SMART AGROCRAFT : AI - DRIVEN AGRICULTURAL OPTIMIZATION USING CONVOLUTIONAL NEURAL NETWORKS

A PROJECT REPORT

Submitted by

ARAVINDHAN A	**6176AC21UCS008**
CHANDRU C	**6176AC21UCS014**
GOPINATH K	**6176AC21UCS035**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING

(An Autonomous Institution)

DR M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2025

SMART AGROCRAFT : AI - DRIVEN AGRICULTURAL OPTIMIZATION USING CONVOLUTIONAL NEURAL NETWORKS

A PROJECT REPORT

Submitted by

ARAVINDHAN A	**6176AC21UCS008**
CHANDRU C	**6176AC21UCS014**
GOPINATH K	**6176AC21UCS035**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING

(An Autonomous Institution)

DR M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY:: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report " **SMART AGROCRAFT : AI - DRIVEN AGRICULTURAL OPTIMIZATION USING CONVOLUTIONAL NEURAL NETWORKS** " is the Bonafide work of " **ARAVINDHAN A (6176AC21UCS008) , CHANDRU C (6176AC21UCS014) , GOPINATH K (6176AC21UCS035) "** who carried out the project under my supervision.

SIGNATURE

Dr. G. FATHIMA, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR,
Department of CSE,
Adhiyamaan College of Engineering,
(An Autonomous Institution),
Dr. M.G.R. Nagar,
Hosur-635 130.

SIGNATURE

Mrs. R.PRAMILA, ME.,
SUPERVISOR
ASSISTANT PROFESSOR,
Department of CSE,
Adhiyamaan College of Engineering,
(An Autonomous Institution),
Dr. M.G.R. Nagar,
Hosur-635 130.

Submitted for the project VIVA-VOCE Examination held on _____at **Adhiyamaan College of Engineering (An Autonomous Institution), Hosur-635 130.**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Expressing profound gratitude, I wish to acknowledge the invaluable support and guidance I've received during the execution of my project at Adhiyamaan College of Engineering (An Autonomous Institution), Hosur.

We show our sincere thanks to **Dr. R. RADHAKRISHNAN, M.E., Ph.D.,** Principal. Adhiyamaan College of Engineering (An Autonomous Institution), Hosur for this valuable opportunity and environment given to us.

We extend our hearty gratitude to **Dr. G. FATHIMA, M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution) Hosur for her guidance and valuable suggestions and encouragement throughout this project.

We extend our gratitude to **Mrs. R.PRAMILA.,ME.,**Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution), Hosur whose immense support, encouragement and valuable guidance were responsible for completing the project successfully.

We extend our hearty thankfulness to our project coordinator and all our department faculty for their support in helping us complete this project successfully.

This project would not have been possible without the collective support and guidance of these individuals and the Adhiyamaan College of Engineering (An Autonomous Institution) community.

ABSTRACT

Smart Agro Craft is an innovative e-commerce platform designed to help farmers sell their vegetables online, integrating artificial intelligence to assess vegetable quality. Using a Convolutional Neural Network (CNN) model, the platform analyzes uploaded images of vegetables and classifies them as either "good quality" or "bad quality." This AI-driven quality assessment is stored in the backend database and displayed to customers, ensuring transparency and building trust between buyers and sellers. The use of machine learning in quality inspection reduces the need for manual checks, saving time and resources while minimizing quality disputes. Farmers benefit from having a system-verified record of their product's quality, which could lead to better prices and a wider customer base. By improving quality control, Smart Agro Craft helps create fairer pricing and operational efficiency, ultimately modernizing agricultural commerce through e-commerce convenience and intelligent automation.

KEYWORDS

CNN Model, Good/Bad Quality Classification, AI Integration, Product Insertion.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

XAMPP	CrossPlatform,Apache,MySQL,PHP,Perl
SQL	Structural Query Language
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
NDVI	Normalized Difference Vegetation Index
IOT	Internet Of Things
CNN	Convolution Neural Network
RF	Random Forest
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Smart AgroCraft is an AI-powered e-commerce platform designed to bring efficiency, reliability, and trust to the agricultural marketplace. By leveraging machine learning and deep learning, it automates vegetable and fruit quality assessment, ensuring that only premium-grade produce is listed for sale. Traditional agricultural markets often struggle with inefficiencies in quality evaluation, leading to buyer dissatisfaction, financial losses for farmers, and market instability. To overcome these challenges, Smart AgroCraft offers a technology-driven alternative, providing accurate, unbiased, and objective produce evaluation through artificial intelligence.

At the heart of Smart AgroCraft is a Convolutional Neural Network (CNN) model that processes images of vegetables and fruits submitted by farmers. It meticulously examines key quality parameters such as freshness, texture, color, and ripeness. The AI system then determines whether the produce meets the required quality benchmarks. Only those that pass the quality check are approved for sale, ensuring buyers receive high-quality products while reducing the risk of poor-quality or misrepresented produce.

A crucial component of Smart AgroCraft is its integration with a secure MySQL database, which organizes records of accepted and rejected products. This database offers insights that help farmers refine their cultivation and post-harvest handling techniques. Farmers also receive structured feedback on rejected produce, enabling them to make necessary improvements for future submissions. Over time, this iterative process enhances agricultural quality, promotes better farming practices, and strengthens market credibility.

The platform’s backend is built using Flask, ensuring a scalable and robust architecture capable of handling extensive image processing tasks efficiently. Its frontend, developed with HTML, CSS, and JavaScript, provides a seamless and user-

friendly experience for farmers and buyers. Farmers can easily upload images of their produce, while buyers can browse, filter, and purchase AI-approved vegetables and fruits with confidence.

To facilitate secure and efficient transactions, Smart AgroCraft integrates Instamojo for payment processing. This ensures hassle-free, swift, and secure payments, eliminating transaction delays and providing farmers with immediate access to earnings. By replacing cash-based transactions with a digital payment gateway, the platform enhances financial security and reduces risks associated with conventional trading methods.

Smart AgroCraft follows a structured workflow that streamlines the entire buying and selling process. Farmers capture and upload images of their vegetables and fruits. The AI-based CNN model evaluates the quality parameters and either approves or rejects the produce. Only approved products are listed on the e-commerce portal, where customers can browse and make purchases.

Payments are completed securely through Instamojo, and farmers receive, process, and dispatch orders to buyers. This approach reduces human intervention, minimizes errors, and ensures that only superior-quality agricultural products reach consumers. By eliminating the need for middlemen, Smart AgroCraft provides farmers direct access to a broad customer base, fostering a more transparent, efficient, and profitable agricultural ecosystem.

One of the platform's key impacts is its potential to reduce food waste. The agricultural industry frequently rejects produce with minor imperfections, even when they are still edible. Through AI-driven grading, Smart AgroCraft categorizes produce into different quality tiers, ensuring that even lower-grade products find suitable buyers instead of being discarded. This approach promotes sustainability in agriculture while maximizing profit margins for farmers by allowing them to generate revenue from produce that would otherwise go to waste.

Beyond its core functionalities, Smart AgroCraft offers numerous advantages over traditional agricultural trading methods. It brings greater transparency by enabling AI-driven pricing and quality evaluation, ensuring fair prices for consumers.

Automated listings and digital payments accelerate transactions, reducing delays. The AI-driven assessment fosters reliability and trust, making online agricultural trading a secure and predictable experience. Moreover, the platform is highly scalable, with the potential to expand its capabilities to include additional agricultural products in the future.

Future improvements for Smart AgroCraft include expanding the AI model to assess more crop varieties, such as grains and spices. Integrating blockchain technology will enhance supply chain transparency and traceability, strengthening the trustworthiness of the platform.

A mobile application will improve accessibility for farmers in remote areas, enabling them to participate more effectively in the digital marketplace. Additionally, enhancing AI models with advanced features like aroma detection and shelf-life prediction will refine the quality assessment process and add further value to both farmers and consumers.

By leveraging artificial intelligence to optimize market operations, minimize waste, and establish fair pricing, Smart AgroCraft is revolutionizing the agricultural e-commerce landscape. This AI-driven solution not only empowers farmers but also provides consumers with a reliable and efficient way to purchase high-quality agricultural products. Future improvements for Smart AgroCraft include expanding the AI model to assess more crop varieties, such as grains and spices. Integrating blockchain technology will enhance supply chain transparency and traceability, strengthening the trustworthiness of the platform.

Through the combination of machine learning, digital transactions, and user-friendly design, Smart AgroCraft is paving the way for a smarter, more sustainable, and equitable agricultural marketplace.

1.2 OBJECTIVE

- **Automate Quality Assessment** – Utilize a Convolutional Neural Network (CNN) model to evaluate vegetables and fruits based on freshness, texture, color, and ripeness, ensuring that only high-quality produce is listed for sale.

- **Empower Farmers** – Provide farmers with direct market access, eliminating middlemen and maximizing their profits while offering feedback on rejected produce to help them improve quality.
- **Enhance Buyer Confidence** – Ensure that consumers receive only AI-approved, premium-grade produce through an unbiased and transparent quality evaluation process.
- **Streamline the Trading Process** – Implement a structured workflow that enables farmers to upload images, get AI-based quality assessments, and list approved produce for buyers to purchase seamlessly.
- **Enable Secure Digital Transactions** – Integrate **Instamojo** for fast, safe, and hassle-free payments, ensuring smooth financial transactions between buyers and farmers.

CHAPTER 2

LITERATURE SURVEY

The digitization of agriculture and the integration of Artificial Intelligence (AI) into farming practices have garnered increasing attention in recent years. As the global demand for food production grows, there is an urgent need for intelligent systems that can enhance the efficiency, accuracy, and sustainability of agricultural operations.

Kiran Shinde[1] developed a web-based agricultural recommendation system that provides personalized guidance to farmers. By leveraging data mining techniques and historical agricultural data, the system generates tailored crop and fertilizer recommendations. This digital tool bridges the information gap between farmers and agricultural experts, enabling data-driven decision-making through a user-friendly online platform.

Vikas Kumar [2] introduced Krishi Mantra, a recommendation system based on Information and Communication Technology (ICT). The system facilitates expert-farmer interaction by analyzing climatic data, soil characteristics, and geographical parameters. By incorporating semantic web technologies, the model ensures that recommendations are context-aware and relevant, thus supporting smarter and more sustainable agricultural planning.

Mansi Shinde [3] proposed a mobile-based application that not only provides crop and fertilizer recommendations but also integrates a purchasing feature. This system uses prior farming patterns and environmental conditions to determine suitable inputs. By allowing farmers to procure recommended fertilizers directly through the app, the solution streamlines agricultural supply chains and reduces the time between decision-making and implementation.

