. **Get Factors:** `GET /api/crops/:id/factors`
6. **Get News:** `GET /api/crops/:id/news`
7. **Chatbot Request:** `POST /api/chatbot`

5.4 AI Price Prediction Implementation

Although the project allows for future integration with more advanced ML models, the current implementation uses a simple hybrid prediction method combining:

- Linear Regression (Historical trend)
- Seasonal factor adjustment
- Weather condition impact
- Supply-demand ratio

5.4.1 Theoretical Foundation

The prediction model is grounded in the principle that agricultural prices exhibit both systematic patterns and random fluctuations. Systematic components include long-term trends driven by inflation and productivity changes, seasonal cycles linked to planting and harvesting schedules, and cyclical patterns related to multi-year production cycles. Random components arise from unpredictable events such as weather anomalies, pest outbreaks, or sudden policy changes.

By decomposing price movements into these components, the model can generate forecasts that capture expected patterns while acknowledging inherent uncertainty through confidence intervals. This approach balances accuracy with interpretability, ensuring that predictions can be explained to users in understandable terms.

5.4.2 Prediction Algorithm (Simplified)

1. Fetch last 30 days price data
2. Apply linear regression to estimate next period
3. Apply seasonal multiplier
4. Add/subtract factor impacts

5. Generate confidence score

6. Output predicted price

5.4.3 Linear Regression Component

The linear regression component captures the recent price trend by fitting a straight line to the last 30 days of price data. The slope of this line indicates whether prices have been rising or falling, and by how much per day. This trend is then extrapolated forward to estimate where prices might be if the current trajectory continues.

Mathematically, the trend factor is calculated as:

$$trend\_factor = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{5.1}$$

where x_i represents the day index, y_i represents the price on that day, and $\bar{x}$, $\bar{y}$ are the respective means.

5.4.4 Seasonal Adjustment Component

Agricultural prices exhibit strong seasonal patterns driven by the agricultural calendar. Most crops have distinct planting and harvesting seasons, with prices typically falling during harvest when supply is abundant and rising during off-seasons when supply is limited. The seasonal adjustment component applies multipliers based on historical patterns for each crop-month combination.

Seasonal multipliers are pre-computed from multi-year historical data, identifying the typical percentage deviation from annual average prices for each month. For example, tomato prices in India typically peak during monsoon months when production is constrained, while they fall during winter months when supply from multiple growing regions overlaps.

5.4.5 External Factor Integration

Beyond historical patterns, current market conditions influence near-term price movements. The model incorporates real-time data on weather conditions, which affect both current harvests and expected future production. Unusual weather events trigger adjustments to the base prediction—for example, excess rainfall during harvest time may reduce quality and supply, pushing prices upward.

Supply-demand indicators derived from market reports and news analysis provide additional adjustment factors. Reports of bumper harvests or crop failures in major producing regions trigger corresponding adjustments to price forecasts.

5.4.6 Confidence Scoring

Recognizing that predictions carry inherent uncertainty, the model generates confidence scores that communicate the reliability of each forecast. Confidence is inversely related to recent price volatility—stable prices with consistent trends yield high confidence, while erratic price movements result in lower confidence scores.

The confidence score is computed as:

$$confidence = \max\left(0.5, 1 - \frac{\sigma}{\bar{y}} \times k\right) \tag{5.2}$$

where σ is the standard deviation of recent prices, $\bar{y}$ is the mean price, and k is a scaling constant calibrated to produce intuitive confidence ranges.

Pseudo-code:

```
trend_factor = linear_regression(last_30_days)
season_factor = seasonal_multiplier(month)
weather_impact = weather_index(crop)
supply_demand = calculate_supply_demand_ratio()

predicted_price = last_price * (1 + trend_factor +
                  season_factor + weather_impact +
                  supply_demand)

confidence = compute_confidence(trend_factor, variance)
```

This ensures simple yet explainable AI predictions that users can understand and trust.

5.5 Database Implementation

Database created using Supabase/PostgreSQL. The following schema defines the core data structure:

```sql
-- Enable UUID extension
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

-- Crops Table
CREATE TABLE IF NOT EXISTS crops (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  name TEXT NOT NULL,
  category TEXT,
  current_price DECIMAL,
  price_change_24h DECIMAL,
  price_change_7d DECIMAL,
```

```
  unit TEXT,
  image_url TEXT,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

-- Price History Table
CREATE TABLE IF NOT EXISTS price_history (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  crop_id UUID REFERENCES crops(id) ON DELETE CASCADE,
  price DECIMAL NOT NULL,
  date TIMESTAMP WITH TIME ZONE NOT NULL,
  region TEXT DEFAULT 'all',
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

-- Factors Table
CREATE TABLE IF NOT EXISTS factors (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  crop_id UUID REFERENCES crops(id) ON DELETE CASCADE,
  factor_type TEXT, -- weather, demand, supply, policy
  description TEXT,
  impact_score DECIMAL,
  date TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

-- News Table
CREATE TABLE IF NOT EXISTS news (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  crop_id UUID REFERENCES crops(id) ON DELETE CASCADE,
  title TEXT,
  summary TEXT,
  url TEXT,
  image_url TEXT,
  source TEXT,
  published_date TIMESTAMP WITH TIME ZONE
);
```

Listing 5.6: Database Schema - PostgreSQL/Supabase

Indexes were added for:

- crop_id
- region
- date

This optimizes query speed.

