INTEGRATED FARM MANAGEMENT AND MARKET PLACE SYSTEM

A PROJECT REPORT

Submitted by

ASHIK V M(SNG22CS050)

to

The APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

Department of Computer Science and Engineering

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 682311

APRIL 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING,
KADAYIRUPPU, 682311

(Affiliated to APJ Abdul Kalam Technological University & Approved by A.I.C.T.E)

CERTIFICATE

This is to certify that the project report,**"INTEGRATED FARM MANAGEMENT AND MARKET PLACE SYSTEM"** submitted by **ASHIK V M** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision . This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. (Dr.) Smitha Suresh	**Asst. Prof. Chinchu Paulose**	**Assoc. Prof. Saini Jacob Soman**
HEAD OF THE DEPARTMENT	**PROJECT GUIDE**	**PROJECT COORDINATOR**

Submitted for the University Evaluation on ……………………………………….

University Register No…………………………………………………………….

Internal Examiner External Examiner

DECLARATION

We undersigned hereby declare that the project report "**Integrated Farm Management and Market Place System**", submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Assoc. Prof. Saini Jacob Soman**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu Ashik V M

Date: 3/04/2025

ACKNOWLEDGEMENT

We sincerely express our gratitude to Sree Narayana Gurukulam College of Engineering, Ernakulam, for providing us with the opportunity and resources to undertake this project, Farmease – An Integrated Farm Management and Marketplace System. We extend our heartfelt thanks to **Prof. (Dr). Smitha Suresh**, Head of the Computer Science and Engineering Department, for her continuous encouragement and support. We are also deeply grateful to our project guide**, Asst. Prof. Chinchu Paulose**, for her valuable guidance, technical insights, and unwavering support, which played a crucial role in shaping our project. Additionally, we extend our sincere thanks to our Project Coordinators, **Asst. Prof. Archana P. S** and **Assoc. Prof. Saini Jacob Soman**, for their valuable advice and assistance throughout our project. We also appreciate the faculty members of the CSE Department for their support and motivation. Lastly, we express our gratitude to our families and friends for their encouragement, which helped us successfully complete this project.

Ashik V M

COURSE OUTCOME AND PROGRAM OUTCOME

Course Outcome	
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools &advanced programming techniques (Cognitive Knowledge Level: Apply)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Program outcomes	
Engineering Graduates will be able to:	
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO’s)	
PSO1	Shall enhance the employability skills by finding innovative solutions for challenges and problems in various domains of CS.
PSO2	Shall apply the acquired knowledge to develop software solutions and innovative mobile applications for various problems.

CO PO PSO MAPPING

	PO1 (Engineering Knowledge)	PO2 (Problem Analysis)	PO3(Design/Development of Solution)	PO4 (Investigation of complex problem)	PO5 (Modern tool usage)	PO6 (The Engineer and Society)	PO7 (Environment and Sustainability)	PO8 (Ethics)	PO9 (Individual and team work)	PO10 (Communication)	PO11(Management and Finance)	PO12 (Life long learning)	PSO1 finding innovative solution	PSO2 Software envelopment
CO1	2	2	2	2		2	2	2	2	2	2	2	2	2
CO2	2	2	2	2	2	2		2	2	2	2	2	2	2
CO3	2	3	3	3	3	2	2	2	3	3	2	2	3	3
CO4	2	2	2	2	2			2	2	2	2	2		2
CO5	3	3	3	3	3	2	2	2	2		3	3	3	3
AVERAGE	2.2	2.4	2.4	2.4	2.5	2.0	2.0	2.0	2.2	2.25	2.2	2.2	2.5	2.4

PO PSO Attainment

PO	Attained Point (0/1/2/3)	Justification
PO1	3	Applied data science and machine learning models for plant disease detection and weather prediction.
PO2	3	Identified challenges in farming, such as disease spread and market price fluctuations, and proposed data-driven solutions.
PO3	3	Designed and implemented an interactive web-based system for farmers to monitor and manage agricultural activities.
PO4	2	Analyzed weather patterns and plant disease symptoms using machine learning models.
PO5	3	Utilized Flask, TensorFlow, and APIs to integrate real-time disease detection and market price updates.
PO6	2	Addressed agricultural challenges faced by farmers through a digital platform that improves productivity.
PO7	2	Encouraged sustainable farming by providing predictive insights on weather and disease prevention.
PO8	3	Maintained secure and ethical handling of farmer data, ensuring privacy and accuracy of information.
PO9	3	Collaborated effectively to build a multi-feature platform, integrating various modules such as disease detection and e-commerce.
PO10	3	Documented project workflow, conducted user training for farmers, and provided a simple and intuitive UI.
PO11	2	Managed project resources, including cloud hosting, database optimization, and API integration.
PO12	3	Researched advanced agricultural technologies and continuously improved the system with user feedback.
PSO1	3	Enhanced problem-solving and technical skills by developing a scalable and deployable farm management system.
PSO2	3	Developed a user-friendly web application integrating based on plant disease detection and real time weather updates.

ABSTRACT

FarmEase is an Integrated Farm Management and Marketplace System designed to empower farmers by bridging the gap between traditional farming methods and modern digital solutions. The platform provides essential services such as direct product selling, real-time market price updates, weather forecasts, agricultural tool rentals, plant disease detection using machine learning, and personalized fertilizer and pesticide recommendations. By leveraging advanced technologies, FarmEase enables farmers to enhance their decision-making, reduce operational costs, and improve crop yield. The plant disease detection system utilizes image recognition and annotated datasets to automatically identify and diagnose diseases, helping farmers take timely preventive measures. Additionally, real-time weather updates and data-driven farming insights assist in planning cultivation and harvesting activities more efficiently. By integrating multiple farming-related services into a single platform, FarmEase promotes efficiency, sustainability, and economic growth, ensuring that farmers are better equipped to face modern agricultural challenges.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
IoT	Internet of Things
ADMA	Agriculture Data Management and Analytics
FAIR	Findable Accessible Interoperable Reusable
PCA	Principle Component Analysis
ExG	Excess Green
ResNet	Residual Network
YOLOv5	You Only Look Once version 5
GAN	Generative Adversarial Network
GB	Giga Byte
RAM	Random Access Memory
OS	Operating System
DBMS	DataBase Management System
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
ER	Entity- Relationship
TC	Test Case
API	Application Programming Interface
IEEE	Institute of Electrical and Electronics Engineers

CHAPTER 1

INTRODUCTION

Agriculture is the backbone of many economies, but farmers face numerous challenges such as market exploitation, unpredictable weather, plant diseases, and limited access to modern tools. Traditional farming methods often result in low productivity and financial instability, making it essential to adopt technology-driven solutions. FarmEase is an Integrated Farm Management and Marketplace System designed to bridge this gap by providing farmers with an all-in-one digital platform. It offers a marketplace for buying and selling agricultural products, real-time weather forecasts to prevent crop loss, machine learning-based plant disease detection, and a tool rental service to improve accessibility to modern farming equipment. Additionally, it provides fertilizer and pesticide recommendations, ensuring optimal use of agricultural resources while maintaining soil health. By integrating AI-driven insights, automation, and user-friendly interfaces, the platform empowers farmers to make informed decisions, maximize productivity, and reduce operational challenges.

By eliminating middlemen, reducing crop loss through early disease detection, and providing financial relief through rental services, FarmEase enhances the overall farming experience. The platform is built to be affordable, easy to use, and scalable, making it a valuable asset for both small and large-scale farmers. With its integrated approach to farm management, FarmEase not only simplifies agricultural operations but also promotes sustainable farming practices. By leveraging modern technology, it aims to revolutionize agriculture, improve farmer incomes, and contribute to food security and economic growth.

The primary objective of FarmEase is to enhance agricultural efficiency, productivity, and sustainability through an integrated digital platform. Farmers often struggle with market access, unpredictable weather, plant diseases, and expensive farming equipment, leading to financial losses. FarmEase aims to provide a comprehensive solution by offering features such as a direct marketplace for buying and selling agricultural products, AI-driven plant disease detection, real-time weather forecasting, and a tool rental service.

Another key objective of FarmEase is to bridge the gap between traditional farming practices and modern technology. The platform ensures that farmers, even with minimal digital literacy, can easily access its benefits through a user-friendly interface. It also promotes sustainable agriculture by providing fertilizer and pesticide recommendations, ensuring proper soil health management. By integrating multiple farming solutions into a single system, FarmEase aims to empower farmers, reduce operational costs, and contribute to the overall growth of the agricultural sector.

FarmEase is designed to modernize and simplify agricultural management by providing farmers with a digital platform that integrates e-commerce, AI-driven insights, and real-time data analytics. The system enables direct buying and selling of agricultural products, eliminating the need for middlemen and ensuring fair pricing. With real-time weather forecasting, farmers can make informed decisions about irrigation, planting, and harvesting, reducing the risk of crop loss due to unpredictable climatic conditions. The AI-based plant disease detection system further enhances farm productivity by allowing farmers to identify diseases early and take preventive measures, ultimately improving crop yield and quality.

Beyond individual farm management, FarmEase has a broader scope in revolutionizing the agricultural sector. The tool rental service ensures that even small-scale farmers can access expensive equipment at an affordable cost, making modern farming more inclusive. Additionally, the fertilizer and pesticide recommendation system promotes efficient resource utilization and sustainable farming practices. As the platform evolves, it has the potential to incorporate government schemes, smart farming IoT solutions, and AI-powered analytics, further transforming agriculture into a more efficient, technology-driven, and profitable industry.

CHAPTER 2
LITERATURE SURVEY

[1] Agriculture is rapidly evolving with the adoption of intelligent data management systems that address key challenges such as data fragmentation, lack of interoperability, and inefficient processing of large datasets. A unified platform, ADMA (Agriculture Data Management and Analytics), enables better organization and accessibility of agricultural data by following FAIR principles—making data Findable, Accessible, Interoperable, and Reusable. It uses semantic data management, interactive portals, and high-performance computing to provide real-time insights and support informed decision-making. This leads to improved resource allocation, enhanced research capabilities, and scalable solutions suitable for various farming environments.