Vikas Kumar [4] developed a mobile-enabled crop cultivation information system that delivers real-time agricultural knowledge and assistance. The focus of their system is to make critical farming insights accessible to small-scale farmers, thereby improving crop yield and reducing dependency on manual advisory

services. Emphasis is placed on intuitive interfaces to promote widespread adoption in rural areas.

Cheng-Lin Liu [5] conducted a comparative benchmarking study on handwritten digit recognition using advanced classification algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNN). Their analysis highlighted the superior accuracy of CNNs in image classification tasks due to their ability to learn spatial hierarchies and deep features. The study further established CNNs as suitable models for visual recognition tasks in agriculture, such as identifying crop types or diseases from images.

Mahmoud M. Abu Ghosh [6] examined the performance of different deep learning architectures Deep Neural Networks (DNN), Deep Belief Networks (DBN), and CNNs using the MNIST dataset. While DNNs showed the highest accuracy, CNNs stood out for their balance between speed and precision, making them a reliable choice for real-time agricultural image analysis applications, especially in resource-constrained environments.

Sharath D. M. and Rohan M. G. [7] proposed an image-processing-based approach for detecting diseases in pomegranate crops. Their method relies on visual attributes such as color variation and edge details to identify signs of infection. Built using Python and OpenCV, the system captures and processes images via mobile devices, offering farmers a portable and low-cost solution for timely disease detection and prevention.

M. Pushpavalli [8] introduced an automated fruit grading system that applies image processing techniques to assess quality parameters such as color, size, and shape. This approach facilitates fast and accurate sorting of fruits, reducing reliance on manual inspection and helping maintain consistent standards in agricultural produce. The implementation of such systems can significantly enhance the efficiency of food processing and supply chains by minimizing human error and labor costs.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The traditional agricultural market operates through physical marketplaces, local vendors, and intermediaries who connect farmers with consumers. Farmers typically sell their produce to wholesalers, who then distribute it to retailers or final consumers. In some cases, farmers sell directly to local markets, but this involves extensive manual effort, logistical challenges, and price negotiations. The process of quality assessment is largely subjective, as it relies on human inspection, leading to inconsistencies and unfair rejection of produce. Without a standardized grading system, both farmers and buyers lack confidence in the fairness of transactions. Additionally, most transactions are cash-based, posing financial risks such as delayed payments, theft, fraud, and financial mismanagement. This inefficient system limits farmers' profitability, creates market instability, and increases food wastage, highlighting the need for a more transparent, technology-driven solution like Smart AgroCraft to streamline the agricultural supply chain and enhance trust between farmers and consumers.

DISADVANTAGES

- Quality checks are performed manually by middlemen or traders, leading to inconsistencies and possible bias in product selection. Buyers often receive subpar or misrepresented produce.
- Farmers often rely on intermediaries to sell their produce, who take a significant cut of the profits, leaving farmers with lower earnings.
- Farmers are often restricted to selling within local markets, reducing their ability to expand their customer base and secure better prices.
- Prices fluctuate based on demand, intermediaries' influence, and lack of transparency in transactions. Farmers often receive unfair pricing, leading to financial losses.

- Due to inefficient logistics, lack of proper storage facilities, and market fluctuations, a large quantity of produce goes to waste, affecting both farmers and consumers.
- The reliance on cash transactions or delayed payments from wholesalers and retailers affects farmers' financial stability and liquidity.
- Farmers need to physically transport produce to markets, negotiate prices, and manage sales, making the process labor-intensive and inefficient.
- Buyers lack access to detailed product information, making it difficult to verify quality and authenticity. This reduces trust in online or remote transactions.

3.2 PROPOSED SYSTEM

Smart AgroCraft is an AI-powered e-commerce platform designed to overcome the inefficiencies of traditional agricultural markets by leveraging machine learning and deep learning for automated quality assessment of vegetables and fruits. The platform integrates an advanced Convolutional Neural Network (CNN) model that analyzes crucial quality parameters such as freshness, texture, color, and ripeness from images uploaded by farmers. Based on this assessment, only high-quality produce is approved for sale, ensuring buyers receive premium-grade products while reducing the risks associated with misrepresented agricultural goods. A secure MySQL database stores records of accepted and rejected products, providing farmers with structured feedback to improve their cultivation and post-harvest handling techniques. The platform is built using Flask for a scalable backend and features a simple yet intuitive frontend using HTML, CSS, and JavaScript, allowing easy navigation for both farmers and buyers. Transactions are facilitated through Instamojo, enabling seamless digital payments and eliminating cash-based financial risks. By directly connecting farmers with buyers, Smart AgroCraft removes the need for middlemen, enhances transparency, and fosters an efficient and trustworthy agricultural marketplace.

ADVANTAGES

- The CNN model ensures unbiased, accurate, and consistent quality evaluation of produce, reducing human errors and subjectivity.
- Direct farmer-to-consumer transactions eliminate intermediaries, ensuring fair pricing and improved profits for farmers.
- Instamojo integration provides a secure digital payment system, preventing fraud and delays associated with cash-based transactions.
- A simple and intuitive frontend allows seamless product uploads by farmers and effortless browsing and purchasing by consumers.
- AI-driven grading and pricing create a reliable and transparent agricultural trading ecosystem.

3.3 PROPOSED SOLUTION

Smart AgroCraft is an AI-powered e-commerce platform designed to modernize and streamline agricultural trading by addressing the inefficiencies of traditional marketplaces. The proposed system utilizes machine learning and deep learning to automate the quality assessment of vegetables and fruits, ensuring that only premium-grade produce is listed for sale. By integrating artificial intelligence into the agricultural supply chain, Smart AgroCraft enhances transparency, improves pricing accuracy, and empowers farmers with direct market access, eliminating the need for intermediaries.

At the core of the platform is an advanced Convolutional Neural Network (CNN) model that evaluates produce quality based on parameters such as freshness, texture, color, and ripeness. Farmers upload images of their produce, and the AI system processes these images to determine whether the produce meets the necessary quality benchmarks. Only approved products are listed on the e-commerce platform, ensuring that buyers receive high-quality agricultural goods.

Additionally, rejected products are recorded in a secure MySQL database, providing structured feedback to farmers to help them enhance their farming and post-harvest handling practices.

The backend of Smart AgroCraft is built using Flask, ensuring a scalable, efficient, and high-performance architecture capable of handling large-scale image processing tasks. The frontend, developed using HTML, CSS, and JavaScript, offers a simple yet intuitive interface for both farmers and buyers. Farmers can easily upload images of their produce, while buyers can browse, filter, and purchase high-quality AI-approved products.

To facilitate secure and hassle-free financial transactions, Smart AgroCraft integrates Instamojo, a trusted payment gateway. This digital payment system eliminates cash-based transactions, reducing financial risks and enabling farmers to receive payments instantly. The platform follows a structured workflow where farmers upload produce images, the AI model assesses quality, accepted products are listed for sale, buyers browse and make purchases, and farmers fulfill orders efficiently.

Beyond its core functionalities, Smart AgroCraft contributes to reducing food waste by categorizing produce into different quality tiers, allowing even lower-grade products to find buyers instead of being discarded. The platform also enhances market credibility by ensuring fair pricing and scalability, with future plans to incorporate additional agricultural products, blockchain for supply chain transparency, and a mobile application for better accessibility.

By leveraging artificial intelligence, digital payments, and a user-friendly interface, Smart AgroCraft offers a comprehensive and technologically advanced solution to transform agricultural e-commerce, benefiting both farmers and consumers while promoting a sustainable and equitable marketplace.

3.4 IDEATION & BRAINSTORMING

The development of Smart AgroCraft began with an extensive ideation and brainstorming phase, where key challenges in traditional agricultural trading were identified and analyzed in depth. The primary objective was to create a technology driven agricultural marketplace that enhances efficiency, transparency, and fairness in the trading of vegetables and fruits. Traditional markets rely heavily on human involvement in quality assessment, which often leads to inconsistencies, inefficiencies, and subjectivity in evaluating agricultural products. Farmers are largely dependent on intermediaries and middlemen, which significantly reduces their profit margins and creates pricing inefficiencies. Additionally, the existing system predominantly involves cash-based transactions, which introduce financial security risks and cause delays in payment processing. Farmers also struggle with limited market reach, as they are confined to selling their produce in local markets or through wholesalers, restricting their ability to maximize earnings.

To address these pressing challenges, Smart AgroCraft was conceptualized as an AI-powered automated quality assessment and e-commerce platform specifically designed for the agricultural sector. The core innovation of this platform is the integration of Convolutional Neural Networks (CNNs) to analyze images of vegetables and fruits based on key quality parameters such as freshness, texture, color, and ripeness. Unlike traditional manual inspections, which can be influenced by human bias and inconsistencies, the AI-driven quality evaluation ensures that only premium-grade produce is approved for sale. This eliminates subjectivity, enhances reliability, and builds trust among buyers who can confidently purchase products with assured quality. The automated verification process also saves farmers significant time and effort, allowing them to focus more on cultivation and production rather than navigating complex quality inspection procedures.

Furthermore, Smart AgroCraft eliminates the reliance on middlemen by implementing a direct farmer-to-consumer trading model. Farmers can independently list their produce on an intuitive and interactive e-commerce portal, where buyers can

browse and purchase high-quality AI-approved products. This direct selling model not only enhances the earnings of farmers but also ensures that buyers get fresh produce at fair prices without inflated costs due to intermediary involvement. Secure and seamless financial transactions are facilitated through Instamojo, a reliable payment gateway that enables instant digital transactions. This shift from traditional cash-based trading to secure online payments mitigates risks associated with delayed or fraudulent transactions, ensuring that farmers receive their earnings promptly and without unnecessary hurdles.

Additionally, Smart AgroCraft provides valuable data-driven insights to farmers, empowering them with structured feedback on why their produce was rejected and how they can improve their cultivation and post-harvest handling techniques. By incorporating data analytics and machine learning, the platform enables farmers to enhance their production quality over time, leading to greater profitability and improved market credibility. The iterative feedback system ensures continuous learning and adaptation, fostering a culture of quality improvement and efficiency within the agricultural industry.

As the ideation phase progressed, it became essential to design an architecture that supports scalability, efficiency, and reliability. After multiple brainstorming sessions and refinements, Smart AgroCraft was structured with a Flask-based backend, ensuring a robust and scalable system capable of handling extensive image processing tasks and real-time quality assessments. A MySQL database was integrated to store and manage structured records of all accepted and rejected products, providing a well-organized data repository for farmers and buyers. The frontend was designed using HTML, CSS, PHP and JavaScript, delivering an interactive and user-friendly experience that simplifies navigation for both farmers and consumers. The interface allows farmers to easily upload produce images, while buyers can browse, filter, and purchase AI-approved vegetables and fruits with complete confidence in the quality and authenticity of their purchases.