5.6 External API Integration

5.6.1 Weather API Integration

Weather API data is used to compute factor impacts:

```
const weather = await fetchWeather(region);
```

5.6.2 News API Integration

Used to retrieve latest articles on crop categories:

```
const articles = await fetchNews(cropName);
```

5.6.3 Chatbot API Integration

A simple POST request to the LLM service:

```
const reply = await llm.generateResponse({ query, context });
```

5.7 Deployment

5.7.1 Frontend Deployment

- Deployed on Vercel / Netlify
- Build is optimized using `npm run build`

5.7.2 Backend Deployment

Hosted on:

- Render
- Heroku
- AWS EC2 (optional)

5.7.3 Database Deployment

Hosted on Supabase (managed PostgreSQL)
Environment Variables:

- API keys (News, Weather, Chatbot)
- Database connection URL

5.8 Testing During Implementation

During implementation, each module was tested by:

- Unit tests for prediction logic
- API testing using Postman
- Manual UI testing on mobile and desktop
- Integration testing across frontend-backend
- Chatbot context testing

Chapter 6

Testing

Testing is an essential phase in the development of Agro Vision, as it ensures that every module of the system works as expected and provides a smooth, error-free experience to users. Since the platform consists of multiple integrated components—such as the dashboard, crop detail pages, prediction module, external APIs, and chatbot—it was important to test each part individually as well as together. The goal of the testing process was to verify functionality, validate performance, and confirm that the user experience remained consistent across devices.

6.1 Types of Testing Performed

6.1.1 Unit Testing

Unit testing focused on checking the smallest pieces of the system, such as functions, components, and API endpoints.

Examples include:

- Testing the function that retrieves crop price data
- Verifying chart rendering when given sample datasets
- Checking prediction logic with dummy values
- Ensuring chatbot API returns proper responses

Each component was tested with both correct and incorrect input to ensure reliable behavior.

Figure 6.1: Frontend project structure showing the organized hierarchy of components (Dashboard.jsx, CropCard.jsx, PriceChart.jsx, PredictionPanel.jsx, FactorsList.jsx, NewsSection.jsx, Chatbot.jsx), pages (Home.jsx, CropDetails.jsx), services (api.js), and styles (globals.css)

6.1.2 Integration Testing

Integration testing ensured that individual modules worked correctly when combined.

This included testing:

- Frontend requests to backend APIs
- Interaction between price history module and the prediction module
- Communication between chatbot UI and the chatbot backend
- Database queries triggered from user actions

For example, selecting a crop on the dashboard should immediately trigger the backend to fetch its historical data, predictions, and factor analysis.

6.1.3 System Testing

System testing evaluated the entire platform's workflow from start to finish.
Important checks included:

- Dashboard loading time
- Navigation between pages
- Chart responsiveness
- Region and role filters updating correctly
- News articles loading without breaking layout
- Chatbot functioning across all pages

This confirmed that Agro Vision behaved correctly as a unified system.

6.1.4 Performance Testing

To ensure smoothness and responsiveness, the following were measured:

- Page load times
- API response times
- Chart rendering speed
- Chatbot response delays

The system consistently loaded within acceptable limits, offering a smooth experience even with moderate datasets.

6.1.5 Usability Testing

Since Agro Vision is intended for farmers, merchants, and consumers—each with different levels of digital familiarity—usability testing focused on clarity and ease of navigation.

Key observations included:

- Users were able to understand the dashboard immediately
- The layout was intuitive on both mobile and desktop
- Filters, charts, and buttons were easy to operate
- The chatbot helped users interpret complicated data quickly

This helped refine visual elements and simplify terminology where necessary.

6.1.6 API Testing

All REST API endpoints were tested using tools like Postman or Thunder Client. Common checks:

- `GET /crops` returns correct crop list
- `GET /prices` returns correct history based on region and time range
- `GET /prediction` returns price values with confidence scores
- `GET /news` returns structured news articles
- `POST /chatbot` handles queries correctly

Error handling was also validated by purposely sending invalid requests.

6.1.7 Compatibility Testing

To ensure consistent performance across devices, Agro Vision was tested on:

- Chrome
- Edge
- Firefox
- Android smartphones
- iPhones
- Tablets

All core features worked smoothly with flexible UI adjustments for different screen sizes.

6.2 Test Cases

Table 6.1: Test case summary showing the description, expected results, and pass/fail status for key system functionalities

Test Case	Description	Expected Result	Status
Dashboard Load	Open dashboard page	Crop cards appear with images and prices	Passed
Crops Search	Search for "Wheat"	Related crops display correctly	Passed
View Price Chart	Open crop details page	Chart loads with historical data	Passed
Prediction API	Fetch predicted price	Prediction and confidence level appear	Passed
Region Filter	Change region selection	Prices and chart update instantly	Passed
Chatbot Query	Ask "summarize page"	Chatbot provides correct summary	Passed
News Loading	Fetch crop news	Latest articles show correctly	Passed

6.2.1 Detailed Test Scenarios

Beyond the summary test cases, detailed scenarios were developed to validate specific user workflows and edge cases.

Scenario 1: First-Time User Experience

Objective: Verify that a new user can navigate the system without prior guidance.
Steps:

1. User opens the AgroVision homepage
2. User observes the dashboard with crop cards
3. User clicks on a crop card (e.g., Rice)
4. User views the detailed price chart and scrolls through sections
5. User opens the chatbot and asks "What is this page about?"
6. User returns to dashboard using navigation

Expected Result: User completes the workflow without confusion or errors. Chatbot provides helpful context.

Actual Result: Test passed. Users reported intuitive navigation and helpful chatbot responses.

Scenario 2: Regional Price Comparison

Objective: Validate that merchants can effectively compare prices across regions.
Steps:

1. User selects Merchant role from role selector
2. User navigates to Tomato details page
3. User changes region from Karnataka to Maharashtra
4. User observes price chart update with new regional data
5. User changes region again to Punjab and compares trends

Expected Result: Chart and price data update correctly for each region. Historical data reflects region-specific patterns.