The integration of artificial intelligence and machine learning within the platform allows for predictive analytics, early disease detection through image recognition, and automated decision support for farming activities. Real-time data processing enables instant weather updates, sensor-based soil and crop monitoring, dynamic market insights, and livestock health tracking. These technologies enhance productivity and responsiveness across the agricultural value chain. Future developments aim to refine AI models, implement blockchain for secure transactions, and expand accessibility to small-scale farmers, paving the way for a more efficient, data-driven, and sustainable agricultural sector.

[2] The web-based agricultural monitoring and sales management system aims to modernize traditional farming practices by addressing key challenges such as inefficient monitoring, poor sales management, and limited market access. By leveraging technologies like IoT, cloud computing, and AI, the system allows for real-time data collection and storage, enabling farmers to track farm activities, monitor crop conditions, and manage sales through a user-friendly online dashboard. The system also

facilitates direct communication between farmers, buyers, and suppliers, reducing reliance on intermediaries and enhancing coordination across the supply chain. Additionally, the integration of automated sales tracking and AI-powered decision support tools empowers farmers to make smarter, data-driven decisions about crop health and pricing strategies.

This technology-driven approach brings significant advantages to the agricultural sector, including improved efficiency, better market accessibility, reduced operational costs, and enhanced sustainability through data-backed practices. The system promotes transparency in trade and offers scalability to accommodate various farm sizes and operations. Future developments, such as integrating mobile applications, advanced AI models, and blockchain technology, are expected to further enhance the system's capabilities. These advancements will make the platform more accessible and secure while supporting precision farming, predictive analytics, and traceability in the agricultural supply chain, leading to a more connected, profitable, and sustainable agriculture ecosystem.

[3] The IoT-based Smart Agriculture Monitoring System with Predictive Analysis introduces a technologically advanced approach to farming by combining real-time environmental monitoring with predictive analytics. The system employs IoT sensors to track critical parameters like soil moisture, temperature, and humidity, and transmits this data to a cloud-based platform. Using machine learning techniques such as Principal Component Analysis (PCA) and neural networks, the system generates predictions and actionable insights to support informed decision-making. Through a mobile and web interface, farmers can access recommendations on irrigation, fertilization, and crop management, thereby improving operational efficiency and planning.

This smart system brings numerous advantages including precise farming, optimized resource usage, and increased crop yield. Automated alerts help reduce water and fertilizer wastage while promoting eco-friendly practices that support sustainability. The scalability of the system makes it applicable to various crop types and agricultural regions. Looking ahead, future developments could include advanced AI models for even more accurate predictions, blockchain integration for supply chain traceability, and AI-based pest detection for proactive crop protection. These enhancements further strengthen the role of IoT and analytics in revolutionizing modern agriculture.

[4] The study on Agriculture Digitalization provides a global perspective on how digital technologies are revolutionizing farming through a detailed bibliometric analysis. It explores the growing integration of tools like IoT, AI, big data, and blockchain in agriculture, which collectively contribute to smarter, more efficient, and sustainable farming practices. Using bibliometric methods, the research analyzes global scientific literature to uncover key themes, research trends, and collaborative efforts in the field of digital agriculture.

The analysis reveals the increasing adoption of digital solutions such as sensor-based monitoring, AI-driven analytics, and blockchain-enabled traceability. These technologies help address challenges like inefficient resource use, climate variability, and limited market access by improving data-driven decision-making and promoting transparency across the agricultural supply chain.

Notably, digital agriculture offers major benefits including better insights through big data, optimized resource management, enhanced collaboration among stakeholders, and improved access to markets. The study emphasizes that ongoing innovation—especially through AI, satellite and sensor integration, and policy support—is essential for scaling digital solutions across diverse regions and crop systems.

In the future, expanded data sources, advanced predictive tools, and strategic policies will play a key role in increasing adoption and effectiveness. As highlighted, agriculture digitalization holds immense potential to tackle critical issues such as food security and sustainability on a global scale.

[5] This paper explores how integrating Artificial Intelligence (AI) and the Internet of Things (IoT) can transform traditional farming into a smart, sustainable system. Current agricultural practices face issues like climate change, resource misuse, and labor shortages. AI and IoT help address these by providing real-time monitoring, predictive analytics, and automation, improving efficiency and reducing environmental impact.

IoT sensors collect data on soil, climate, and pests, which is analyzed using AI models like PCA and neural networks. These models offer timely insights on irrigation, pest control, and crop health, helping farmers make informed decisions through mobile apps or automated systems.

The system optimizes resource use, reduces labor dependency, and improves crop management. It promotes higher yields and sustainability. Future enhancements may include AI for climate adaptation, blockchain for data transparency, and affordable solutions for small farmers.

In conclusion, AI and IoT hold great promise for the future of agriculture by boosting productivity, lowering costs, and supporting environmental sustainability, though challenges like cost and data security remain.

[6] The IoT-Based Smart Agriculture Monitoring System explores how integrating IoT technology with predictive analytics can enhance farming by providing real-time insights into environmental conditions. Traditional farming lacks timely data on factors like soil moisture and temperature, leading to inefficiencies. This system addresses those gaps by using sensors and AI models to monitor, analyze, and optimize crop growth.

IoT devices collect data from the field, which is then processed using techniques like Principal Component Analysis and neural networks. These tools help identify patterns and predict issues like drought or pest outbreaks. The system offers farmers actionable recommendations through a user-friendly interface, enabling better irrigation, fertilization, and pest control.

Key benefits include real-time monitoring, improved resource use, and early threat detection, which lead to higher productivity and sustainability. In the future, integrating advanced AI models and making the system adaptable to various crops and regions could further enhance its impact. Overall, this IoT-based system represents a promising step toward smarter, more efficient, and sustainable farming practices.

[7] The paper Remote Sensing for Agriculture in the Era of Industry 5.0 explores how modern technologies like satellites, drones, and IoT devices are transforming farming. As agriculture evolves with Industry 5.0, remote sensing enables real-time monitoring of crops, soil, and environmental conditions, helping farmers make data-driven decisions that enhance productivity and sustainability.

Traditional farming faces challenges like inefficient resource use, lack of real-time data, and delayed detection of issues like crop diseases. Remote sensing addresses these problems by providing accurate data through high-resolution imagery and smart sensors. This data is analyzed using AI and machine learning to forecast yields, identify risks, and optimize farming practices.

The system enhances precision and sustainability by allowing targeted use of water, fertilizers, and pesticides, reducing waste and environmental harm. It also empowers farmers with insights to respond proactively to climate variations and field conditions.

Looking ahead, deeper AI integration, satellite-based global monitoring, and advanced data analytics promise even more effective tools for precision agriculture. Although high costs and technical barriers exist, continued innovation and support can make remote sensing a cornerstone of sustainable, smart farming worldwide.

[8] The paper “A Deep Learning Image Augmentation Method for Field Agriculture" addresses the challenge of limited labeled datasets in agricultural AI applications. Deep learning models, used for tasks like plant disease detection and crop classification,

require large and diverse image datasets, but manual labeling is time-consuming and expensive.

To overcome this, the study proposes an image augmentation method that generates synthetic images using techniques like segmentation with the Excess Green (ExG) index. These synthetic images are added to existing datasets, enhancing diversity and reducing the need for manual annotation. The augmented datasets are then used to train models like ResNet, YOLOv5, and DeepLabV3, resulting in improved performance and accuracy.

The method offers key benefits, including reduced labeling efforts, better model performance, and cost-effective dataset generation. Looking ahead, integrating advanced AI techniques like GANs and enabling real-time data generation could further enhance this approach. Overall, this technique supports more scalable, accurate, and accessible deep learning applications in agriculture.

CHAPTER 3

PROBLEM STATEMENT AND OBJECTIVE

3.1 PROBLEM STATEMENT

Agriculture plays a crucial role in ensuring food security and economic stability. However, farmers face numerous challenges in managing their resources, predicting environmental conditions, and optimizing agricultural productivity. Traditional farming methods often rely on experience-based decision-making, which lacks precision and can lead to inefficiencies in farm management. Additionally, many farmers rely on fragmented software solutions that do not provide integrated, real-time data analysis. This results in poor decision-making, increased operational costs, reduced crop yields, and unsustainable agricultural practices.

One of the key challenges farmers face is inefficient resource management. Water, fertilizers, and pesticides are often overused or underutilized due to a lack of precise data on soil health, crop needs, and climatic conditions. This not only increases production costs but also contributes to environmental degradation.

Weather forecasting and climate monitoring are also significant challenges. Farmers need accurate and real-time weather predictions to plan their agricultural activities effectively. Unpredictable weather conditions, such as unexpected rainfall, droughts, or temperature fluctuations, can severely impact crop growth and yield. The lack of advanced forecasting tools makes it difficult for farmers to adapt to changing environmental conditions, leading to crop losses.

Another major issue is market unpredictability. Farmers often struggle to gain real-time insights into market trends, consumer demand, and price fluctuations. Without access to accurate market insights, they may end up selling their produce at low prices or face losses due to excess supply. The absence of an intelligent pricing prediction system prevents farmers from making informed selling decisions, affecting their financial stability.

Additionally, sustainability monitoring is becoming increasingly important in modern agriculture. The excessive use of chemical fertilizers, pesticides, and water resources contributes to soil degradation, water pollution, and greenhouse gas emissions. Farmers need better tools to monitor their environmental impact and adopt more sustainable agricultural practices. However, current agricultural systems lack integrated solutions for tracking sustainability metrics in real-time.

To address these challenges, there is a need for a smart and data-driven farm management system that integrates modern technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and Big Data analytics. This system should enable farmers to monitor real-time farm conditions, make data-driven decisions, and optimize resource utilization for improved productivity and sustainability.

3.2 OBJECTIVES

The primary objective of this project is to empower farmers through technology by minimizing their reliance on middlemen, enhancing profit margins, and providing real-time support for informed agricultural decisions. By integrating AI, remote sensing, and data analytics, the system aims to make farming more efficient, sustainable, and economically rewarding.