By leveraging machine learning, AI-powered quality evaluation, digital transactions, and a structured e-commerce model, Smart AgroCraft stands as a revolutionary solution that redefines agricultural trading. The platform not only streamlines market operations but also empowers farmers with better financial control, minimizes food waste through smart grading systems, and establishes a transparent, technology-driven agricultural ecosystem. Through continuous innovation and refinement, Smart AgroCraft is paving the way for a more efficient, sustainable, and inclusive agricultural marketplace that benefits farmers and consumers alike.

3.5 PROBLEM SOLUTION FIT

In the context of Smart AgroCraft, addressing the inefficiencies in traditional agricultural markets and implementing innovative solutions is crucial to achieving a meaningful problem-solution fit. Below is an analysis of key challenges and how Smart AgroCraft effectively resolves them.

INCONSISTENT QUALITY ASSESSMENT

Traditional agricultural markets rely on manual inspection methods to assess the quality of fruits and vegetables. These assessments are often subjective, inconsistent, and prone to human error, leading to disputes, financial losses, and unfair trade practices. Smart AgroCraft eliminates this issue by integrating a Convolutional Neural Network (CNN) model, which performs an AI-powered quality assessment. By analyzing critical parameters such as freshness, texture, color, and ripeness, the model provides an accurate, unbiased, and standardized evaluation of produce quality. This ensures that only premium-grade products are listed for sale, enhancing buyer confidence and preventing quality-related disputes.

INCONSISTENT QUALITY ASSESSMENT

Traditional agricultural markets rely on manual inspection methods to assess the quality of fruits and vegetables. These assessments are often subjective, inconsistent, and prone to human error, leading to disputes, financial losses, and unfair trade practices. Smart AgroCraft eliminates this issue by integrating a Convolutional Neural Network (CNN) model, which performs an AI-powered quality assessment. By analyzing critical parameters such as freshness, texture, color, and ripeness, the model provides an accurate, unbiased, and standardized evaluation of produce quality. This ensures that only premium-grade products are listed for sale, enhancing buyer confidence and preventing quality-related disputes.

DEPENDENCY ON MIDDLEMEN

In traditional markets, farmers rely on intermediaries to sell their produce, often resulting in low profit margins due to unfair pricing and commissions. These middlemen dictate market rates, leading to financial exploitation of farmers and inflated costs for consumers. Smart AgroCraft disrupts this model by enabling farmers to directly connect with buyers through a user-friendly e-commerce platform. This direct trade model ensures that farmers receive fair compensation, while consumers get fresh produce at competitive prices, fostering transparency and trust in the supply chain.

CASH-BASED TRANSACTIONS AND PAYMENT DELAYS

Conventional agricultural trade relies heavily on cash transactions, which often lead to payment delays, financial insecurity, and fraud risks. Farmers frequently experience delayed settlements from wholesalers, impacting their cash flow and sustainability. To solve this, Smart AgroCraft integrates Instamojo, a secure digital payment gateway, ensuring that transactions are processed instantly. This eliminates the risks of delayed payments, provides financial security to farmers, and simplifies transactions for buyers, making the process seamless and efficient.

FOOD WASTE DUE TO REJECTION OF PRODUCE

A significant issue in agriculture is the rejection of produce due to minor imperfections, leading to substantial food wastage. In traditional markets, buyers often discard products that do not meet aesthetic standards, even if they are nutritionally intact and edible. Smart AgroCraft addresses this by implementing an AI-driven grading system that categorizes produce into different quality tiers. This ensures that even lower-grade products find suitable buyers, thereby reducing food waste and maximizing farmers' earnings.

LIMITED ACCESS TO TECHNOLOGY FOR FARMERS

Many farmers in rural areas lack access to digital trading platforms, limiting their ability to reach a wider customer base. Traditional marketplaces restrict farmers to local buyers, reducing their sales opportunities. Smart AgroCraft enhances accessibility by offering an intuitive online platform where farmers can easily upload images of their produce, list products, and manage orders. Future plans to develop a mobile application will further improve ease of use and market penetration, allowing farmers from even the most remote areas to benefit from AI-driven trading.

LACK OF TRANSPARENCY AND FAIR PRICING

Pricing in traditional agricultural markets is often influenced by intermediaries, leading to price manipulation and unfair trade practices. Farmers are frequently underpaid for their produce, while buyers may end up paying inflated prices. Smart AgroCraft ensures price transparency by allowing farmers to set competitive prices based on AI-verified quality assessments. The platform also enables buyers to make informed purchasing decisions, promoting fairness and stability in agricultural trade.

3.6 ARCHITECTURE

Due The architecture of Smart AgroCraft is thoughtfully designed as a modular, scalable, and efficient system that seamlessly integrates multiple components to offer a reliable AI-powered agricultural e-commerce platform. At the forefront of this architecture lies the user interface (UI) layer, which is developed using modern web technologies such as HTML, CSS, and JavaScript. This intuitive, visually appealing, and responsive interface is tailored to accommodate two types of users—farmers and buyers. Farmers use the platform to upload high-quality images of their produce with ease, while buyers are provided with smart browsing capabilities to explore and purchase only AI-approved fruits and vegetables, ensuring confidence in their purchase decisions.

The backend layer forms the core of the platform's logic and is implemented using the Python-based Flask web framework. This layer is responsible for handling essential application functionalities, including managing user authentication, routing uploaded images to the AI model, integrating with the database, and securely processing payments. The centerpiece of this backend system is a highly trained Convolutional Neural Network (CNN) model. This model is specifically designed to process produce images and assess various quality indicators such as freshness, color uniformity, ripeness level, and surface texture. Based on the assessment, the AI model determines if the submitted product meets the quality standards defined by the platform.

Only the vegetables and fruits classified as “Good Grade” by the CNN model are permitted to be inserted into the database and listed on the marketplace for buyers to view. If the model detects that the produce fails to meet the minimum standards, it is automatically rejected, and the system logs a structured feedback message explaining the reasons for rejection. This feedback is valuable to farmers as it helps them improve future submissions and refine their farming and post-harvest practices over time.

Supporting all these processes is a secure and robust MySQL database, which serves as the foundation for data storage and retrieval. This database meticulously maintains user credentials, product information, AI quality assessment scores, feedback for rejected items, and complete transaction histories. It ensures that all operations on the platform are tracked and accessible for both users and administrators, thereby enhancing transparency and accountability.

To facilitate smooth and secure monetary transactions, Instamojo is integrated into the backend as a digital payment gateway. This integration allows for seamless, real-time financial interactions between buyers and farmers, replacing traditional cash-based systems. The result is a streamlined, trustworthy, and faster payment process, where farmers can access their earnings without delays.

Altogether, the architecture of Smart AgroCraft supports the full workflow from image upload and AI-driven quality verification to product listing, purchase by the consumer, and order fulfillment by the farmer. Each layer and module works cohesively to provide a transparent, intelligent, and reliable agricultural trading ecosystem that benefits all stakeholders and moves the sector toward smarter, technology-enhanced practices.