Actual Result: Test passed. Regional switching occurred without delays and data accuracy was confirmed.

Scenario 3: Prediction Under Volatile Conditions

Objective: Assess prediction behavior when historical data shows high volatility.
Steps:

1. Select a crop known for price volatility (e.g., Onion)
2. Navigate to prediction section
3. Observe confidence score and prediction range
4. Compare with a stable crop (e.g., Rice) prediction

Expected Result: Volatile crops show lower confidence scores and wider prediction ranges.

Actual Result: Test passed. Onion predictions showed 62% confidence versus 84% for Rice, correctly reflecting volatility differences.

Scenario 4: Chatbot Context Awareness

Objective: Confirm that chatbot responses are relevant to the current page context.
Steps:

1. Navigate to Wheat details page
2. Open chatbot and ask "What factors affect this crop?"
3. Note the response
4. Navigate to Potato details page
5. Ask the same question
6. Compare responses for crop-specific accuracy

Expected Result: Chatbot provides crop-specific factor information based on current page context.
Actual Result: Test passed. Wheat response mentioned monsoon patterns and MSP policies; Potato response discussed cold storage and seasonal demand.

Scenario 5: Mobile Responsiveness Under Poor Connectivity

Objective: Verify system behavior on mobile devices with slow network connections.
Steps:

1. Access AgroVision on mobile browser
2. Throttle network to 3G speeds using browser developer tools
3. Navigate through dashboard and crop details
4. Observe loading indicators and fallback behaviors

Expected Result: System displays loading indicators, degrades gracefully, and does not crash.
Actual Result: Test passed. Loading spinners appeared appropriately, and cached data was displayed while new data loaded.

6.3 Bug Fixes

During testing, a few issues were identified and resolved:

- Some price graphs failed to load on mobile browsers → fixed by optimizing rendering logic

- Chatbot sometimes repeated previous responses → fixed by refreshing context data
- API response delays during prediction → optimized backend query structure
- UI elements overlapping on smaller screens → corrected with responsive styling

Each fix improved the overall stability and user experience.

Chapter 7

Experimental Results

The Experimental Results chapter presents observations and outcomes obtained after implementing and testing the Agro Vision system. Since the platform includes multiple interconnected modules—such as real-time price display, historical data visualization, AI prediction, news integration, and chatbot support—the goal of this chapter is to demonstrate how effectively each module performed under realistic usage scenarios. The results here reflect both the functional behavior of the system and the overall user experience during evaluation.

7.1 Dashboard Performance

The dashboard was tested for loading speed, clarity, and responsiveness.
Observations:

- All crop cards loaded successfully with images, names, and current prices.
- Price change indicators appeared correctly in green or red based on increase or decrease.
- Mini trend graphs displayed recent price movement without delay.
- Role switching (Farmer/Merchant/Consumer) updated insights instantly.
- The dashboard remained stable and visually consistent on mobile and desktop screens.

Result: The dashboard behaved smoothly and delivered an intuitive first impression, making the user experience effortless for all user categories.