To improve the livelihood of farmers and promote smarter agriculture, the project focuses on integrating technology-driven solutions that address core challenges in farming. The aim is to reduce dependency on middlemen, increase profitability, and provide timely information to support better decision-making in agriculture.

1. The system will directly connect farmers to markets, reducing the influence of middlemen and ensuring they receive fair prices for their produce.
2. It will help maximize farmer profits by analyzing market trends, demand patterns, and suggesting the most profitable times and locations to sell crops.
3. Real-time weather updates will be provided to help farmers plan irrigation, sowing, and harvesting activities, reducing risks from unexpected climate changes.
4. The platform will offer current market prices of various crops, helping farmers make informed selling decisions based on accurate and timely data.
5. The system will also include a plant disease detection feature using image analysis, allowing early identification and treatment to minimize crop damage and yield loss.

CHAPTER 4

REQUIREMENTS SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

The functional requirements of the Farmease system define the key features essential for improving farm management, resource optimization, and productivity. These functionalities ensure that farmers have access to real-time information, automation, and marketplace services that simplify agricultural activities and decision-making. The system includes user authentication, weather updates, product management, plant disease detection, tool rental services, and notifications, all designed to enhance efficiency and sustainability in farming.

1. User Roles and Authentication: The system provides secure login and role-based access control to ensure different users, such as farmers, buyers, and tool renters, have appropriate permissions and secure interactions.

2. Weather Condition Updates: Farmers receive real-time weather forecasts and climate data, helping them plan irrigation, harvesting, and pest control strategies to minimize risks from adverse weather conditions.

3. Product Management: This feature allows farmers to list, manage, and sell agricultural products efficiently, providing an easy-to-use platform for buyers to access quality farm produce.

4. Plant Disease Detection: An AI-based system enables farmers to upload images of diseased plants, analyze them for infections, and receive recommendations for effective treatment, reducing crop losses.

5. Tool Rental Service: A dedicated marketplace where farmers can rent and lease agricultural tools and machinery, ensuring small-scale farmers have access to expensive equipment at affordable rates.

4.2 NON FUNCTIONAL REQUIREMENTS

The non-functional requirements of the Farmease system define the essential attributes that ensure the platform operates efficiently, securely, and reliably. These requirements enhance the system's performance, scalability, usability, and security, ensuring a seamless user experience and long-term sustainability. Non-functional requirements are critical for maintaining system integrity, responsiveness, and overall user satisfaction.

1. Reliability: The system must function without failures, ensuring consistent availability and accuracy in data processing. It should handle agricultural operations efficiently without unexpected downtimes.

2. Scalability: As the number of users, transactions, and data volumes increase, the system must scale efficiently. This ensures smooth performance even with a growing number of farmers, buyers, and service providers.

3. Performance: The platform should offer fast response times, minimizing delays in data retrieval, processing, and real-time updates such as weather conditions and market prices.

4. Usability: The system must be user-friendly, intuitive, and accessible to farmers with varying levels of technical expertise, ensuring easy navigation and interaction with features.

5. Security: Strong authentication mechanisms and data encryption must be implemented to protect user information, transactions, and sensitive agricultural data from cyber threats and unauthorized access.

By ensuring these non-functional requirements, Farmease guarantees a reliable, high-performing, and secure platform that enhances modern agricultural practices and supports a wide range of users.

4.3 HARDWARE SPECIFICATION

The system requires the following hardware components:

Processor: A modern multi-core processor is required to handle multiple tasks simultaneously and efficiently execute computations related to farm management and image processing.

RAM: A minimum of 8 GB RAM is recommended to ensure smooth system operation, particularly for processing large datasets and handling real-time requests.

Storage: The system should have at least 256 GB of storage to store essential data, including user profiles, marketplace transactions, and plant disease detection images.

Network: A stable network connection is required for seamless communication between users, cloud-based services, and the integrated modules of the system.

4.4 SOFTWARE SPECIFICATION

The software components necessary for the development and operation of the system include:

Frontend: The user interface is developed using HTML and CSS, ensuring a responsive and user-friendly design for farmers and stakeholders.

Backend: The core functionalities of the system are implemented using JavaScript, PHP, and Python, enabling efficient data processing, marketplace operations, and plant disease detection.

Database Management System (DBMS): A robust database is required to store large volumes of agricultural data, including crop information, weather updates, and transaction records.

Operating System (OS): The system can run on Windows 10/11 or Linux (or higher), providing flexibility for deployment based on user preferences.

These system requirements ensure that Farmease operates effectively, delivering high performance and reliability while supporting farmers in managing their agricultural activities efficiently.

4.5 USER REQUIREMENTS

The Farmease – Integrated Farm Management and Marketplace System is designed to cater to different user roles, ensuring smooth interaction between stakeholders in the agricultural ecosystem. The primary users of the system include Admins, Farmers(user) each with specific functionalities and requirements.

4.5.1 Admin

- Manage user accounts, ensuring security and smooth registration processes.
- Update market prices of agricultural products based on market trends.
- Manage system-generated notifications, including critical weather alerts, to assist farmers in planning their activities effectively.

4.5.2 Farmers

- List and sell their agricultural products directly through the platform.
- Rent farming tools and equipment for improved productivity.
- Receive real-time weather condition notifications to optimize farming decisions.
- Get regular updates on market prices to ensure fair trade.
- Detect plant diseases by uploading images, allowing the system to analyze and suggest potential solutions.
- Browse available farm products and make purchases directly from farmers.
- View current market prices to make informed purchasing decisions.

4.5.3 Buyers

- Browse available farm products and make purchases directly from farmers.
- View current market prices to make informed purchasing decisions.
- Buyers can track their orders and get delivery updates.

By fulfilling these user requirements , Farmease ensures a seamless and efficient market

place for farmers and buyers ,while also allowing administrators to manage platform operations effectively.

CHAPTER 5

SYSTEM DESIGN

5.1 USE CASE DIAGRAM

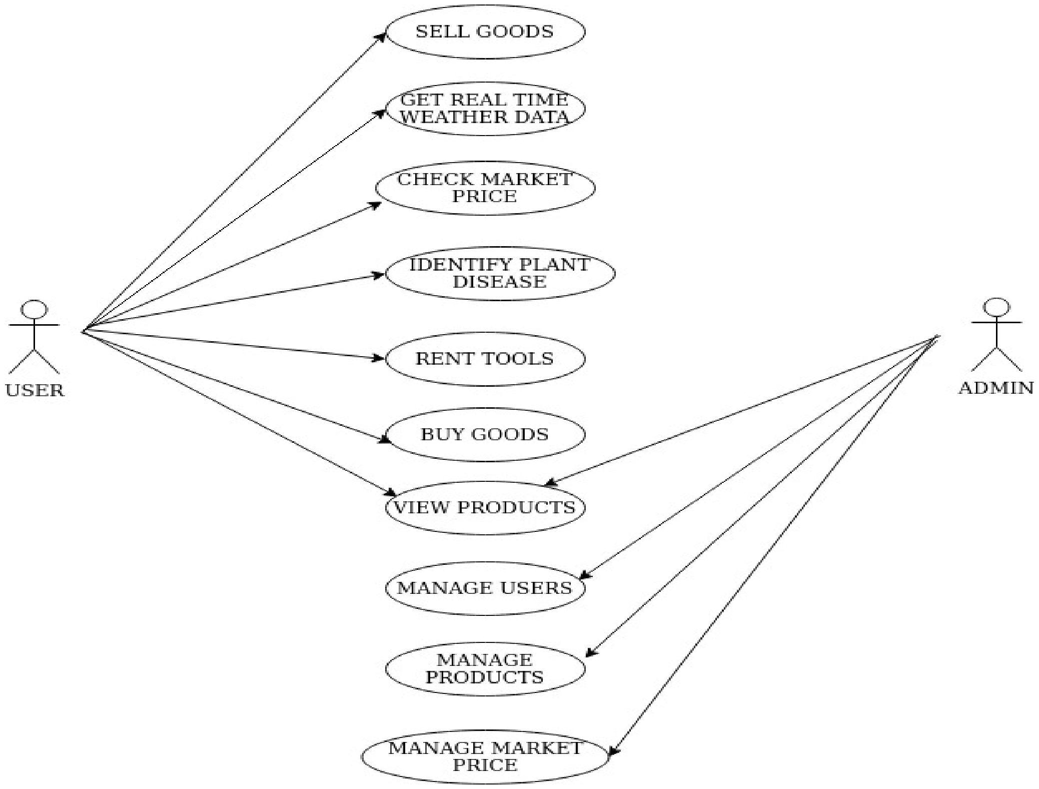


Fig. 5.1 Use Case Diagram of Farmease

This Use Case Diagram represents the functionalities of the Farmease – Integrated Farm Management and Marketplace System, showing interactions between users and the system. There are two primary actors: the User (such as farmers, buyers, or sellers) and the Admin (who manages the platform). The User can perform various actions, including selling goods, getting real-time weather data, checking market prices, identifying plant diseases by uploading images, renting tools, buying goods, and viewing available products. On the other hand, the Admin has key management responsibilities, such as managing users, overseeing product listings, and regulating market prices. This diagram helps in understanding the system's workflow, ensuring clarity in system design and development.