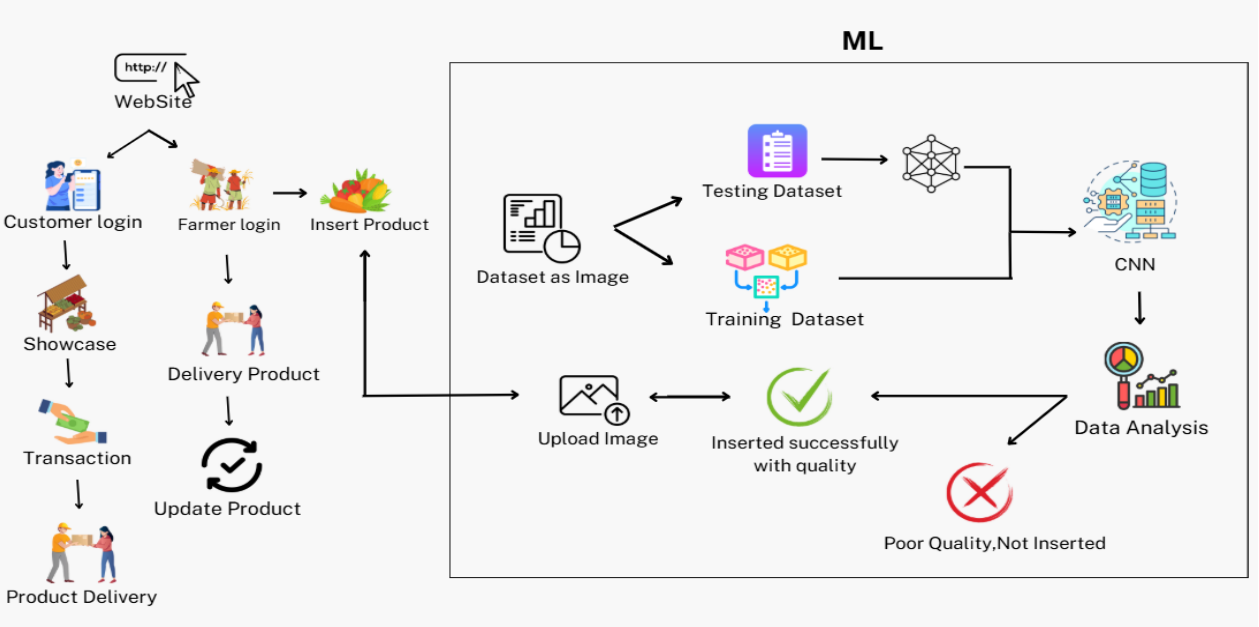

Fig 3.6 Model Architecture

3.7 DATAFLOW DIAGRAM

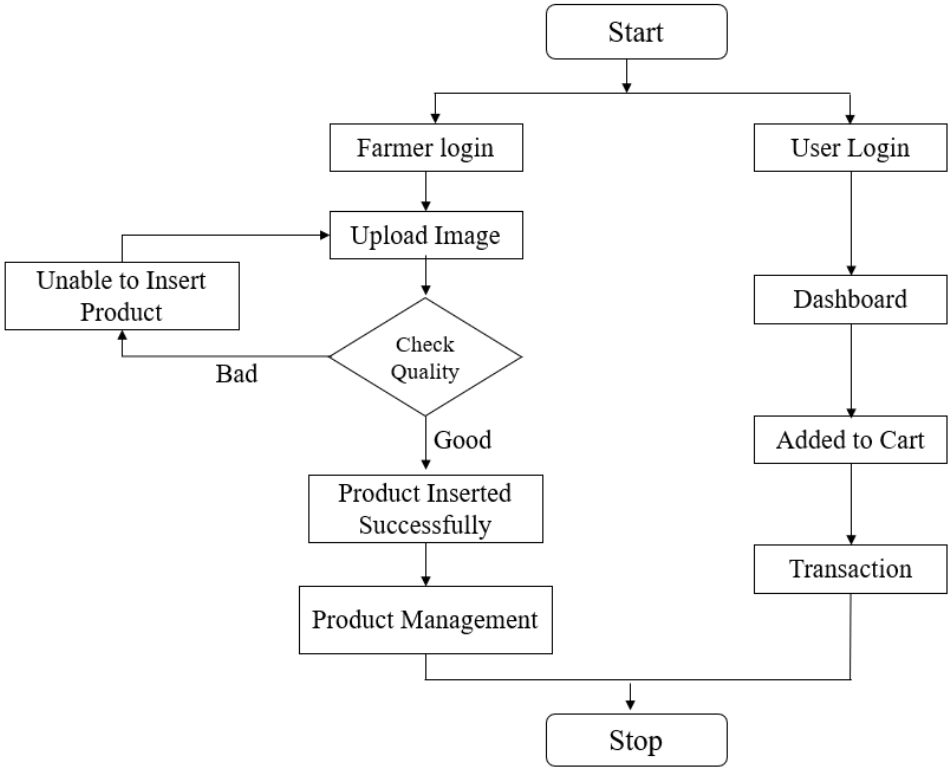

Fig 3.7 Dataflow Diagram

The data flow of the system begins with either a farmer or a user logging into the platform. When a farmer logs in, they upload an image of the vegetable product they want to sell. This image is then processed using a trained CNN model to check its quality. If the vegetable is classified as having bad quality, the system prevents the product from being inserted and displays an "Unable to Insert Product" message. If the quality is good, the product is successfully added to the system and enters the product management phase where farmers can handle stock, pricing, and other details.

On the other side, when a user logs in, they are directed to a dashboard where they can view available products. Users can add desired items to their cart, and once satisfied, proceed with a transaction to complete the purchase. After either product management by the farmer or successful transaction by the user, the process ends. This flow ensures only good-quality vegetables are listed, enhancing reliability and

trust in the platform.

3.8 MODULES

1. User Management Module
2. Product Management Module
3. CNN-Based Quality Analysis Module
4. Payment Integration Module

3.8.1 User Management Module

The User Management Module forms the backbone of the Smart AgroCraft system, providing a secure and role-specific access environment for three types of users: farmers, industrial buyers, and administrators. This module allows users to register with unique credentials and log in based on their designated roles. Farmers can upload their products, buyers can browse and purchase items, and administrators have the authority to manage users, verify product quality data, and oversee system activity. The module ensures that each user is given appropriate privileges, maintaining data security and system integrity. It also includes functionality for profile management, password changes, and user activity tracking, creating a seamless user experience tailored to different access levels.

3.8.2 Product Management Module

The Product Management Module enables farmers to upload and manage their vegetable listings, facilitating the digital sale of high-quality agricultural produce. Farmers can input essential product details such as name, quantity, pricing, and upload images, which are then evaluated by the CNN-based quality analysis module. Only the products classified as good in quality are listed for purchase. Buyers can easily browse through the verified products, view details, filter based on preferences, and add them to the cart for checkout. The module also allows farmers to update or delete listings, and ensures a streamlined, transparent product lifecycle from upload to order fulfillment.

3.8.3 CNN-Based Quality Analysis Module

The CNN-Based Quality Analysis Module is an intelligent feature that leverages a trained Convolutional Neural Network (CNN) model to assess the quality of vegetables uploaded by farmers. When an image is submitted during product entry, this module is activated to classify the vegetable as either “Good” or “Poor” quality. Only products classified as “Good” are allowed to proceed to the product listing page, ensuring buyers are offered only the best produce. This automated quality-checking system eliminates manual inspection errors, enhances trust among customers, and upholds quality assurance. It plays a vital role in maintaining product standards and supports farmers by ensuring that only marketable goods are showcased on the platform.

3.8.4 Payment Integration Module

The Payment Integration Module handles all financial transactions between buyers and farmers, enabling a secure and efficient payment process using the Instamojo payment gateway. After selecting and confirming their purchases, buyers are redirected to the payment interface, where they can complete the transaction using various modes such as credit/debit cards, UPI, or net banking. Upon successful payment, the system updates the order status and sends a digital confirmation receipt to both the buyer and the farmer. This module not only facilitates real-time payment processing but also ensures transparency, reduces cash-related risks, and maintains a detailed record of all transactions for administrative monitoring and reporting.

CHAPTER 4

SYSTEM REQUIREMENT

4.1 HARDWARE REQUIREMENTS

Processor	:64-bit, Quad-core
RAM	:18 GB
Hard Disk	:250 GB
Monitor	:Minimum 75Hz
Internet	:Minimum 1 Mbps
Graphics	:Basic GPU for CNN model processing

4.2 SOFTWARE REQUIREMENTS

Operating System	:Windows / Linux
Programming Language	:Python 3.10
Machine Learning	:TensorFlow
Web Development	:Flask Framework, HTML, CSS, JavaScript
Database	:MySQL
Payment Gateway	:Instamojo API Integration
IDE / Code Editor	:Visual Studio Code / PyCharm
Browser	:Chrome/M.Edge (latest version)

SOFTWARE DESCRIPTION

PYTHON 3.10

Python 3.10 is the primary programming language used in this project due to its readability, extensive library support, and strong community. It is particularly well-suited for artificial intelligence, machine learning, and web development, which are all essential components of Smart AgroCraft. With support for a wide range of frameworks and tools, Python streamlines the development of both backend logic and AI models in the system.

TENSORFLOW / KERAS

TensorFlow and Keras are the deep learning libraries used for building and training the Convolutional Neural Network (CNN) models. These models are responsible for analyzing and evaluating the quality of fruits and vegetables submitted by farmers. TensorFlow's performance optimization and Keras's ease of use allow for quick model experimentation and deployment, making them ideal for real-time image classification tasks.

FLASK

Flask is the lightweight Python web framework used for building the backend of Smart AgroCraft. It facilitates seamless integration between the AI models, user interfaces, and the MySQL database. Flask’s modularity allows rapid development of the platform’s features, such as user authentication, product upload, and image analysis, all while maintaining a clean and maintainable codebase.

MYSQL

MySQL serves as the primary relational database system for managing user details, product listings, transaction histories, and feedback data. It stores information about approved and rejected produce, enabling structured data retrieval, efficient query handling, and long-term data maintenance. The database also supports analysis and feedback mechanisms to help farmers improve product quality.

INSTAMOJO API

Instamojo is integrated as the secure payment gateway for all buyer transactions. It supports fast, cashless payments and gives farmers instant access to

their earnings. Its API is easy to implement within Flask, allowing for hassle-free checkout and robust transaction tracking. Instamojo enhances the trust and convenience of the e-commerce aspect of Smart AgroCraft.

HTML, CSS, & JAVASCRIPT

The frontend of the platform is designed using HTML for structure, CSS for styling, and JavaScript for interactivity. This trio ensures a responsive and user-friendly interface that allows farmers to upload images easily and buyers to browse and purchase AI-approved products. The simplicity of the design ensures smooth navigation across devices and user types.

OPENCV

OpenCV is used for pre-processing and manipulating images uploaded by farmers. It performs essential tasks such as resizing, normalization, and image enhancement before feeding them into the CNN model. This preprocessing ensures consistent input quality, which leads to higher model accuracy during prediction.

NUMPY & PANDAS

These data handling libraries are crucial for data manipulation and processing within the AI pipeline. NumPy supports mathematical operations on arrays, while Pandas is used to manage tabular data structures. Both libraries assist in handling backend operations, such as maintaining records, logging model outputs, and performing data analysis.

PREPROCESSING TOOLS

Data preprocessing is essential for both image and user input handling. Images undergo cleaning, resizing, normalization, and augmentation to improve model generalization. On the database end, user entries and product details are validated and filtered to ensure consistency and data quality. These tools ensure accurate predictions and smooth platform performance.

CHAPTER 5
IMPLEMENTATION

5.1 INPUT DESIGN

The input design of the Smart AgroCraft system is centered around simplicity, accuracy, and usability to ensure that both farmers and buyers can efficiently interact with the platform fig (5.1 Input page)while maintaining data integrity.

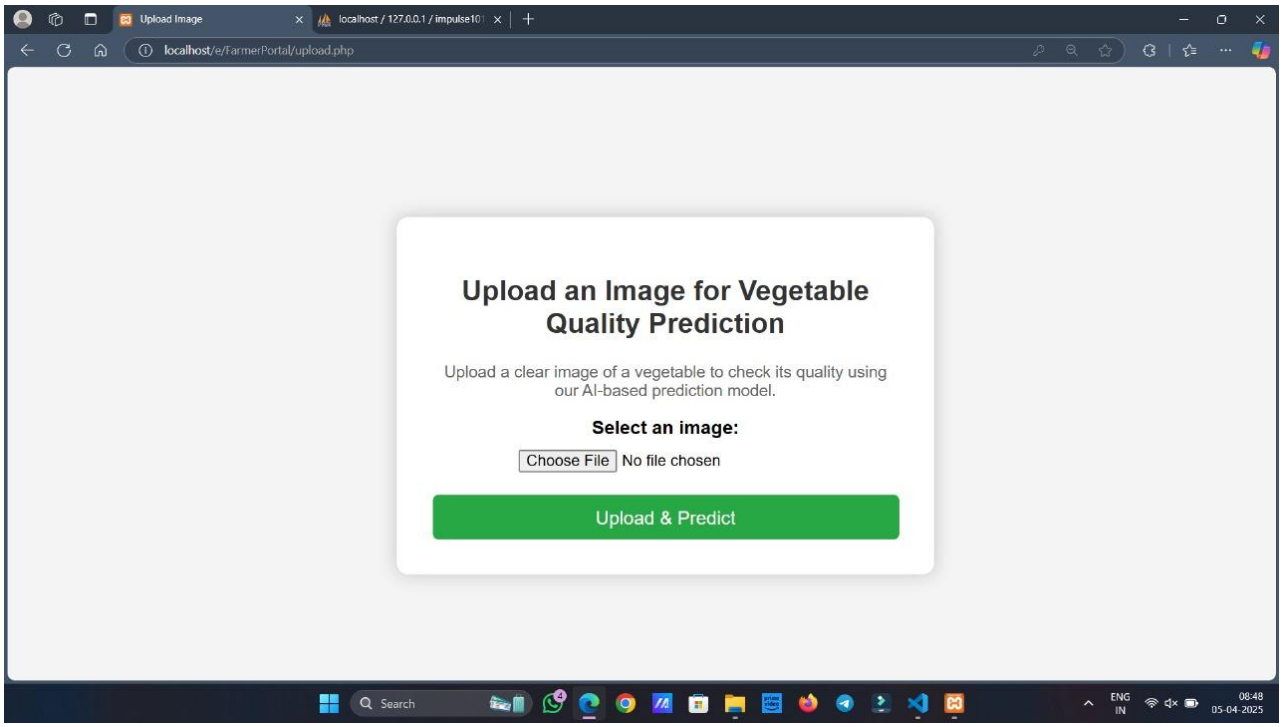

Fig 5.1 Input Page

USER AUTHENTICATION

Secure login functionality is implemented to differentiate between different types of users Farmers, Industrial Buyers, and Administrators. Each user type is granted access to features relevant to their role, ensuring a secure and role-based experience. Authentication involves validating username and password credentials, maintaining data confidentiality.