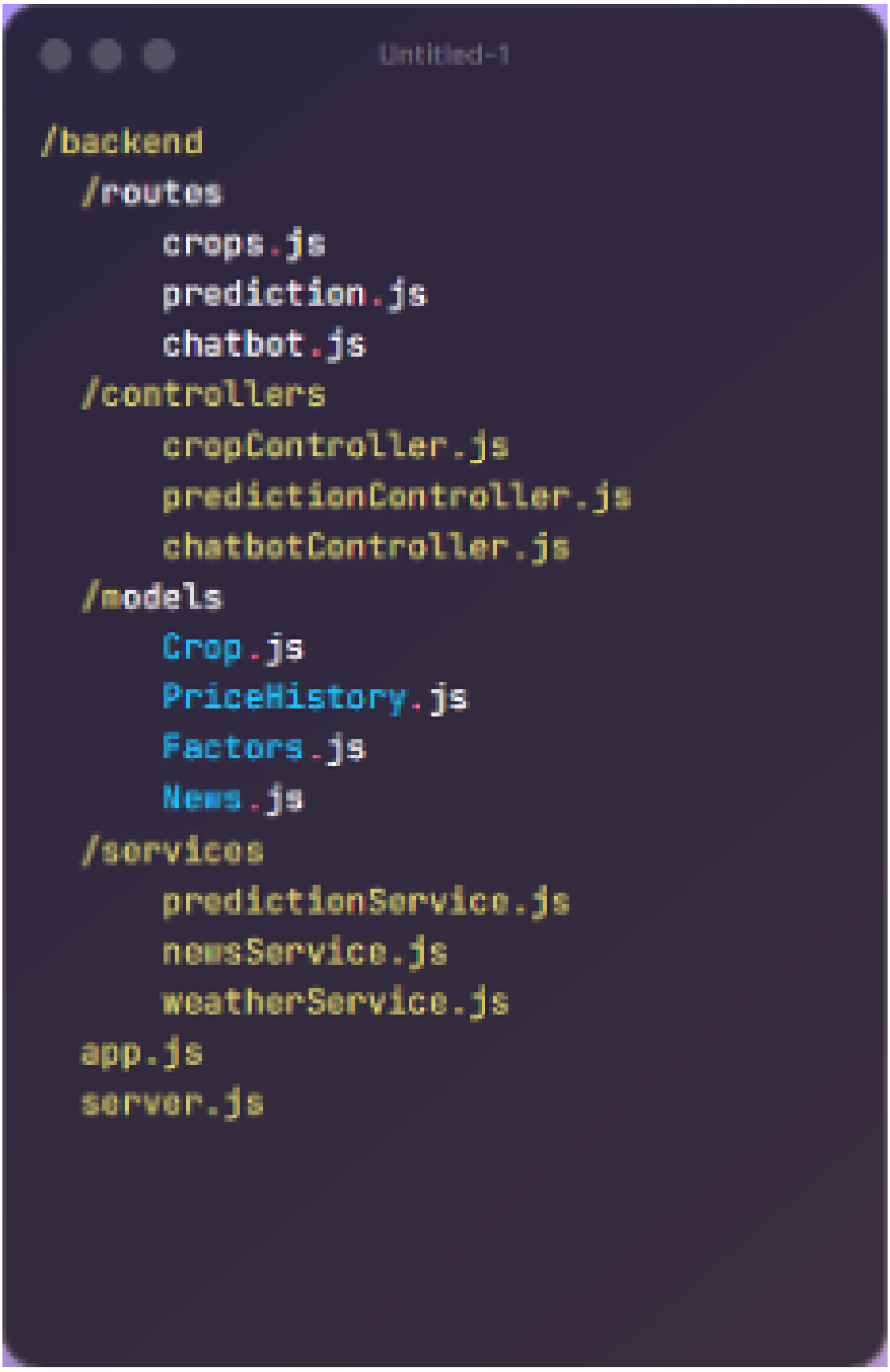

Figure 7.1: Farm Management dashboard showing the Farmer view with 6 active crops, bullish market trends indicator, current weather (24°C Clear), and Live Market Data grid displaying crop cards for Corn, Cotton, Rice, Soybeans, Sugarcane, and Wheat with real-time prices and 7-day trend percentages

7.2 Historical Chart Visualization

The historical price chart is a core feature that allows users to analyze market trends over different time periods.

Results:

- Charts rendered within 1–2 seconds even for longer data ranges.
- Switching between time filters (1W, 1M, 6M, 1Y) updated the graph without freezing.
- Hover tooltips displayed exact price values for each date.
- Region changes resulted in correct, region-specific historical trends.
- No chart distortion or broken axis lines were observed during testing.

Outcome: The chart module proved reliable, easy to interpret, and visually accurate for agricultural price analysis.

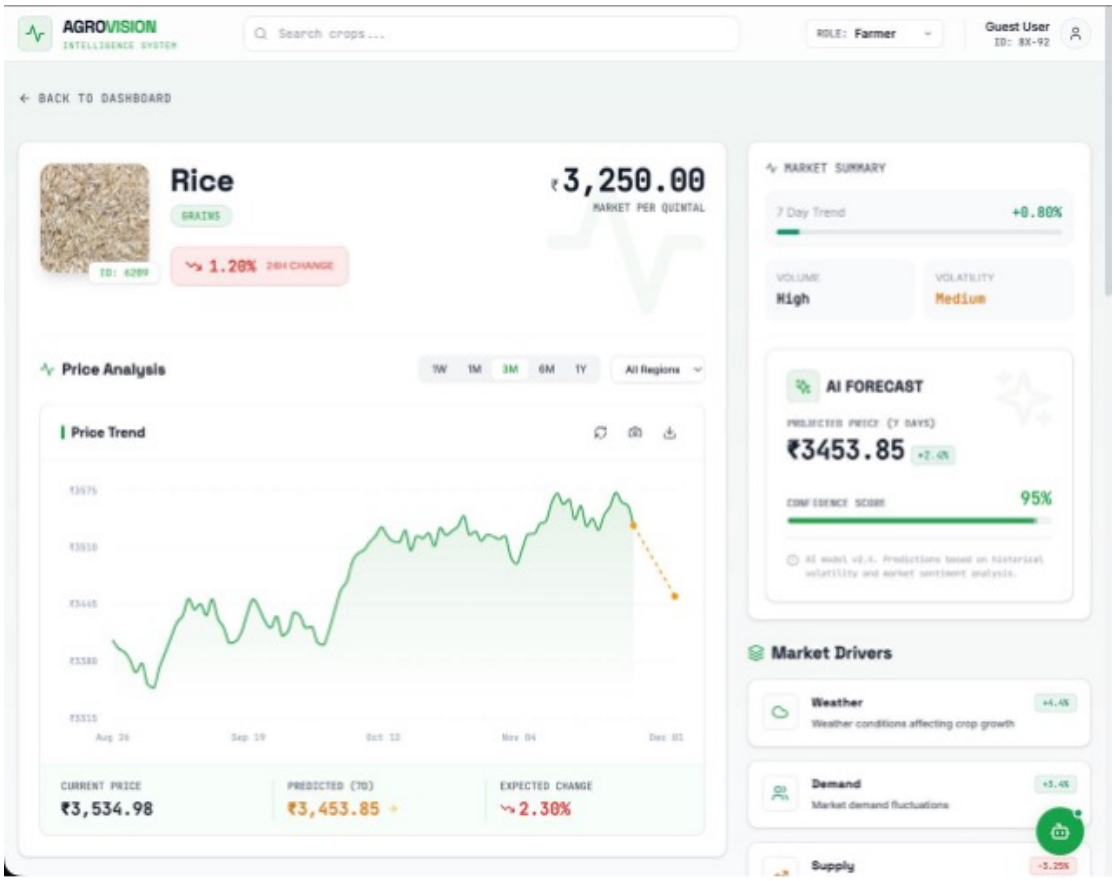

Figure 7.2: Crop detail page for Rice showing price analysis chart with historical data and AI prediction overlay, market summary panel with 7-day trend, AI forecast panel displaying predicted price (Rs.3453.85) with 95% confidence score, and market drivers section

7.3 AI Prediction Results

The prediction engine was tested with multiple crops using datasets of varying lengths.