5.2 ER DIAGRAM

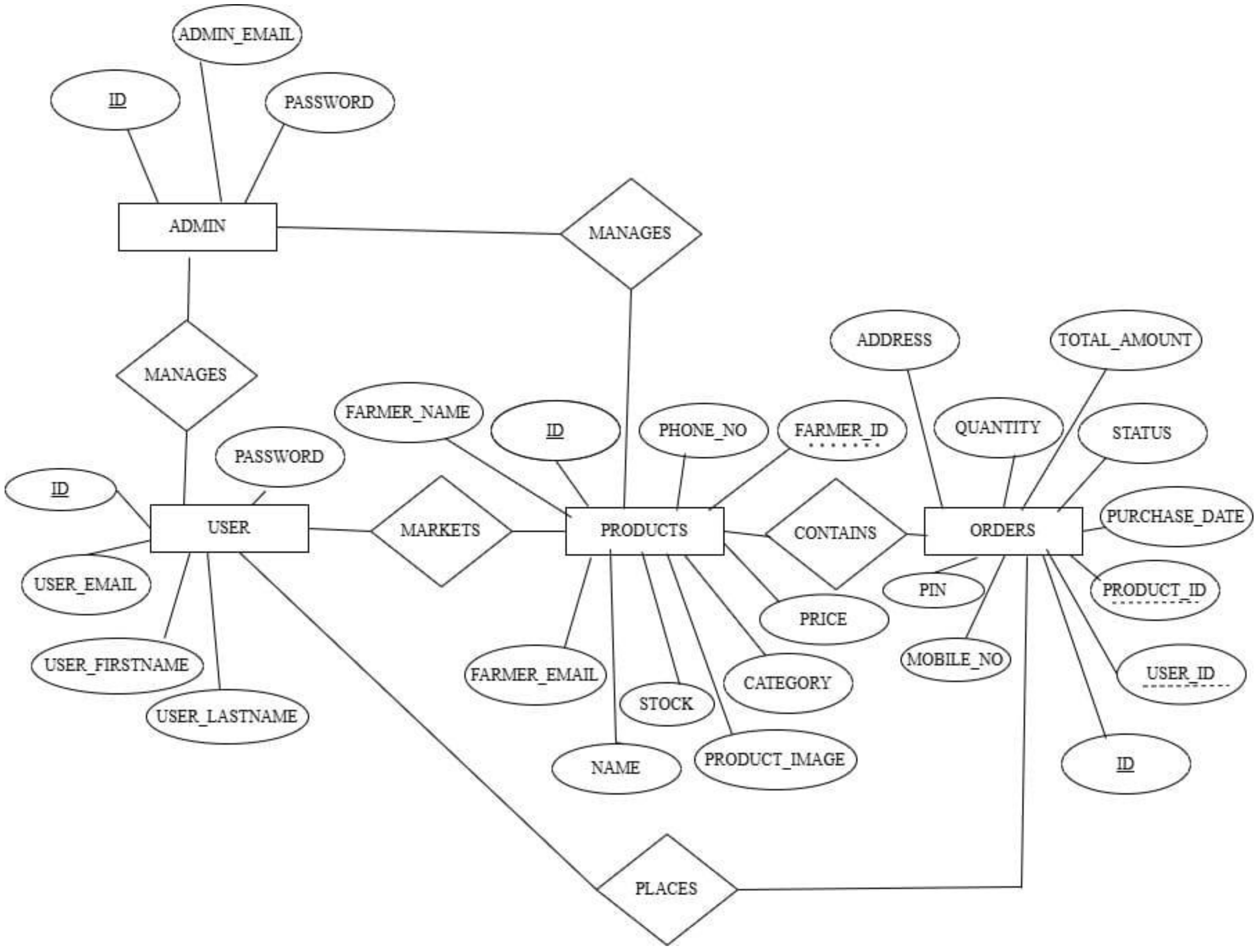


Fig. 5.2 ER Diagram of Farmease

The ER Diagram of Farmease – Integrated Farm Management and Marketplace System represents the relationships between different entities in the system. The Admin entity manages both users and farmers, ensuring system regulation. The User entity contains attributes such as user ID, email, and password, and they can browse and market products. The Product entity, linked to farmers, includes product details such as product ID, name, category, stock, price, and images. Users can place orders, which contain details like order ID, purchase date, quantity, total amount, and status. The system also stores farmer details, including farmer ID, name, phone number, and email. This ER diagram efficiently models the database structure, enabling seamless interactions between administrators, farmers, users, products, and orders.

5.3 ARCHITECTURAL DESIGN

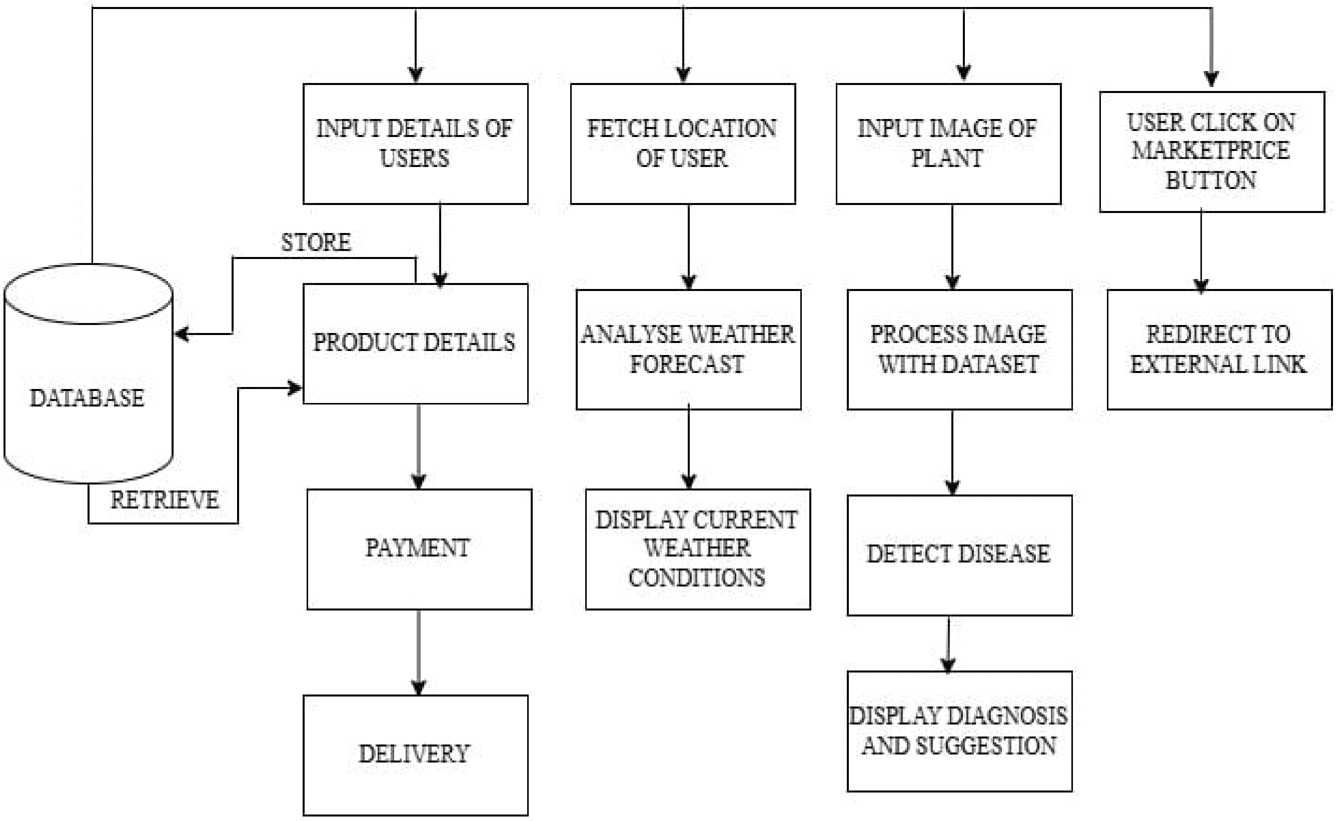
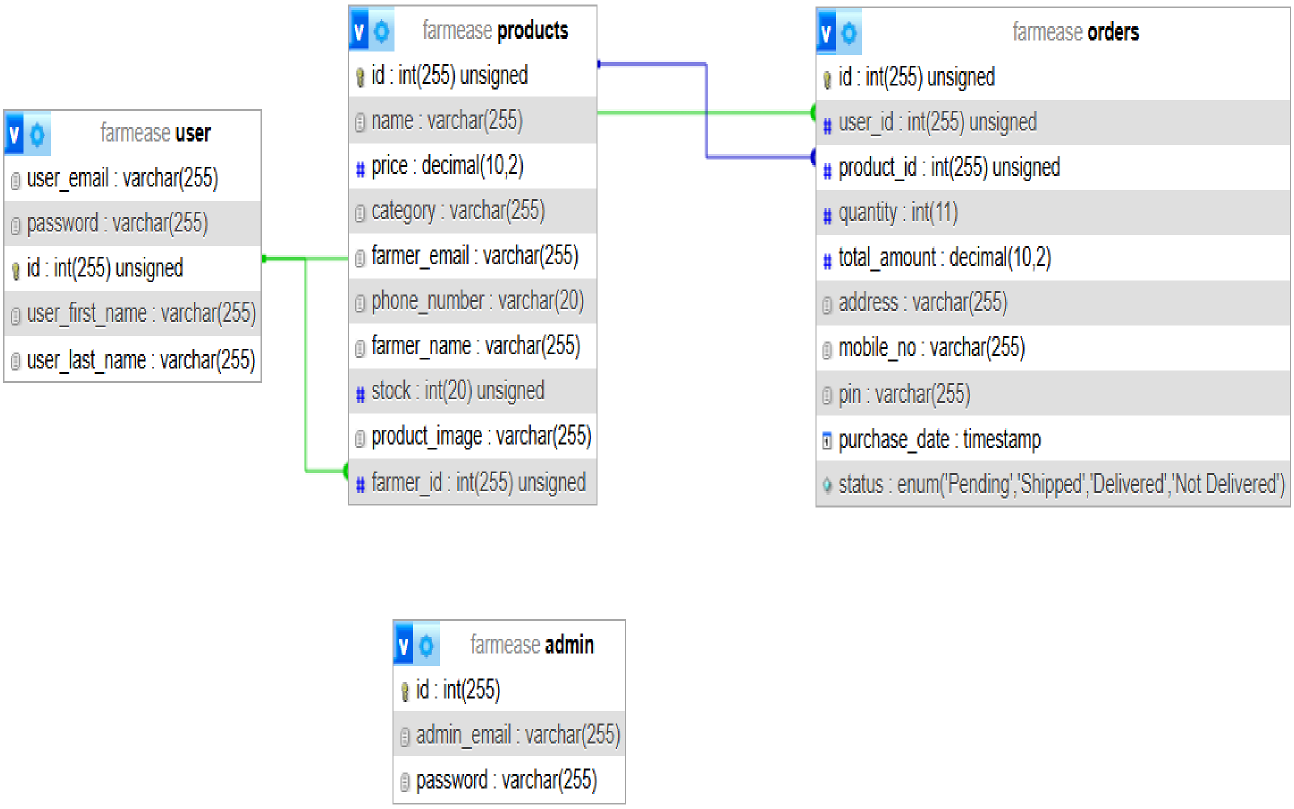