PRODUCT IMAGE UPLOAD

Farmers can upload images of their vegetables or fruits through a simple and intuitive form. These images are then sent to the backend, where a Convolutional

Neural Network (CNN) model analyzes the quality(fig 5.1 upload image). Only high-grade products are allowed for listing based on model prediction results.

VEGETABLE QUALITY

The system allows users to upload crop images for both quality verification and disease detection(fig 5.2 aler). These inputs are processed using the trained CNN model to predict whether the vegetable is Healthy, Diseased, or Substandard.

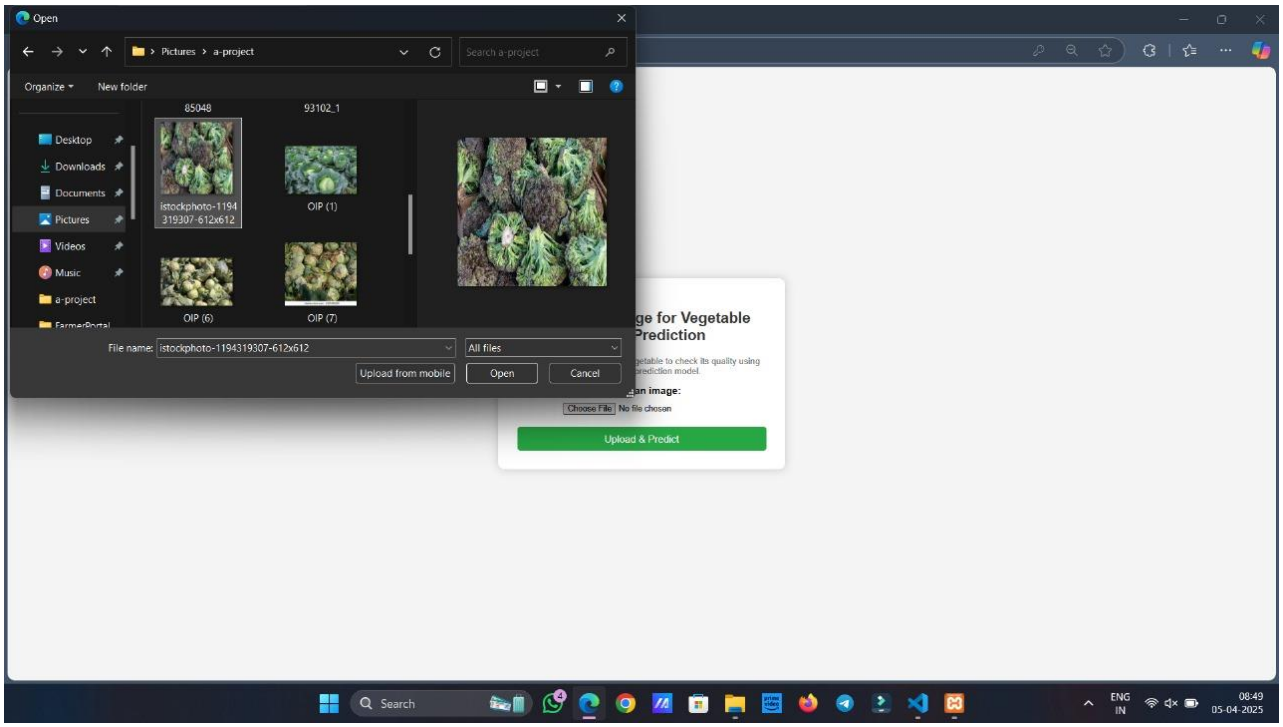

Fig 5.2 Upload Image(bad)

FARMER PRODUCT DETAILS INPUT

Along with the image, farmers provide necessary details such as vegetable name, quantity, price, and location. Form validations ensure correct formats and mandatory fields are filled to reduce user input errors.

ADMIN DASHBOARD INPUT

Admins can manage user records, verify product listings, monitor transactions, and update platform configurations. The admin input forms are designed for high efficiency and bulk management.

ERROR HANDLING

All input forms are equipped with frontend and backend validation. Proper

guidance is shown for missing or incorrect fields. Image formats, price inputs, and product details are all checked before submission to avoid errors during processing.

5.2 OUTPUT DESIGN

The output design of Smart AgroCraft ensures that both farmers and buyers receive timely, relevant, and clear information regarding product status, prediction results, and transactional feedback.

PRODUCT QUALITY PREDICTION DISPLAY

Once an image is uploaded, the CNN model evaluates the product and displays the output as either Good Grade (Accepted) or Poor Grade (Rejected). If the prediction is “Good,” the product is automatically listed in the marketplace (fig 5.3 Farmer Interface,uploaded Image).

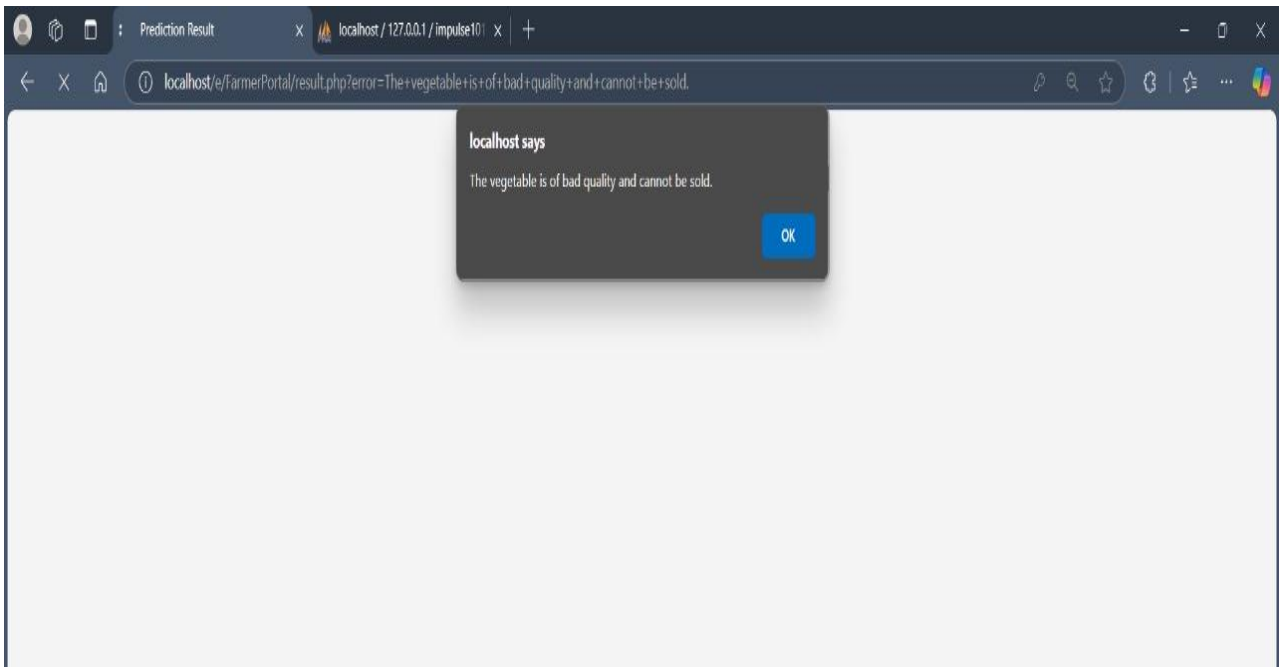

Fig 5.3 Result alert (bad)

DASHBOARD VISUALS FOR USERS

Farmers and buyers can view their dashboard containing product statuses, history of uploads, transaction details, and prediction logs. The data is presented in

tabular and card formats for easy understanding.

INTERACTIVE PRODUCT LISTING

Buyers see only verified and quality-passed products with clear photos, seller information, and pricing (Fig 5.2 Buyerhome page). They can add items to their cart, proceed to payment using Instamojo, and receive confirmation messages after purchase.

ADMINISTRATIVE OUTPUT

Admins receive alerts and logs of all product uploads, prediction results, and system activity. They can view rejected products, transaction summaries, and maintain platform-wide control.

5.3 CODE IMPLEMENTATION

Step 1: Set up the Login page

This PHP code implements a farmer login portal. It includes a styled HTML form that collects a phone number and password, then encrypts the password using AES encryption before checking it against a MySQL database. If the credentials match a record, the user is redirected to the farmer homepage.