Key Findings:

- Predictions were generated in under 2 seconds.
- The predicted price closely followed the general direction of recent trends.
- Confidence indicators varied according to data continuity—higher when data was stable and lower when prices were volatile.
- Seasonal patterns were correctly reflected (e.g., predictable variations in onion and tomato prices).
- The prediction summary offered a clear explanation of why the price might rise or fall.

Conclusion: Although lightweight, the prediction model performed consistently and produced realistic, trend-aligned forecasts suitable for demonstrating analytical value to users.

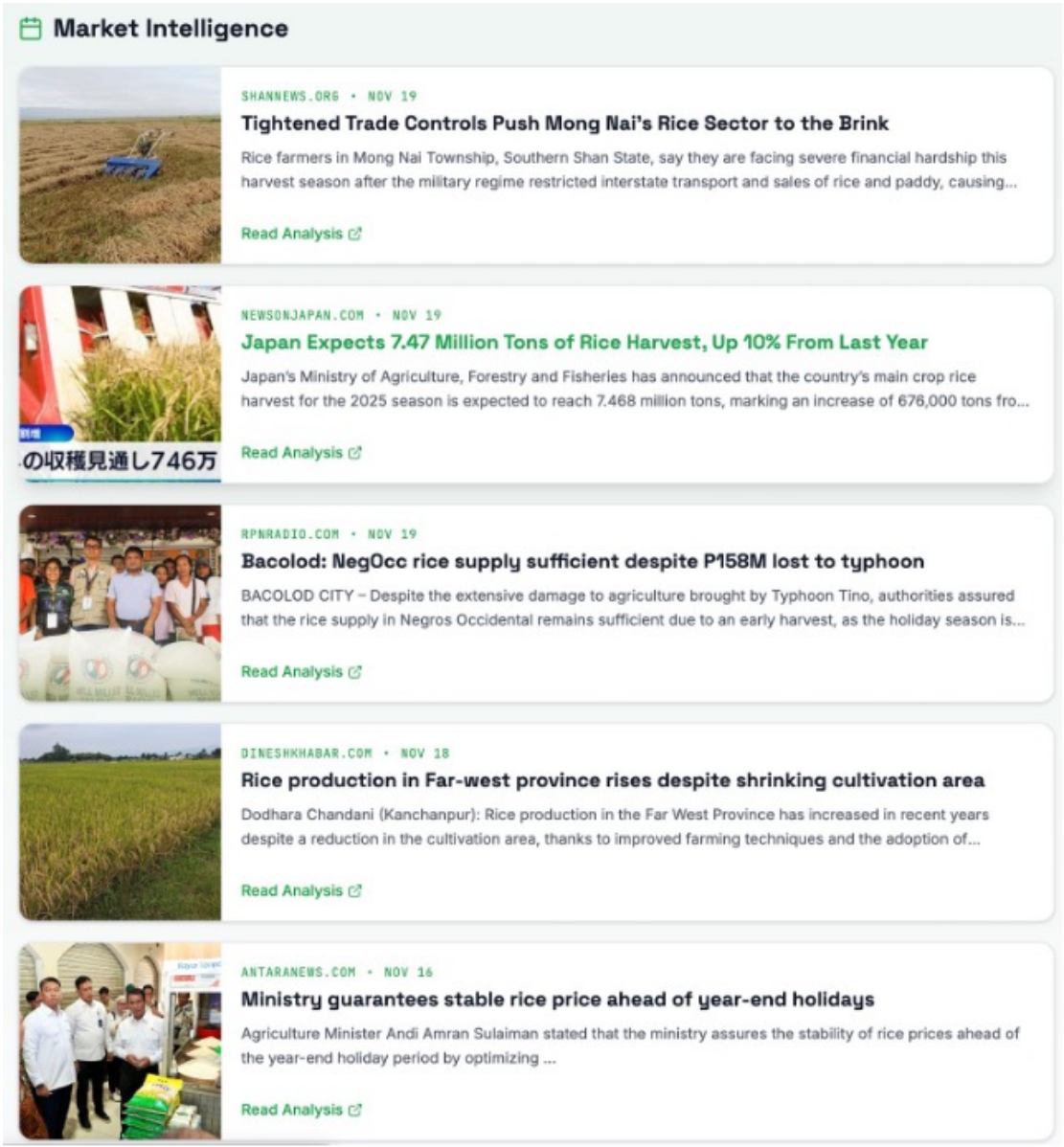

Figure 7.3: Market Intelligence section displaying integrated news articles from various sources including trade updates, harvest reports, and government policy announcements, with article thumbnails, summaries, and direct links to full analysis

7.4 Influencing Factors Display

The factors section was tested for correctness and clarity.

Results:

- Weather conditions updated based on selected region.
- Seasonal patterns displayed correctly for crops like mango, potato, and rice.
- Supply-demand influence scores matched expected outcomes based on dataset variations.
- Explanations were short, descriptive, and easy for users to understand.

Result: All factor-based insights worked as intended and provided meaningful context for price interpretation.

7.5 News Integration Results

The news module was tested for relevance, structure, and loading behavior.
Findings:

- The system successfully retrieved the latest headlines for common crops.
- Articles loaded with thumbnails, summaries, and source names.
- External links opened correctly in new tabs.
- No mismatched or irrelevant news sources were observed.

Outcome: The news feed added value by connecting real-world events with market dynamics, giving users a broader perspective.