Fig. 5.3 Architectural Design of Farmease

The System Flow Diagram represents the core functionalities of the Farmease – Integrated Farm Management and Marketplace System, ensuring a seamless experience for users. The system starts by storing and retrieving user details in a database. Users can input their details, fetch their location, upload plant images for disease detection, or check marketplace prices. The weather forecast analysis processes the user's location to display current weather conditions, helping farmers make informed decisions. When a user uploads an image of a plant, the system processes it using a pre-trained dataset to detect diseases and provides relevant diagnoses and suggestions. Additionally, users clicking on the marketplace button are redirected to an external link for further transactions. The system also manages product details, enabling secure payment processing and delivery services. This structured approach enhances farm management, marketplace accessibility, and disease detection, offering farmers a comprehensive digital solution to improve agricultural productivity.

5.4 DATABASE DESIGN



farmease **user**

user_email : varchar(255)
password : varchar(255)
id : int(255) unsigned
user_first_name : varchar(255)
user_last_name : varchar(255)

farmease **products**

id : int(255) unsigned
name : varchar(255)
price : decimal(10,2)
category : varchar(255)
farmer_email : varchar(255)
phone_number : varchar(20)
farmer_name : varchar(255)
stock : int(20) unsigned
product_image : varchar(255)
farmer_id : int(255) unsigned

farmease **orders**

id : int(255) unsigned
user_id : int(255) unsigned
product_id : int(255) unsigned
quantity : int(11)
total_amount : decimal(10,2)
address : varchar(255)
mobile_no : varchar(255)
pin : varchar(255)
purchase_date : timestamp
status : enum('Pending','Shipped','Delivered','Not Delivered')

farmease **admin**

id : int(255)
admin_email : varchar(255)
password : varchar(255)

Fig. 5.4 Database Design of Farmease

The Farmease Database Schema is designed to efficiently manage users, products, orders, and administrators within the platform. The user table stores essential user information such as email, password, and personal details. The products table manages agricultural product listings, including price, category, stock, and farmer details. The orders table tracks transactions with attributes like quantity, total amount, delivery address, and order status (Pending, Shipped, Delivered, or Not Delivered). The admin table is responsible for handling administrative access and platform management. Relationships between these tables ensure seamless data flow, enabling users to browse and purchase products, while farmers can list and sell their items effectively. The schema ensures scalability, security, and efficient transaction management for a smooth user experience.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Home Page

```
<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Home - FarmEase</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="navbar" style="background-color: #27dd2797; text-align: center; height:
60px; line-height: 60px;">
            <h1 class="logo" style="margin: 0;">FarmEase</h1>

        </div>

        <nav style="background-color: #27dd2797; padding: 10px; text-align: center;">
```

```
            <a href="home.html">Home</a> |

            <a href="sell.html">Sell Products</a> |

            <a href="account.php">Account</a> |

            <a href="weather.html">weather</a> |

            <a href="plant.html">Plant Disease Prediction</a> |

            <a  href="https://vegetablemarketprice.com/market/kerala/today">Market  Price</a>

            <div style="margin-top: 10px;">

                <input type="text" id="searchInput" placeholder="Search products..."
oninput="searchProduct()">
            </div>

</nav>

</header>

    <main>

        <section class="categories">

            <button onclick="filterCategory('All')">All</button>

            <button onclick="filterCategory('Fruits')">Fruits</button>

            <button onclick="filterCategory('Vegetables')">Vegetables</button>

            <button onclick="filterCategory('Tools')">Tools</button>
        </section>
        <section class="products">

            <h2>Trending Products</h2>

            <div id="products-container"></div>

        </section>

    </main>

    <script>

        // Function to filter products by category
        function filterCategory(category) {
            const productsContainer = document.getElementById("products-container");
            productsContainer.innerHTML = "";


            // Fetch products for the selected category
            fetch(fetch_products.php?category=${category})
                .then(response => response.json())

                .then(products => {

                    if (products.length > 0) {
                        products.forEach(product => {
                            const productCard = document.createElement("div");
                            productCard.className = "product-card";


                            const priceText = product.category.toLowerCase() === "tools" ? "<p
class='price'>Rent</p>" : <p class='price'>₹${product.price}/kg</p>;


                            productCard.innerHTML = `
                        <img src="${product.product_image}" alt="${product.name}"
style="width:100%">
                        <p class="product-name">${product.name}</p>

                        <p class="product-desc">${product.category} - By
${product.farmer_name}</p>

                        ${priceText}

                    productCard.addEventListener("click", () => {
                    window.location.href = product_details.html?id=${product.id};
                    });

                    productsContainer.appendChild(productCard);

               });


               } else {

                  productsContainer.innerHTML = "<p>No products found in this
category.</p>";

               }

            })

            .catch(error => console.error("Error fetching products:", error));

      }


      document.addEventListener("DOMContentLoaded", () => {
         filterCategory('All');
      });
```

```
        function searchProduct() {
    const searchQuery =
document.getElementById("searchInput").value.trim().toLowerCase();
    const productsContainer = document.getElementById("products-container");
    productsContainer.innerHTML = "";


    fetch(search_products.php?query=${searchQuery})

        .then(response => response.json())

        .then(products => {

            if (products.length > 0) {
                products.forEach(product => {
    const productCard = document.createElement("div");
    productCard.className = "product-card";

    const priceText = product.category.toLowerCase() === "tools" ? "<p class='price'>Rent
</p>"  : <p class='price'>₹${product.price}/kg</p>;


    productCard.innerHTML = `

        <img src="${product.product_image}" alt="${product.name}" style="width:100%">

        <p class="product-name">${product.name}</p>

        <p class="product-desc">${product.category} - By ${product.farmer_name}</p>

        ${priceText}

    `;

    productCard.addEventListener("click", () => {
```

```
    window.location.href = product_details.html?id=${product.id};
    });

    productsContainer.appendChild(productCard);

});

            } else {

                productsContainer.innerHTML = "<p>No products found.</p>";

            }

        })

        .catch(error => console.error("Error searching products:", error));
}
    </script>
</body>
</html>
```

6.2 Account Page

```
<?php
     session_start();
    if (!isset($_SESSION['user_email'])) {
        header("Location: index.html");
        exit();
    }
?>

<!DOCTYPE html>
```

```
 <html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Account Page</title>
    <link rel="stylesheet" href="style.css">
    <script defer src="account.js"></script>
    <style>
        body, h1, p, button {
            margin: 0;
            padding: 0;
        }
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 0px;
        }
        .header {
    background-color: #232f3e;
    color: white;
    padding: 0; display: flex;
    align-items: center;
    justify-content: space-between;
}
        .header h1 {
        margin: 0;
            font-size: 24px;
        }
        .header button {

            background-color: #ff5e57;

            color: white;

            border: none;

            padding: 10px 20px;

            cursor: pointer;

            border-radius: 5px;

            width: 100p


}

  .header button:hover {

background-color: #ff2a15;


        }

        .container {

            background-color: white;

            padding-top: 10px;

            padding-bottom: 10%;

            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

            width: 100%;

        }

        profile img {

            width: 100px;

            height: 100px;

            border-radius: 50%;
            object-fit: cover;
        }

        .section {
            margin-bottom: 20px;
        }

        .section button {
            padding: 10px 15px;
            background-color: #232f3e;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 300px;
            display: block;
            margin: auto;
        }
        .section button:hover {
            background-color: #3a4455;
        }
        #username{
            padding-left: 10px;
        }
    </style>
</head>
```

```
<body>
  <header>
     <div class="navbar">
       <h1 class="logo">FarmEase</h1>
     </div>
     <nav style="background-color: #27dd2797; padding: 10px; text-align: center">
       <a href="home.html">Home</a> |
       <a href="sell.html">Sell Products</a> |
       <a href="account.php">Account</a>
     </nav>
  </header>

  <div class="header">
     <h1 id="username">
       <?php
         if (isset($_SESSION['user_name'])) {
           echo "Hello, " . $_SESSION['user_name'];
         } else {
           echo "Hello, Guest";
         }
       ?>
     </h1>
     <button onclick="logout()">Logout</button>
  </div>
```

```
<div class="container">

<div class="section">

    <h2>My Orders</h2>

    <button onclick="viewmyOrders()">View Orders</button>

  </div>

  <div class="section">

    <h2>Orders Received</h2>

    <button onclick="viewOrders()">View Orders</button>

  </div>

  <div class="section">

    <h2>Your Lists</h2>

    <button onclick="viewLists()">See All Lists</button>

  </div>

</div>

<script>

  function changeProfilePicture() {

    const newPic = prompt("Enter the URL of the new profile picture:");
    if (newPic) {
        document.getElementById("profile-pic").src = newPic;
        alert("Profile picture updated successfully!");
    }

}



function editAccountName() {

    const newName = prompt("Enter your new account name:");
    if (newName) {
        document.getElementById("new_name").value = newName;
        document.getElementById("change-name-form").submit();
    }

}

function logout() {
    window.location.href = "index.html";
}


function viewOrders() {



    window.location.href = "orders.php";

}

function viewmyOrders(){
    window.location.href="my_orders.php";
}
```

```
        function viewLists() {
        window.location.href = "sold_products.php";
    }
    </script>

</body>
</html>
```