Code Snippet

```
<?php
 include("../Includes/db.php");
 //session_start();
 include("../Functions/functions.php");
 ?>
 <!DOCTYPE html>
 <html>
 <head>
 <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Farmer Login portal</title>
<!-- <link rel="stylesheet" type="text/css" href="../Styles/FarmerLogin.css"> -
->
<script src="https://kit.fontawesome.com/c587fc1763.js"
crossorigin="anonymous"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js
"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></s
cript>
<a href="https://icons8.com/icon/83325/roman-soldier"></a>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src="https://kit.fontawesome.com/c587fc1763.js"
crossorigin="anonymous"></script>
<style></head>
<body>
<div class="card-header text-left" style="background-color:#292b2c">
<h4  style="font-style:bold;color:goldenrod;text-align:left">Login</h4>
</div>
<div class="card-body border border-dark">
<form name="my-form" action="FarmerLogin.php" method="post">
<div class="form-group row">
```

```
<label for="phone_number" class="col-md-4 col-form-label text-md-right"><i
class="fas fa-phone-alt mr-2"></i><b>Phone Number</b></label>
<div class="col-md-6">
<input type="text" id="phone_number" class="form-control border border-
dark" name="phonenumber" placeholder="Phone Number" required>
</div>
</div>
<div class="form-group row">
<label for="p1" class="col-md-4 col-form-label text-md-right"><i class="fas
fa-lock mr-2"></i><b>Password</b></label>
<div class="col-md-6">
<input id="p1" class="form-control border border-dark" type="password"
name="password" placeholder="Password" required>
</div>
<br>
<div class="col-md-6 offset-md-4">
<label id="forgotPassword"  class="text-left"><a id='link' style=""
href="FarmerForgotPassword.php"><b style="color:black ;text-align:left">
Forgot your password </b></a></label>
<br>
<label id="account" class="text-left"><a id='link'
href="FarmerRegister.php"><b style="color:black"> Create New
Account</b></a></label>
</main></body></html>
<?php
if (isset($_POST['login'])) {
$phonenumber = mysqli_real_escape_string($con, $_POST['phonenumber']);
$password = mysqli_real_escape_string($con, $_POST['password']);
$ciphering = "AES-128-CTR";
$iv_length = openssl_cipher_iv_length($ciphering);
```

```
$options = 0;
$encryption_iv = '2345678910111211';
$encryption_key = "DE";
$encryption = openssl_encrypt(
$encryption_key,
$options,
$encryption_iv
);
$query = "select * from farmerregistration where farmer_phone =
'$phonenumber' and farmer_password = '$encryption'";
echo $query;
$run_query = mysqli_query($con, $query);
$count_rows = mysqli_num_rows($run_query);
if ($count_rows == 0) {
echo "<script>alert('Please Enter Valid Details');</script>";
echo "<script>window.open('FarmerLogin.php','_self')</script>";
}
<br>
<div class="col-md-6 offset-md-4">
<label id="forgotPassword"  class="text-left"><a id='link' style=""
href="FarmerForgotPassword.php"><b style="color:black ;text-align:left">
Forgot your password </b></a></label>

while ($row = mysqli_fetch_array($run_query)) {
$id = $row['farmer_id'];
}
$_SESSION['phonenumber'] = $phonenumber;
echo
"<script>window.open('../FarmerPortal/farmerHomepage.php','_self')</script
}
```

Step 2: Insert product page

The insertproduct.php file allows farmers to add new products to the platform. It presents a form that collects details such as product title, stock, category, type, expiry date, price, description, keywords, and delivery availability. If an image is uploaded beforehand, its path is reused and moved to the correct directory. Upon submission, the product details, along with the associated farmer ID (retrieved from the session), are inserted into the database.

Code Snippet

```
<?php
include("../Includes/db.php");
session_start();
$sessphonenumber = $_SESSION['phonenumber'];
$imagePath = isset($_GET['file']) ? urldecode($_GET['file']) : '';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Insert Product</title>
</head>
<body>
    <h2>Insert Your Product</h2>
    <form action="insertproduct.php" method="post"
enctype="multipart/form-data">
        <label>Product Title:</label>
```

```
        <input type="text" name="product_title" required>
        <label>Product Stock (in kg):</label>
        <input type="text" name="product_stock" required>
        <label>Product Category:</label>
        <select name="product_cat" required>
            <option>Select a Category</option>
            <?php
            $get_cats = "SELECT * FROM categories";
            $run_cats = mysqli_query($con, $get_cats);
            while ($row_cats = mysqli_fetch_array($run_cats)) {
                echo "<option
value='{$row_cats['cat_id']}'>{$row_cats['cat_title']}</option>";
            }
            ?>
        </select>

        <label>Product Type:</label>
        <input type="text" name="product_type" required>
        <label>Product Expiry:</label>
        <input type="date" name="product_expiry" required>
        <!-- <label>Product Image:</label>
        <input type="file" name="product_image"> -->
        <?php if ($imagePath): ?>
            <p>Using uploaded image: <img src="<?php echo $imagePath;
?>" width="100"></p>
            <input type="hidden" name="uploaded_image" value="<?php
echo $imagePath; ?>">
        <?php endif; ?>
        <label>Product Price (Per kg):</label>
        <input type="text" name="product_price" required>
```

```
        <label>Product Description:</label>
        <textarea name="product_desc" rows="3" required></textarea>
        <label>Product Keywords:</label>
        <input type="text" name="product_keywords" required>
        <label>Delivery:</label>
        <input type="radio" name="product_delivery" value="yes"
required> Yes
        <input type="radio" name="product_delivery" value="no" required>
No
        <button type="submit" name="insert_pro">INSERT</button>
    </form>
</body>
</html>
        <label>Product Type:</label>
        <input type="text" name="product_type" required>
        <label>Product Expiry:</label>
        <input type="date" name="product_expiry" required>
        <!-- <label>Product Image:</label>
        <input type="file" name="product_image"> -->
        <?php if ($imagePath): ?>
            <p>Using uploaded image: <img src="<?php echo $imagePath;
?>" width="100"></p>
            <input type="hidden" name="uploaded_image" value="<?php
echo $imagePath; ?>">
        <?php endif; ?>
        <label>Product Price (Per kg):</label>
        <input type="text" name="product_price" required>


<?php
if (isset($_POST['insert_pro'])) {
```

```
    $product_title = $_POST['product_title'];
    $product_desc = $_POST['product_desc'];
    $product_keywords = $_POST['product_keywords'];
    $product_delivery = $_POST['product_delivery'];
    $product_image = ""; // Initialize empty
    if (!empty($_POST['uploaded_image'])) {
        // Image path received from upload.php (uploads/image.jpg)
        $old_image_path = $_POST['uploaded_image'];
        // Extract just the filename (e.g., image.jpg)
        $image_filename = basename($old_image_path);
        // Define the new destination
        $new_image_path = "../Admin/product_images/" . $image_filename;
        // Move the image to the correct directory
        if (rename($old_image_path, $new_image_path)) {
            $product_image = $image_filename; // Save relative path
        } else {
            echo "<script>alert('Error moving image!');</script>";
            exit();
        }
    } else {
        echo "<script>alert('No image provided!');</script>";
        exit();
    }
    // Ensure session is active
    if (isset($_SESSION['phonenumber'])) {
        $getting_id = "SELECT * FROM farmerregistration WHERE
farmer_phone = '$sessphonenumber'";
        $run = mysqli_query($con, $getting_id);
        $row = mysqli_fetch_array($run);
        $id = $row['farmer_id'];
```

```
        // Insert product into database with the updated image path
        $insert_product = "INSERT INTO products (farmer_fk,
 product_title, product_cat, product_type, product_expiry, product_image,
 product_stock, product_price, product_desc, product_keywords,
 product_delivery)
        I
 f (mysqli_query($con, $insert_product)) {
            echo "<script>alert('Product has been added
 successfully!');</script>";
            echo "<script>window.location.href =
 'farmerHomepage.php';</script>";
        } else {
            echo "<script>alert('Error inserting product!');</script>";
        }}
?>
```

5.4 RESULT

The Smart AgroCraft project successfully bridges the gap between traditional farming and modern digital commerce by providing an integrated platform that enhances the agricultural supply chain. Through the use of a CNN-based quality analysis system, the platform ensures that only high-quality vegetables are listed for sale, promoting consumer trust and farmer credibility. Farmers benefit from an easy-to-use interface to upload and manage their products, while buyers both individual customers and industrial clients gain access to verified, top-grade produce through a convenient online portal.(fig 5.5,6,7,8) The integration of a secure payment gateway further streamlines the transaction process, offering fast and transparent financial exchanges.

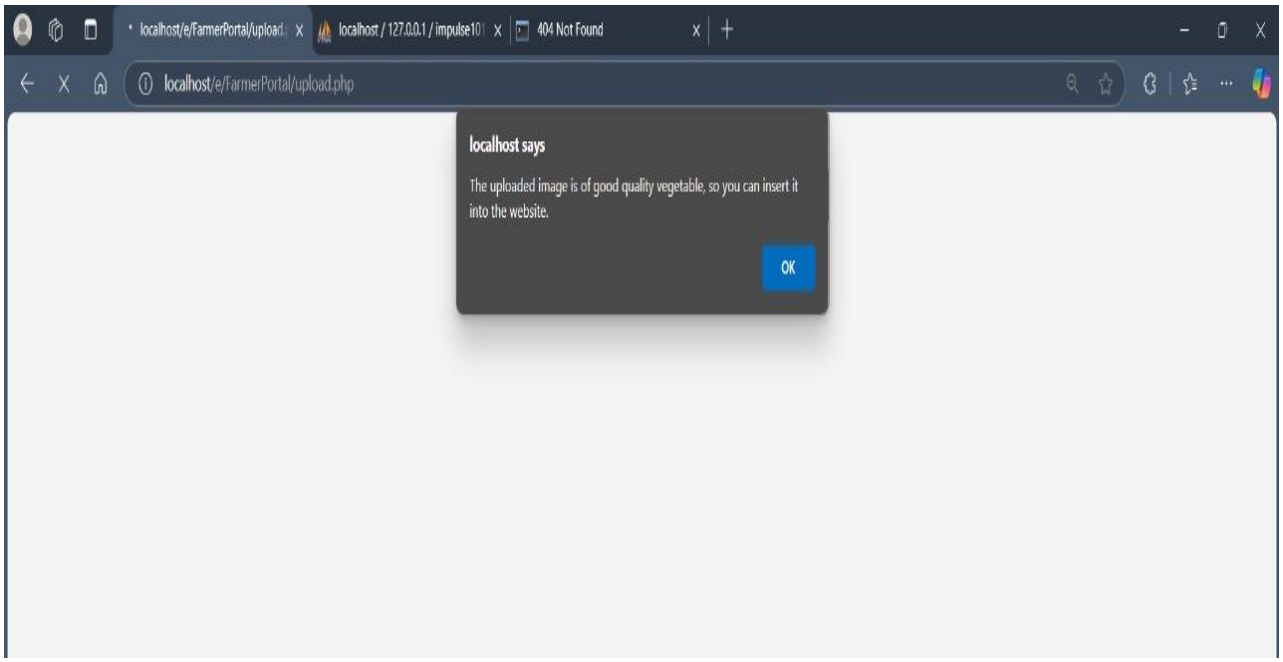

Fig 5.5 Result alert(Good)