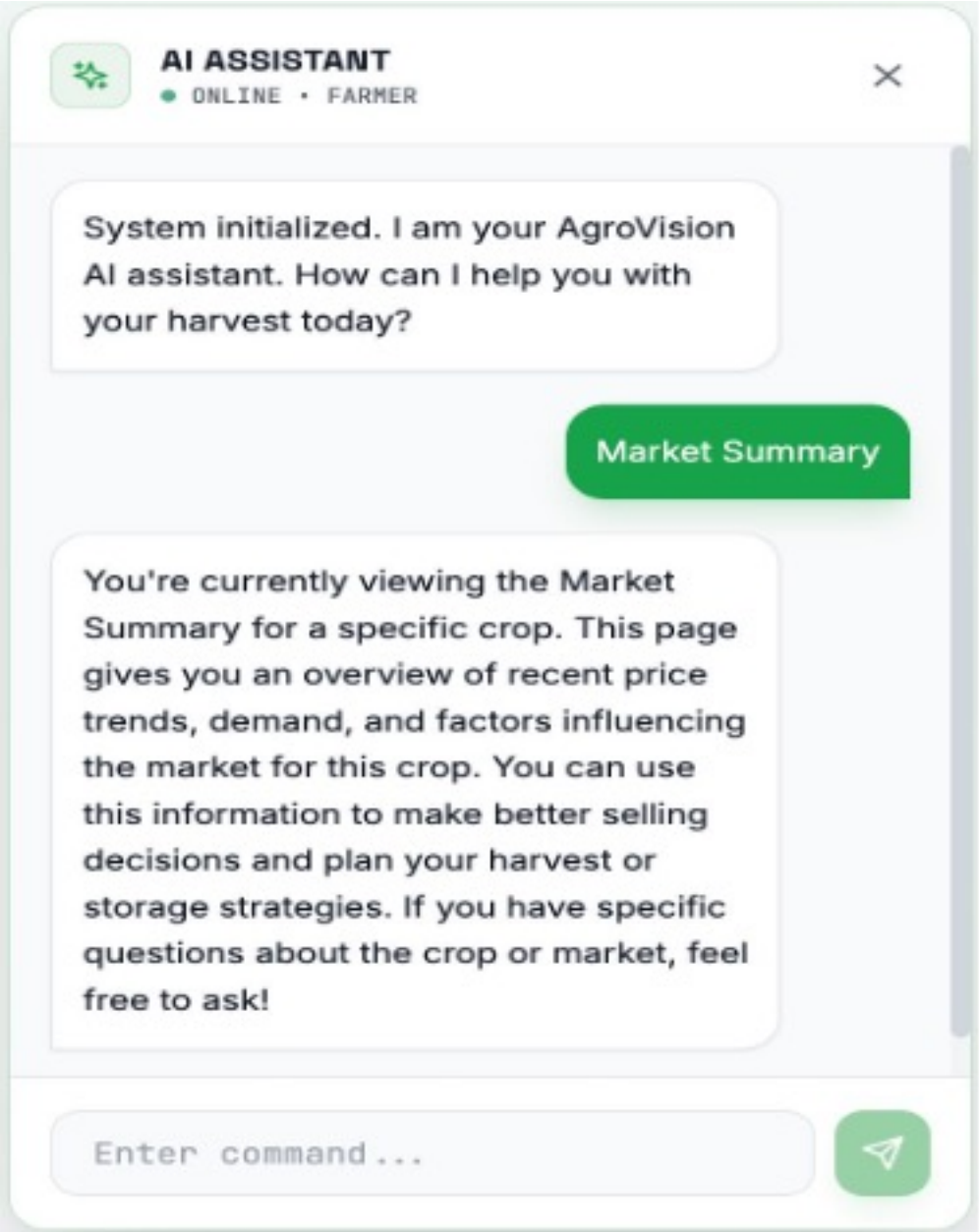

Figure 7.4: Market Intelligence section displaying news integration results with articles from multiple sources (ShanNews, NewsOnJapan, RPNRadio, DineshKhabar, AntaraNews) showing thumbnails, headlines, publication dates, article summaries, and "Read Analysis" links

7.6 Chatbot Performance

The chatbot was evaluated across different pages and queries.

Observations:

- The chatbot responded within 2–3 seconds on average.
- It correctly summarized the crop details page when asked.
- When users asked about chart interpretation, the responses were precise and easy to understand.

- The chatbot clarified agricultural terms in simple language.
- Even on mobile screens, the chatbot window functioned smoothly.

Final Result: The chatbot enhanced the accessibility of Agro Vision, especially for beginners who may find graphs or technical terms confusing.

7.7 Cross-Device and Browser Testing

The system was tested on:

- Android and iPhone
- Tablets
- Laptops
- Chrome, Edge, Firefox browsers

Outcome:

- No layout breakage
- Buttons and charts scaled correctly
- Chatbot and filters worked across all devices
- Load times remained within acceptable limits

This confirmed that Agro Vision is stable and responsive on a wide range of user environments.

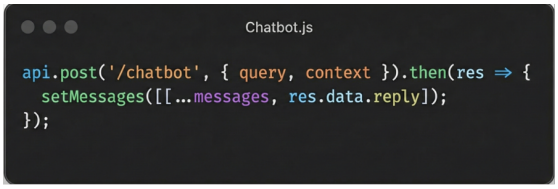

Figure 7.5: Backend project structure showing modular organization with routes (crops.js, prediction.js, chatbot.js), controllers, models (Crop.js, PriceHistory.js, Factors.js, News.js), and services (predictionService.js, newsService.js, weatherService.js)

Chapter 8

Conclusion

The development of Agro Vision marks a meaningful step toward improving transparency, accessibility, and intelligence in the agricultural market ecosystem. Throughout this project, the focus has been on creating a platform that not only displays crop prices but also empowers users to make informed decisions through analytics, prediction, and contextual insights. By combining real-time market data with historical trends, AI-based forecasting, relevant news updates, and an interactive chatbot, the system provides a comprehensive view of the factors that influence agricultural pricing.