6.3 Model Training Code

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import os

train_dir = "dataset/train"
val_dir  =  "dataset/valid"
img_size = (128, 128)
batch_size = 16
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    train_dir,
    image_size=img_size,
    batch_size=batch_size,
    label_mode="categorical",
    shuffle=True
)
```

```
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    val_dir,

    image_size=img_size,
    batch_size=batch_size,
    label_mode="categorical"
)

model = keras.Sequential([
    layers.Conv2D(32, (3, 3), padding="same", input_shape=(128, 128, 3)),
    layers.BatchNormalization(),
    layers.LeakyReLU(negative_slope=0.1),
    layers.MaxPooling2D(),

    layers.Conv2D(64, (3, 3), padding="same"),
    layers.BatchNormalization(),
    layers.LeakyReLU(negative_slope=0.1),
    layers.MaxPooling2D(),

    layers.Conv2D(128, (3, 3), padding="same"),
    layers.BatchNormalization(),
    layers.LeakyReLU(negative_slope=0.1),
    layers.MaxPooling2D(),
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation="relu"),


    layers.Dropout(0.3),
    layers.Dense(38, activation="softmax")
])


model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=10
)


model.save("plant_disease_model.h5")
```

6.4 Plant Disease Detection

```
from flask import Flask, request, jsonify

from flask_cors import CORS


from keras.models import load_model

from PIL import Image

import numpy as np

import io


app = Flask(_name_)

CORS(app)


model_path = "C:/xampp/htdocs/FarmEase/plant_disease_model.h5"

model = load_model(model_path)

class_labels = [

    "Apple__Apple_scab", "Apple_Black_rot", "Apple_Cedar_apple_rust", "Apple__healthy",

    "Blueberry  healthy", "Cherry(including_sour)  healthy",
"Cherry(including_sour)___Powdery_mildew",

    "Corn_(maize)  Cercospora_leaf_spot Gray_leaf_spot", "Corn(maize)  Common_rust",
"Corn_(maize)___healthy",

    "Corn_(maize)  Northern_Leaf_Blight", "Grape_Black_rot",
"Grape_Esca(Black_Measles)", "Grape___healthy",
    "Grape  Leaf_blight(Isariopsis_Leaf_Spot)",
"Orange  Haunglongbing(Citrus_greening)",
    "Peach  Bacterial_spot", "Peach_healthy", "Pepper,_bell_Bacterial_spot",
"Pepper,_bell  healthy",
    "Potato__Early_blight", "Potato_healthy", "Potato_Late_blight", "Raspberry__healthy",

    "Soybean  healthy", "Squash_Powdery_mildew", "Strawberry_healthy",
"Strawberry  Leaf_scorch",
    "Tomato  Bacterial_spot", "Tomato_Early_blight", "Tomato_healthy",
"Tomato  Late_blight",
    "Tomato  Leaf_Mold", "Tomato_Septoria_leaf_spot", "Tomato  Spider_mites Two-
spotted_spider_mite",
    "Tomato  Target_Spot", "Tomato_Tomato_mosaic_virus",
"Tomato  Tomato_Yellow_Leaf_Curl_Virus"
]

def preprocess_image(image):
    image = image.resize((128, 128))  # Resize to match training data
    image = np.asarray(image)


    image = np.expand_dims(image, axis=0)  # Expand batch dimension
    return image


@app.route('/predict', methods=['POST'])
def predict():
    if 'image' not in request.files:
        return jsonify({'success': False, 'error': 'No file uploaded'}), 400

    file = request.files['image']

    if file.filename == '':
        return jsonify({'success': False, 'error': 'No selected file'}), 400

    try:
        image = Image.open(io.BytesIO(file.read()))
        processed_image = preprocess_image(image)
        prediction = model.predict(processed_image)



        print("Prediction Probabilities:", prediction)  # Debugging
        predicted_class = np.argmax(prediction, axis=1)[0]
        disease_label = class_labels[predicted_class]
        return jsonify({'success': True, 'disease': disease_label})
    except Exception as e:
        return jsonify({'success': False, 'error': str(e)}), 500
if _name_ == '_main_':
    app.run(debug=True)
```

6.5 Database Connection

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "FarmEase";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}

$conn->set_charset("utf8");

?>
```

CHAPTER 7

SYSTEM TESTING

7.1 TEST CASES

Test Case 1: User Login Functionality
Verify that users can successfully log in with valid credentials and receive an error message when invalid credentials are entered.

Test Case 2: Add Product to Marketplace
Check if users can add a product by entering valid details such as product name, price, category,image URL, stock quantity, and seller information, and receive a confirmation message upon successful addition.

Test Case 3: Product Search Functionality
Ensure that users can search for products using keywords or by selecting a specific category and receive a relevant list of matching products.

Test Case 4: Place Order Functionality
Verify that users can place an order successfully after selecting a product and receive confirmation that the purchase has been completed.

Test Case 5: Display Weather Data
Check if real-time weather information is displayed correctly based on the user's location, including details such as temperature, humidity, and general weather condition.

Test Case 6: Plant Disease Detection
Test the functionality that allows the users to upload an image of a plant leaf and receive the name of the detected disease or a message indicating no disease was found, along with appropriate suggestions.

Test Case 7: Edit Product Details
Verify that users can successfully update the details of products they have previously added and that the updated information is correctly reflected.

Test Case 8: Admin Manage Users and Products

Ensure that admins can add, update, or delete user and the product information, and that these changes are accurately reflected in the system.

Test Case 9: Check Market Price via External Link

Verify that users can view current market prices for agricultural products by clicking an external link that opens a relevant web page.

Test Case 10: Logout Functionality

Ensure that users can securely log out from the system and are redirected to the login page after ending their session.

7.1 TEST CASES

Table 7.1 Test Cases for Farmease

Test Case ID	Description	Input	Expected Output	Actual Output
TC_01	User Authentication	Valid and invalid login credentials	Successful login or error message shown	Pass
TC_02	Product Addition	Product name, price, category, image url, stock...	Product is added successfully	Pass
TC_03	Product Search	Search keyword or category selection	Relevant products displayed or show product not found	Pass
TC_04	Order Placement	Selected product, payment details	Order confirmation message displayed	Pass
TC_05	Weather Monitoring	User location	Real-time weather data displayed	Pass
TC_06	Plant Disease Detection	Upload plant image	Disease detected or "No disease found"	Pass
TC_07	Update Product Details	Edit product details (price, stock, etc.)	Product details updated successfully	Pass
TC_08	Admin Manage Users and Products	Add, update, or remove users and products	Changes reflected in the system	Pass
TC_09	Check Market Price	Click on the market price option	Redirected to an external pricing source	Pass
TC_10	Logout functionality	Click on the Logout button	Logout successfully	Pass

7.2 TRACEABILITY MATRIX

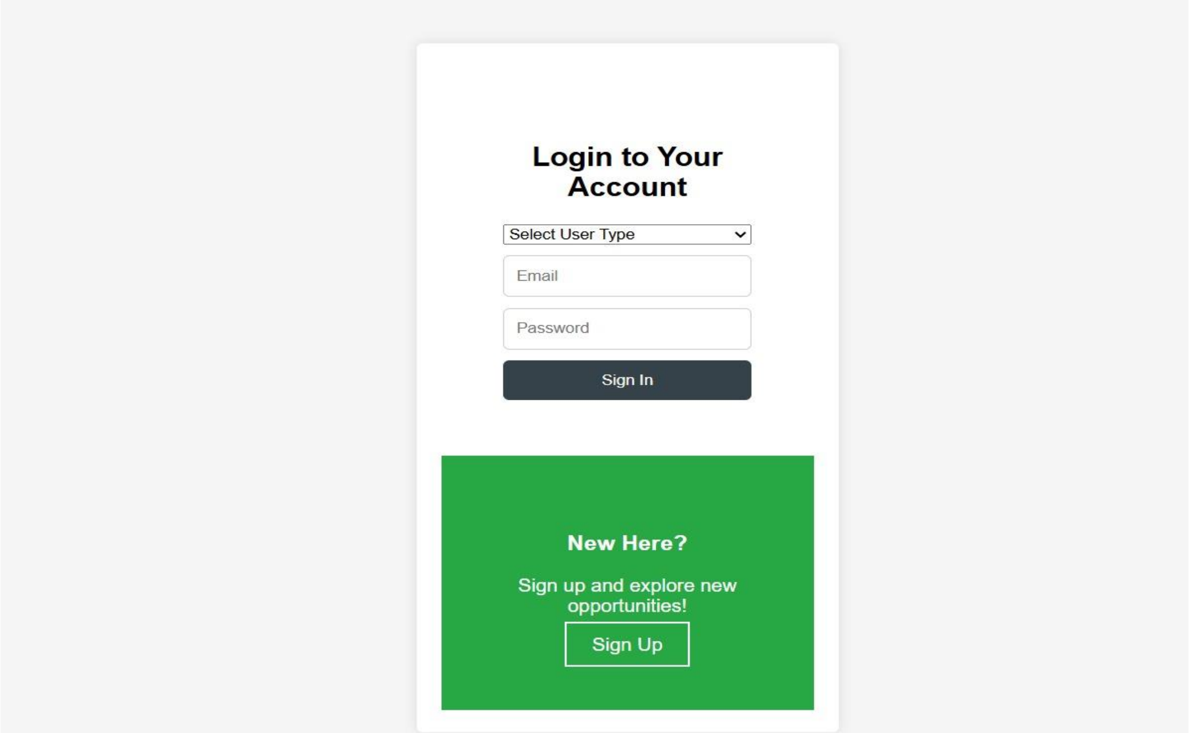
Table 7.2 Traceability Matrix for Farmease