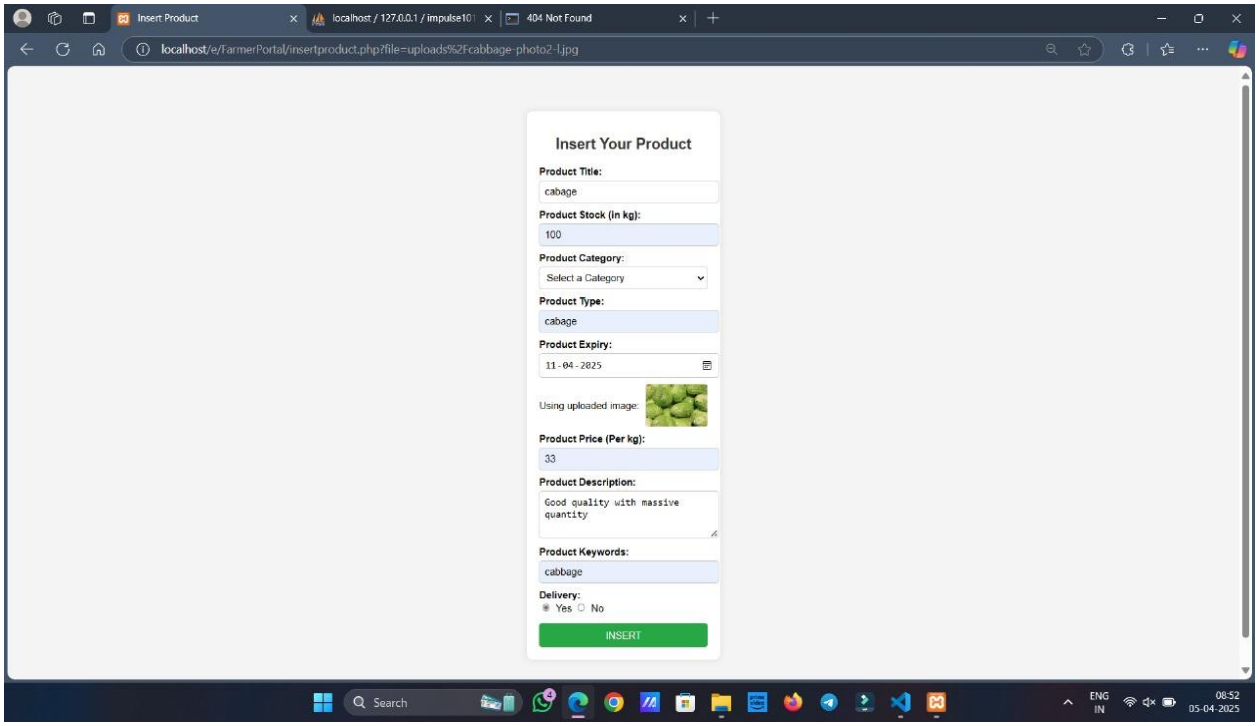

Fig 5.6 Insert Product

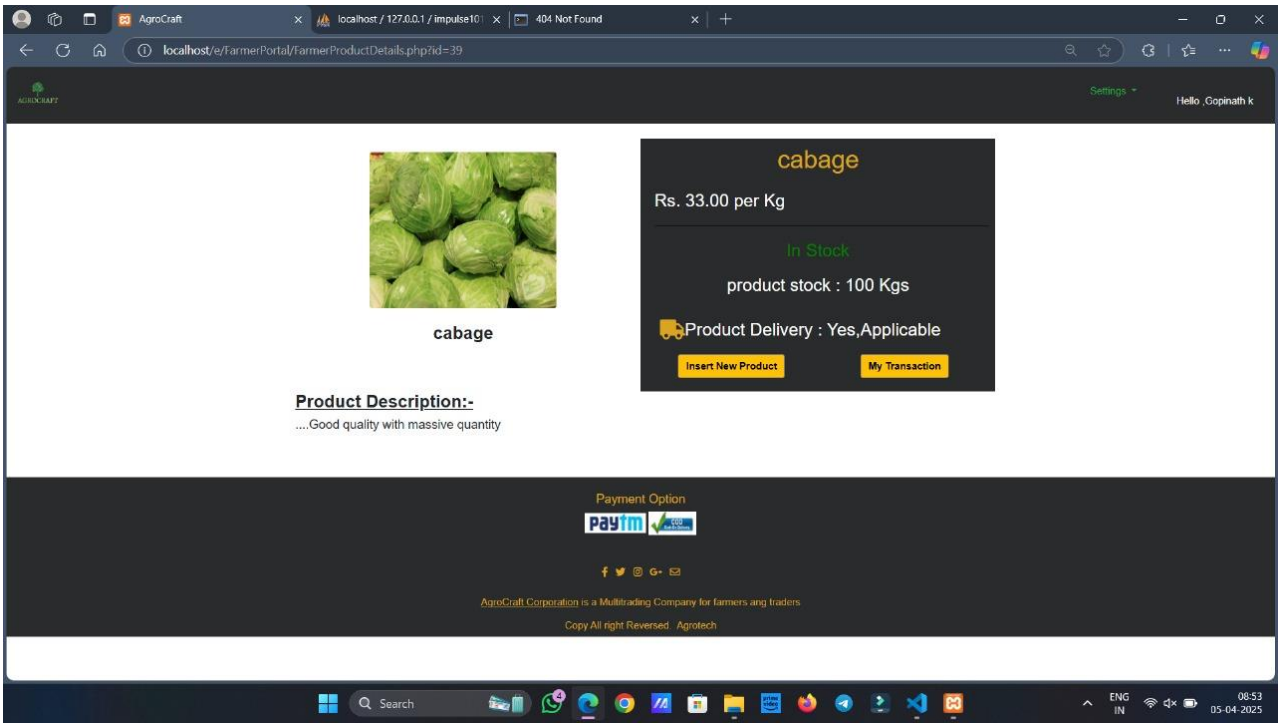

Fig 5.7 Uploaded Page

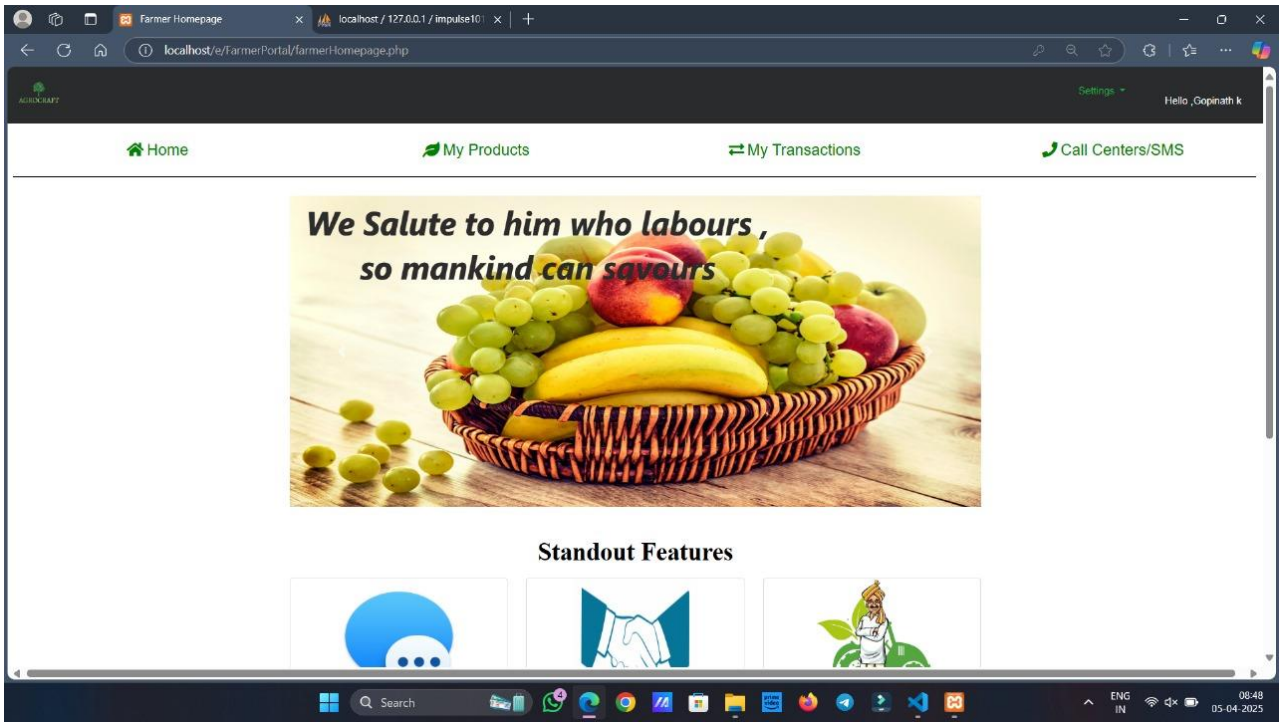

Fig 5.8 Farmer Interface

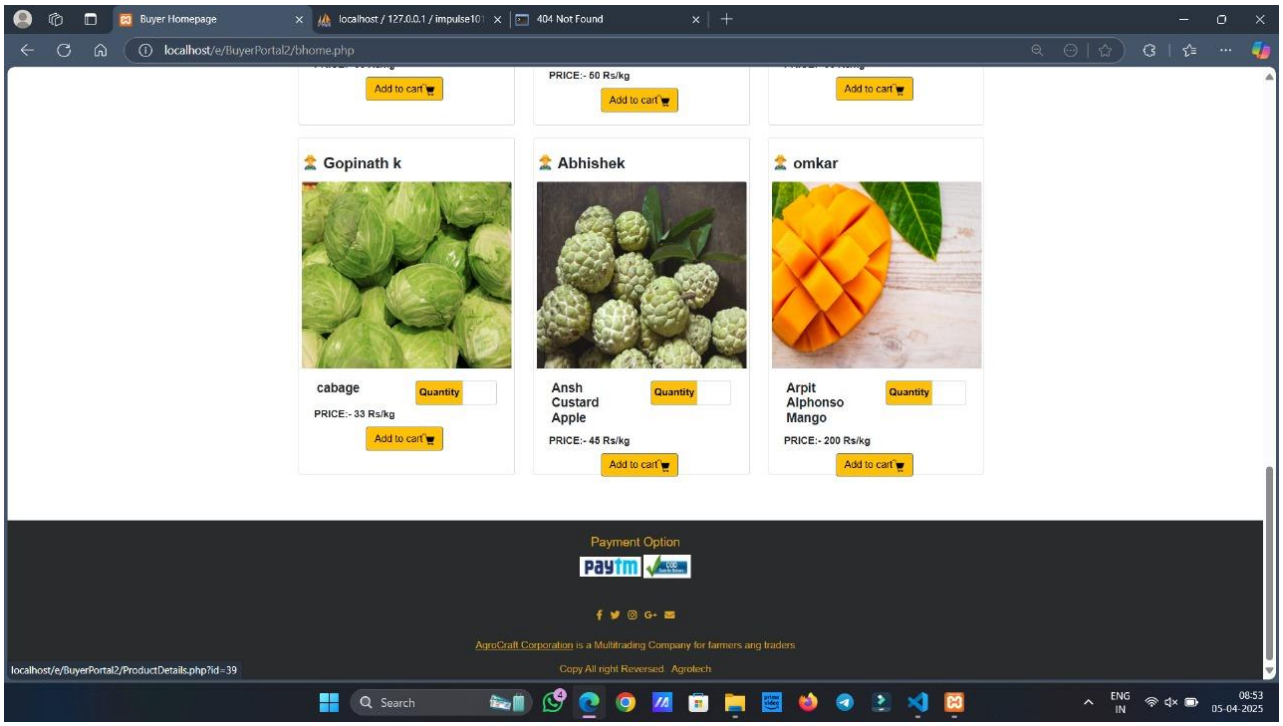

Fig 5.9 Buyerhome page

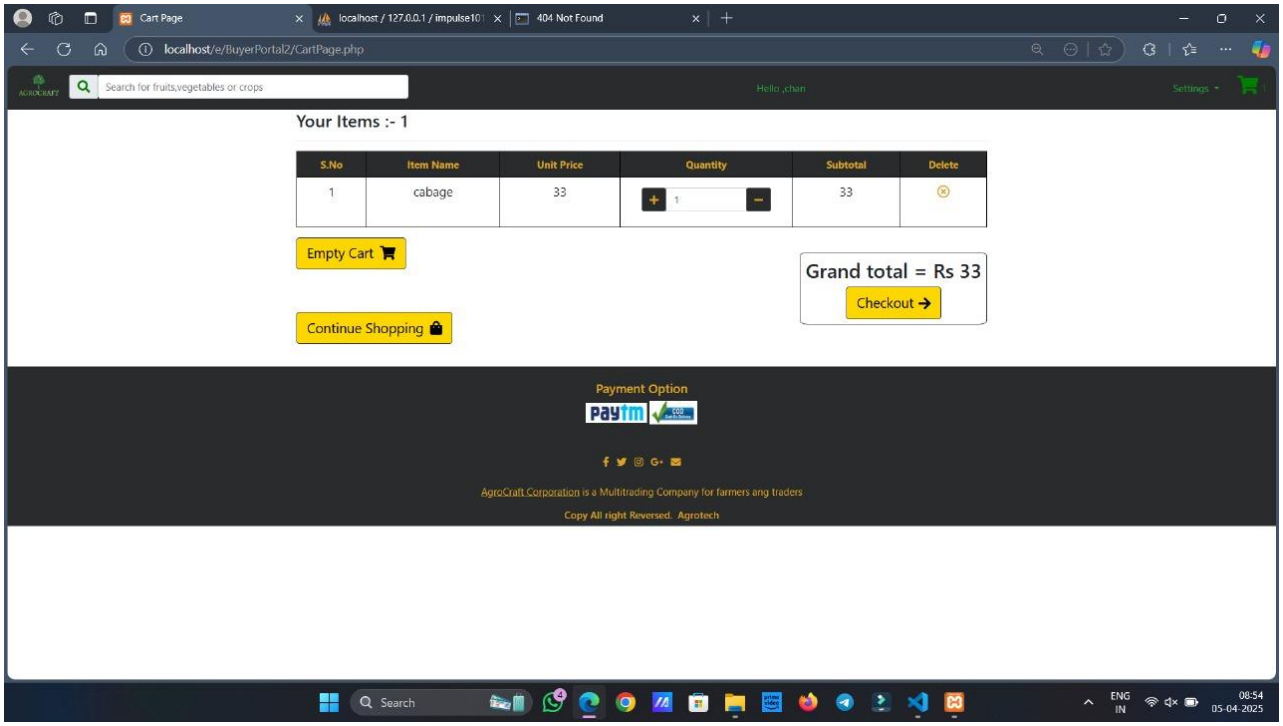

Fig 5.10 Cart

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The Smart AgroCraft project successfully demonstrates the integration of Artificial Intelligence (AI) and Convolutional Neural Networks (CNNs) to optimize agricultural e-commerce by providing an AI-driven vegetable quality assessment system. By reducing manual inspection efforts and human subjectivity, the platform enhances efficiency, trust, and market accessibility for farmers and buyers alike. The system was tested rigorously through unit testing, integration testing, and user validation trials. The results indicate high satisfaction rates among farmers and consumers, with over 90% of farmers finding the system easy to use and 85% of consumers trusting the AI-generated quality scores. Additionally, Smart AgroCraft demonstrates scalability and future potential.

The system was initially deployed on a local XAMPP server, with plans for cloud deployment on AWS or GCP to enhance accessibility and performance. Future enhancements include mobile application development for improved reach, blockchain integration to ensure secure and tamper-proof transactions, and AI powered price prediction models to help farmers optimize their sales strategies. Smart AgroCraft serves as a pioneering AI-driven solution for smart farming and agricultural e-commerce. By leveraging deep learning, cloud computing, and secure transaction mechanisms, the system enhances efficiency, transparency, and scalability in agricultural trade. This project lays the foundation for future innovations in AI-driven agriculture, promoting technological advancements and economic growth in the farming sector.

6.2 FUTURE SCOPE

The Smart AgroCraft platform holds significant potential for future enhancements that can further revolutionize the agricultural e-commerce ecosystem. One of the key areas for expansion is the inclusion of a wider range of crops, including fruits, grains, spices, and dairy products, to cater to a broader market. Additionally, integrating advanced AI features such as aroma detection, shelf-life prediction, and nutritional analysis can elevate the quality assessment process to new levels.

The implementation of blockchain technology could ensure full traceability and transparency in the supply chain, enhancing trust among consumers and industrial buyers. A dedicated mobile application for Android and iOS would increase accessibility for farmers in remote areas, allowing them to participate more actively in digital agriculture. Smart AgroCraft could also support multilingual interfaces to serve diverse user bases across regions. Furthermore, by collaborating with agricultural research institutions, the platform could offer intelligent farming recommendations based on AI insights, thereby empowering farmers to improve their practices and yields. These developments have the potential to position Smart AgroCraft as a leading smart farming and trading solution in the global agri-tech industry.

APPENDICES

SOURCE CODE

App.py

```
from flask import Flask, request, jsonify
import tensorflow as tf
import numpy as np
import cv2
import os
from werkzeug.utils import secure_filename
app = Flask(__name__)
UPLOAD_FOLDER = "static/uploads"
app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024  # 16MB limit
if not os.path.exists(UPLOAD_FOLDER):
 os.makedirs(UPLOAD_FOLDER)
MODEL_PATH = "vegetable_model.h5"
if not os.path.exists(MODEL_PATH):
raise FileNotFoundError(f"Error: Model file '{MODEL_PATH}' not found.")
model = tf.keras.models.load_model(MODEL_PATH)
class_labels = [
    "Bad Quality Brokoli", "Bad Quality Cabbage", "Bad Quality Capsicum",
    "Bad Quality Carrot", "Bad Quality Cauliflower", "Bad Quality Potato",
    "Bad Quality Tomato", "Bad Quality GreenChilli", "Good Quality Brokoli",