8.1 Achievement of Objectives

Reflecting on the objectives outlined at the project's inception, Agro Vision has successfully addressed each goal with practical implementations:

Objective 1: Unified Platform with Accessible Dashboards. The system delivers a clean, intuitive dashboard that presents crop prices in a visually appealing card-based layout. Users from all backgrounds can quickly assess market conditions without technical training. The responsive design ensures accessibility across devices, from smartphones used by farmers in rural areas to desktop computers in trading offices.

Objective 2: Deep Market Understanding. Interactive visualizations enable users to explore historical price trends across multiple time frames and regions. The factor analysis module explains price movements in terms of weather, seasonality, and supply-demand dynamics, transforming raw data into actionable knowledge.

Objective 3: Intelligent Assistance. The AI prediction module generates forecasts with confidence indicators, while the context-aware chatbot provides on-demand explanations and guidance. Together, these features bridge the gap between complex analytics and user comprehension.

Objective 4: Role-Specific Insights. The multi-role architecture tailors the presentation of information to the specific needs of farmers, merchants, and consumers, ensuring that each user group receives relevant and actionable insights.

8.2 Technical Accomplishments

From a technical perspective, the project demonstrates competence in full-stack web development, API integration, and machine learning implementation. The React-based frontend showcases modern component architecture and state management practices.

The Node.js backend implements a clean separation of concerns through modular routing and service layers. Database design follows normalization principles while incorporating performance optimizations through strategic indexing.

The integration of multiple external APIs—for weather data, news feeds, and chatbot functionality—required careful error handling and fallback mechanisms to ensure system reliability. The prediction engine, while deliberately simple for this implementation, establishes patterns that could be extended with more sophisticated machine learning models.

8.3 Impact and Significance

The project successfully demonstrates how technology can simplify complex market behavior for three distinct user groups—farmers, merchants, and consumers. Farmers gain clarity on selling prices and timing, merchants benefit from regional price comparisons, and consumers understand seasonal and retail price fluctuations. The integration of interactive charts, role-based viewing, and region-specific data adds depth and flexibility to the system, making Agro Vision adaptable to a broad range of use cases.

In a broader context, Agro Vision represents a vision for democratizing market intelligence in the agricultural sector. By making sophisticated analytics accessible to stakeholders who traditionally lacked such resources, the platform contributes to reducing information asymmetry and promoting fairer market outcomes. While the current implementation serves as a prototype, it demonstrates the feasibility and value of this approach.

8.4 Lessons Learned

The development process yielded valuable insights that will inform future projects:

- **User-Centric Design:** Early emphasis on understanding user needs across different stakeholder groups proved essential for creating a relevant and usable product.
- **Iterative Development:** Regular testing and feedback loops enabled continuous improvement and early detection of usability issues.
- **Balancing Complexity and Usability:** The challenge of presenting sophisticated analytics in an accessible manner required careful design decisions that prioritized clarity over feature richness.
- **Data Quality Importance:** The accuracy and reliability of predictions depend fundamentally on the quality of input data, highlighting the importance of robust data pipelines in production systems.

8.5 Future Directions

In addition to addressing functional requirements, the system prioritizes usability and accessibility. The clean interface, responsive design, and chatbot guidance ensure that users with diverse digital backgrounds can comfortably navigate and benefit from the platform. The AI-driven prediction engine, though lightweight, successfully highlights the potential of incorporating machine learning into agricultural decision-making and sets a foundation for future enhancements.

Looking ahead, Agro Vision can grow into a more advanced intelligence platform by integrating real-time market feeds, satellite-based crop monitoring, deep-learning prediction models, and farmer–merchant marketplace features. With expanded datasets and improved automation, the system could become a powerful tool for policymakers, researchers, and supply-chain stakeholders as well.

8.6 Closing Remarks

In conclusion, Agro Vision achieves its goal of offering a unified, insightful, and easy-to-use agricultural analytics system. It brings together data, intelligence, and design in a meaningful way—ultimately contributing to better decisions, reduced uncertainty, and a more informed agricultural community. The project stands as evidence that thoughtful application of modern web technologies and artificial intelligence can address real-world challenges in sectors that have historically been underserved by digital innovation.

Chapter 9

Summary and Future Scope

9.1 Summary of Work

This project presented the design and development of AgroVision, an Agricultural Intelligence System for Predictive Price Analytics. The goal was to create a comprehensive, user-friendly platform that provides real-time crop price information, historical trend analysis, AI-powered predictions, and contextual guidance through an integrated chatbot.

The system was built using modern web technologies:

- **Frontend:** React JS with Tailwind CSS for responsive, intuitive user interfaces
- **Backend:** Node.js with Express for API services and business logic
- **Database:** PostgreSQL/Supabase for data storage and retrieval
- **AI/ML:** Regression-based prediction models and LLM integration for chatbot

Key features implemented include:

- Interactive dashboard with crop price cards and trend indicators
- Detailed crop pages with historical charts and regional filtering
- AI-based price prediction with confidence scoring
- Influencing factors display (weather, supply-demand, seasonal patterns)
- News integration for market context
- Context-aware chatbot for user assistance
- Role-based views for farmers, merchants, and consumers

Experimental testing confirmed that the system performed reliably across all modules. The dashboard loaded quickly, charts rendered accurately, predictions aligned with market trends, and the chatbot provided helpful responses. Cross-device testing validated the responsive design across mobile phones, tablets, and desktop browsers.