Requirement ID	Requirement Description	Linked Test Case ID
R01	The system shall allow users to log in using valid credentials and show an error for invalid attempts	TC_01
R02	The system shall allow users to add products with complete and valid information to the marketplace.	TC_02
R03	The system shall allow users to search for products by keyword or category.	TC_03
R04	The system shall support order placement for selected products including payment confirmation.	TC_04
R05	The system shall display real-time weather data based on the user's location.	TC_05
R06	The system shall detect plant diseases from uploaded images and suggest appropriate remedies.	TC_06
R07	The system shall allow users to edit and update product details such as price and stock.	TC_07
R08	The admin shall be able to manage users and products by adding, updating, or deleting them.	TC_08
R09	The system shall allow users to view current market prices by accessing an external source.	TC_09
R10	The system shall allow users to securely log out and return to the login screen.	TC_10

CHAPTER 8

RESULT

8.1 Login Page



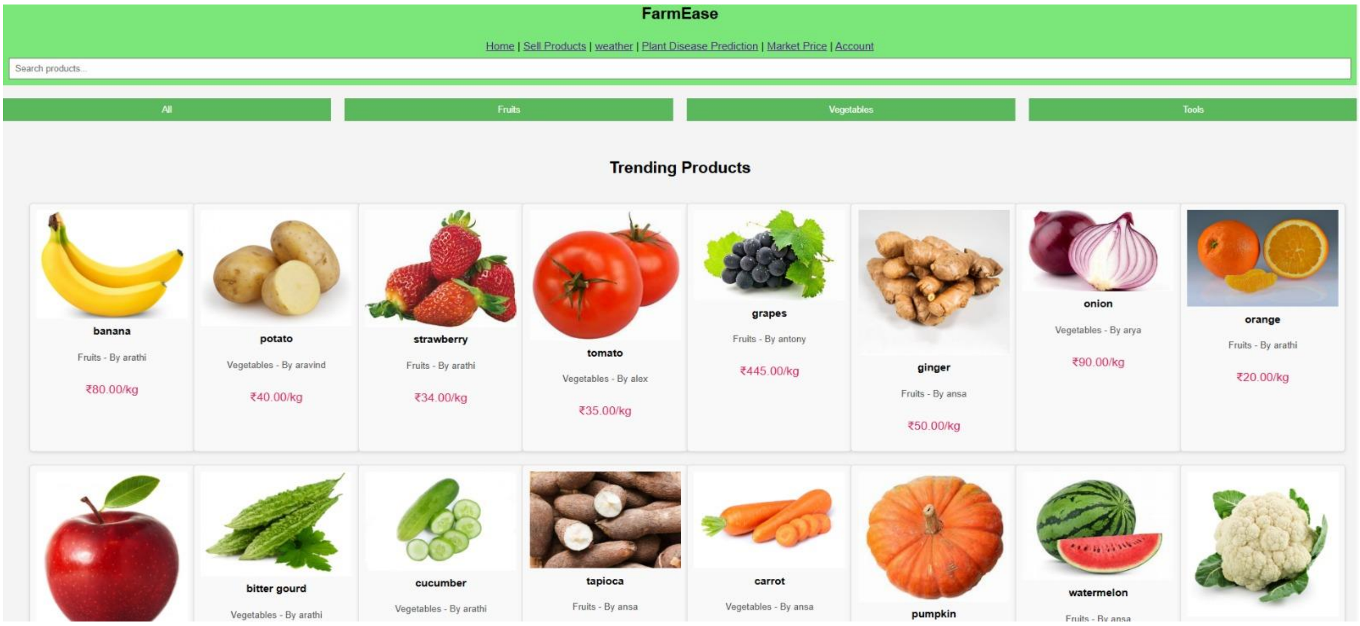
Login to Your Account

New Here?

Sign up and explore new opportunities!

Fig. 8.1 Login Page

8.2 Home Page



FarmEase

Trending Products

banana
Fruits - By arathi
₹80.00/kg

potato
Vegetables - By aravind
₹40.00/kg

strawberry
Fruits - By arathi
₹34.00/kg

tomato
Vegetables - By alex
₹35.00/kg

grapes
Fruits - By antony
₹445.00/kg

ginger
Fruits - By ansa
₹50.00/kg

onion
Vegetables - By arya
₹90.00/kg

orange
Fruits - By arathi
₹20.00/kg

bitter gourd
Vegetables - By arathi

cucumber
Vegetables - By arathi

tapioca
Fruits - By ansa

carrot
Vegetables - By ansa

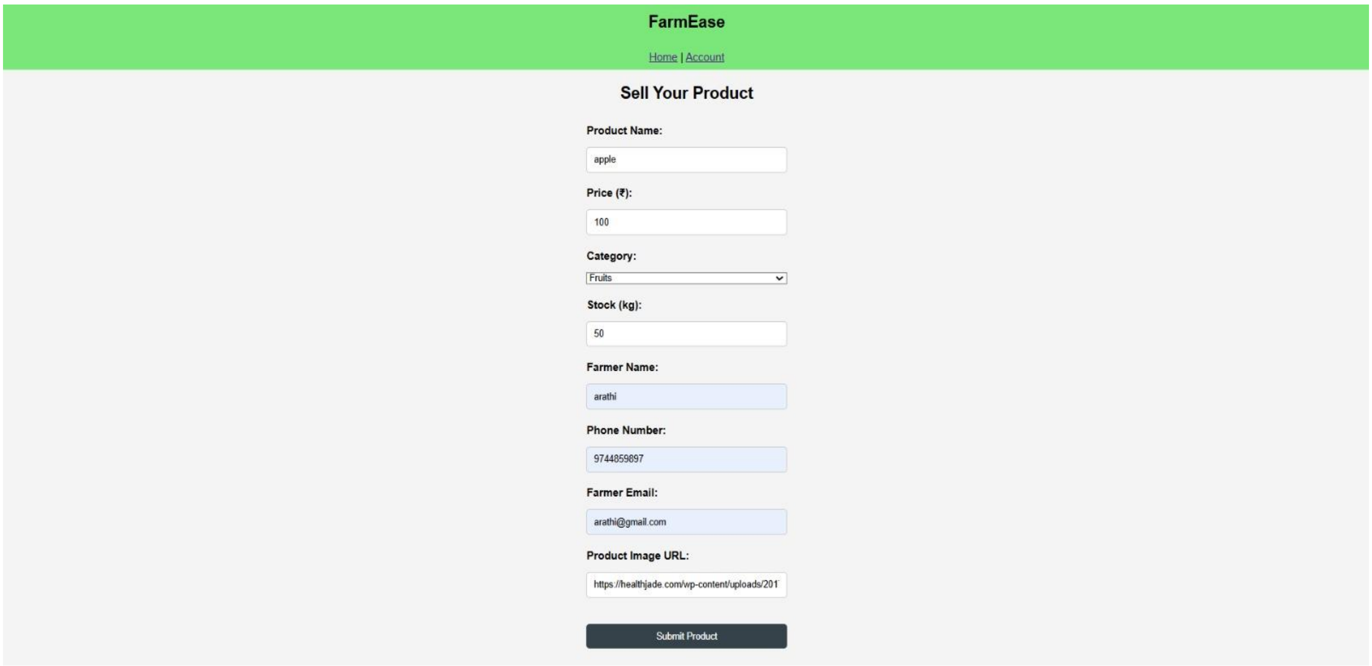
pumpkin

watermelon
Fruits - By ansa

Home | Sell Products | weather | Plant Disease Prediction | Market Price | Account
Search products...
All
Fruits
Vegetables
Tools

Fig. 8.2 Home Page

8.3 Seller Dashboard



FarmEase

Sell Your Product

Product Name:
apple

Price (₹):
100

Category:
Fruits

Stock (kg):
50

Farmer Name:
arathi

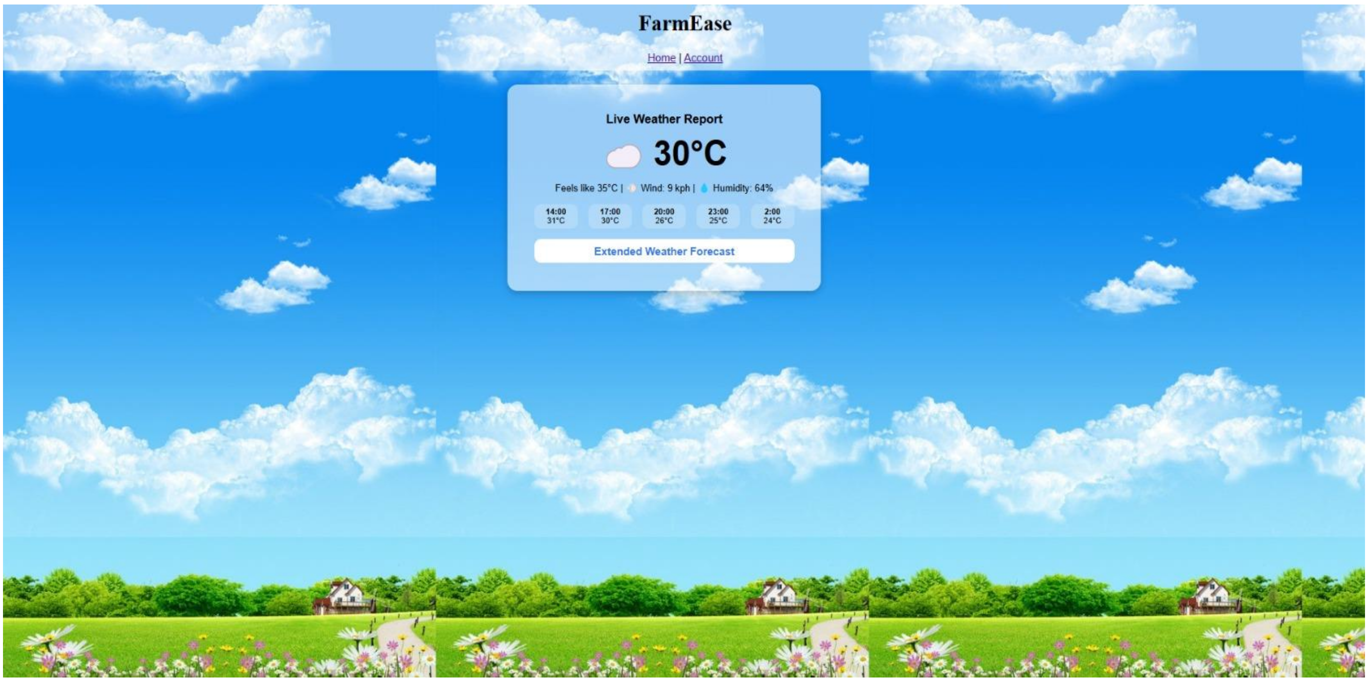
Phone Number:
9744859897

Farmer Email:
arathi@gmail.com

Product Image URL:
https://healthjade.com/wp-content/uploads/201

Fig. 8.3 Seller Dashboard

8.4 Weather



FarmEase

Live Weather Report

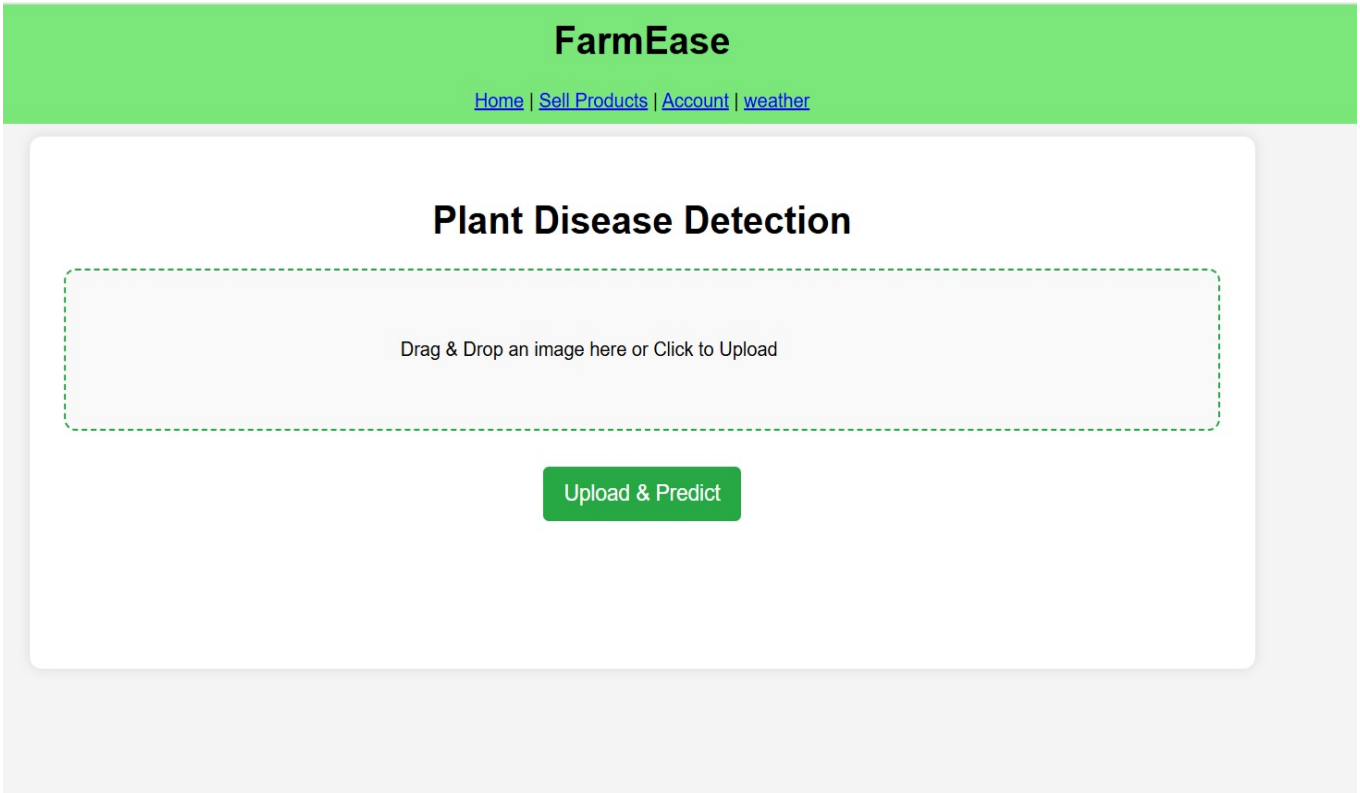
30°C

Feels like 35°C | Wind: 9 kph | Humidity: 64%

14:00 31°C | 17:00 30°C | 20:00 26°C | 23:00 25°C | 2:00 24°C

Fig. 8.4 Weather

8.5 Plant Disease Detection Input

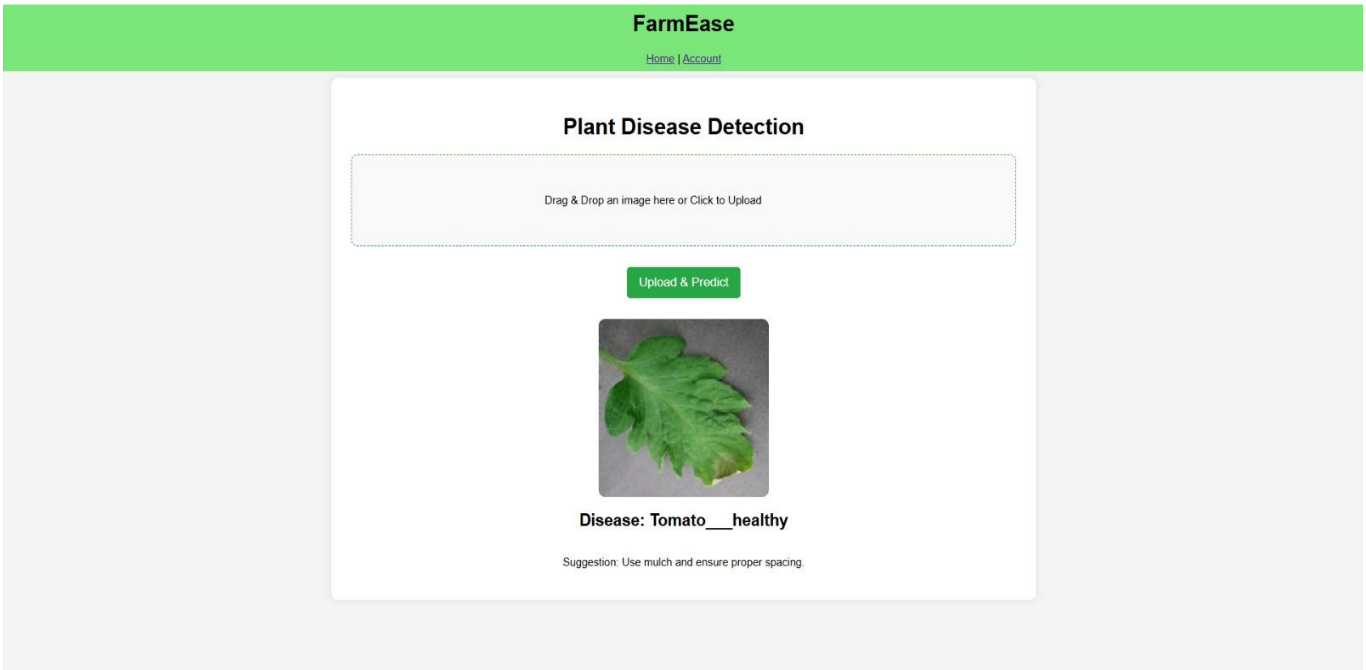


Plant Disease Detection

Drag & Drop an image here or Click to Upload

Fig. 8.5 Plant Disease Detection Input

8.6 Plant Disease Detection Output



Plant Disease Detection

Drag & Drop an image here or Click to Upload

Disease: Tomato___healthy

Suggestion: Use mulch and ensure proper spacing.

Fig. 8.6 Plant Disease Detection Output

CHAPTER 9
CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

The Farmease database schema is a well-structured system that efficiently manages user interactions, product listings, order tracking, and administrative control. By integrating a relational database model, the platform ensures data consistency, security, and ease of access. Users can seamlessly browse and purchase agricultural products, while farmers can effectively manage their inventory. The schema supports real-time order processing and status tracking, contributing to a streamlined marketplace for agricultural needs.

9.2 FUTURE SCOPE

The Farmease platform can be enhanced with several advanced features:

1. AI-Powered Recommendations: Implementing machine learning to suggest products based on user preferences and past purchases.
2. Blockchain for Secure Transactions: Ensuring transparency and security in farmer-buyer transactions.
3. Automated Order Management System: Integrating smart logistics to optimize delivery processes.
4. Mobile Application Development: Expanding accessibility with a dedicated mobile app.
5. Weather and Crop Advisory Services: Providing real-time weather updates and AI-based crop health analysis for farmers.

By incorporating these features, Farmease can evolve into a comprehensive smart farming ecosystem, improving productivity and bridging the gap between farmers and consumers.

REFERENCES

[1] Transforming Agriculture with Intelligent Data Management and Insights. *IEEE Xplore*, 2023.

[2] Web-Based Agricultural Monitoring and Sales Management System. *IEEE International* . *Conference on Smart Computing and Electronic Enterprise*, 2022.

[3] IoT-Based Smart Agriculture Monitoring System with Predictive Analysis. *IEEE International Conference on Computational Intelligence and Communication Networks*, 2023.

[4] Agriculture Digitalization: A Global Examination Based on Bibliometric Analysis. *IEEE Access*, 2023

[5] Artificial Intelligence and Internet of Things for Sustainable Farming and Smart Agriculture. *IEEE Internet of Things Journal*, 2022.

[6] IoT-Based Smart Agriculture Monitoring System. *IEEE International Conference on Smart Technologies for Smart Nation*, 2021.

[7] Remote Sensing for Agriculture in the Era of Industry 5.0 – A Survey. *IEEE Geoscience and Remote Sensing Magazine*, 2023.

[8] A Deep Learning Image Augmentation Method for Field Agriculture. *IEEE Transactions on Computational Agriculture*, 2024.