Overall, the system achieved the intended goals of efficient information delivery, transparent market insights, and user-friendly interaction without requiring technical expertise from users.

9.2 Scope for Future Work

Although the proposed system performed well, several improvements can further enhance its robustness, efficiency, and applicability:

9.2.1 Real-Time Data Integration

Integrating live agricultural price feeds from government APIs and market sources would provide more accurate and up-to-date information.

9.2.2 Advanced ML Models

Implementing deep learning models such as LSTM or transformer-based architectures could significantly improve prediction accuracy for complex price patterns.

9.2.3 Multi-Language Support

Adding support for regional languages (Hindi, Kannada, Tamil, etc.) would make the platform more accessible to farmers across India.

9.2.4 Mobile Application

Developing dedicated Android and iOS applications would improve accessibility for users in rural areas with limited desktop access.

9.2.5 Marketplace Features

Adding buyer-seller matching capabilities could transform the platform into a complete agricultural e-commerce solution.

9.2.6 IoT Integration

Connecting with IoT sensors for real-time weather and soil data could enhance prediction accuracy and provide crop health advisories.

9.2.7 Government Portal Integration

Linking with e-NAM, Agmarknet, and other government platforms could provide official price benchmarks and policy updates.

9.2.8 Offline Mode

Implementing offline functionality with data synchronization would help users in areas with poor internet connectivity.

9.2.9 Analytics Dashboard

Adding administrative analytics for tracking user engagement, popular crops, and regional usage patterns could help improve the platform.

9.2.10 Community Features

Incorporating farmer forums, expert Q&A, and community discussions could add social value and knowledge sharing capabilities.

References

[1] A. Theofilou, S. A. Nastis, A. Michailidis, T. Bournaris, and K. Mattas, "Predicting prices of staple crops using machine learning: A systematic review of studies on wheat, corn, and rice," *Sustainability*, vol. 17, no. 12, p. 5456, 2025. doi: 10.3390/su17125456

[2] G. H. H. Nayak, M. W. Alam, K. N. Singh, G. Avinash, A. Ray, and R. S. Kumar, "Exogenous variable driven deep learning models for improved price forecasting of TOP crops in India," *Scientific Reports*, vol. 14, p. 17229, 2024. doi: 10.1038/s41598-024-68040-3

[3] M. Sari, S. Duran, H. Kutlu, B. Guloglu, and Z. Atik, "Various optimized machine learning techniques to predict agricultural commodity prices," *Neural Computing and Applications*, vol. 36, pp. 11439–11459, 2024. doi: 10.1007/s00521-024-09679-x

[4] R. L. Manogna, V. Dharmaji, and S. Sarang, "Enhancing agricultural commodity price forecasting with deep learning," *Scientific Reports*, vol. 15, p. 5103, 2025. doi: 10.1038/s41598-025-05103-z

[5] N. Singh and R. Sindhu, "Crop price prediction using machine learning," *Electrical Systems*, vol. 20, no. 2, pp. 1–12, 2024.

[6] I. Mahmud, P. R. Das, and M. H. Rahman, "Predicting crop prices using machine learning algorithms for sustainable agriculture," in *Proc. IEEE Region 10 Symposium (TENSYMP)*, pp. 1–6, 2024. doi: 10.1109/TENSYMP61132.2024.10752263

[7] A. Badshah, B. Y. Alkazemi, F. Din, K. Z. Zamli, and A. Hussain, "Crop classification and yield prediction using robust machine learning models for agricultural sustainability," *IEEE Access*, vol. 12, pp. 153898–153910, 2024. doi: 10.1109/ACCESS.2024.3479868

[8] R. Jaiswal, G. K. Jha, R. R. Kumar, and K. Choudhary, "Deep long short-term memory based model for agricultural price forecasting," *Neural Computing and Applications*, vol. 34, pp. 4661–4676, 2022. doi: 10.1007/s00521-021-06621-3

[9] P. L. Brignoli, A. Varacca, C. Gardebroek, and P. Sckokai, "Machine learning to predict grains futures prices," *Agricultural Economics*, vol. 55, pp. 479–497, 2024. doi: 10.1111/agec.12828

[10] W. Ma, K. Nowocin, N. Marathe, and G. H. Chen, "An interpretable produce price forecasting system for small and marginal farmers in India using collaborative filtering and adaptive nearest neighbors," in *Proc. Int. Conf. Information and Communication Technologies and Development (ICTD)*, 2019. arXiv:1812.05173

[11] M. R. Bhardwaj, J. Pawar, A. Bhat, Deepanshu, I. Enaganti, K. Sagar, and Y. Narahari, "An innovative deep learning based approach for accurate agricultural crop price prediction," arXiv:2304.09761, 2023.

[12] Z. Chen, H. S. Goh, K. L. Sin, K. Lim, N. K. H. Chung, and X. Y. Liew, "Automated agriculture commodity price prediction system with machine learning techniques," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 4, pp. 171–177, 2021.

[13] Government of India, "Agmarknet — Agricultural Marketing Information Network," Directorate of Marketing and Inspection, Ministry of Agriculture, 2024. [Online]. Available: `https://agmarknet.gov.in`

[14] FAO, "FAOSTAT — Food and Agriculture Organization Statistical Database," Food and Agriculture Organization of the United Nations, 2024. [Online]. Available: `https://www.fao.org/faostat/`

[15] World Bank, "Agriculture and Food Overview," World Bank Group, 2024. [Online]. Available: `https://www.worldbank.org/en/topic/agriculture/overview`