    "Good Quality Cabbage", "Good Quality Capsicum", "Good Quality Carrot",
    "Good Quality Cauliflower", "Good Quality GreenChilli", "Good Quality Potato",
    "Good Quality Tomato"
]
def preprocess_image(image_path):
img = cv2.imread(image_path)
if img is None:
raise ValueError("Error: Unable to read the uploaded image. Please upload a valid
image file.")
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224, 224))
img = img / 255.0
img = np.expand_dims(img, axis=0)
return img
```

```
@app.route("/predict", methods=["POST"])
def predict():
if "file" not in request.files:
return jsonify({"error": "No file uploaded"}), 400
file = request.files["file"]
if file.filename == "":
return jsonify({"error": "No selected file"}), 400
if file:
filename = secure_filename(file.filename)
file_path = os.path.join(app.config["UPLOAD_FOLDER"], filename)
file.save(file_path)
try:
img = preprocess_image(file_path)
prediction = model.predict(img)
predicted_class = class_labels[np.argmax(prediction)]
# Check if the vegetable is of good quality
if predicted_class.startswith("Good Quality"):
return jsonify({"prediction": predicted_class, "file_path": file_path})
else:
return jsonify({"error": "Please provide a good quality vegetable!"})
except Exception as e:
return jsonify({"error": str(e)}), 500
if __name__ == "__main__":
app.run(debug=True)
```

Train_model.py

```
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# Define dataset path
DATASET_PATH = "D:\zasprin\cnn1 model for quality analyssi\dataset"  # Update this with your dataset path
IMG_SIZE = (224, 224)
BATCH_SIZE = 32
```

```
# Automatically detect the number of classes
train_datagen = ImageDataGenerator(rescale=1.0/255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
DATASET_PATH, target_size=IMG_SIZE, batch_size=BATCH_SIZE,
class_mode="categorical", subset="training"
)
NUM_CLASSES = train_generator.num_classes  # Automatically set the number of classes
# Load MobileNetV2 without top layers
base_model = MobileNetV2(weights="imagenet", include_top=False,
input_shape=(224, 224, 3))
base_model.trainable = False  # Freeze base model layers
# Add custom layers
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation="relu")(x)
output_layer = Dense(NUM_CLASSES, activation="softmax")(x)  # Corrected class count
# Create model

model = Model(inputs=base_model.input, outputs=output_layer)
# Compile the model
model.compile(optimizer="adam", loss="categorical_crossentropy",
metrics=["accuracy"])
# Validation data
val_generator = train_datagen.flow_from_directory(
   DATASET_PATH, target_size=IMG_SIZE, batch_size=BATCH_SIZE,
class_mode="categorical", subset="validation"
)
# Train the model
model.fit(train_generator, validation_data=val_generator, epochs=10)

# Save the trained model
model.save("vegetable_model.h5")
print(f"✅ Model training complete with {NUM_CLASSES} classes and saved as
'vegetable_model.h5'")
```

Buyerhome.php

```
<?php
include("../Functions/functions.php");
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Buyer Homepage</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></s
cript>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<a href="https://icons8.com/icon/83325/roman-soldier"></a>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src="https://kit.fontawesome.com/c587fc1763.js"
crossorigin="anonymous"></script>
<script>
function state() {
var a = document.getElementById('states').value;
if (a === '31') {
var array = ['Andamans', 'Nicobars'];
} else if (a === '01') {
        var array = ['Adilabad', 'Nizamabad', 'Karimnagar', 'Medak', 'Hyderabad',
'Rangareddi', 'Mahbubnagar', 'Nalgonda', 'Warangal',
'Khammam', 'Srikakulam', 'Vizianagaram', 'Visakhapatnam',
'East Godavari', 'West Godavari', 'Krishna', 'Guntur', 'Prakasam',
'Nellore', 'Cuddapah', 'Kurnool', 'Anantapur', 'Chittoor'];
} else if (a === '02') {
        var array = ['Kokrajhar', 'Dhubri', 'Goalpara', 'Bongaigaon', 'Barpeta', 'Kamrup',
'Nalbari', 'Darrang', 'Marigaon', 'Nagaon', 'Sonitpur',
```

'Lakhimpur','Dhemaji', 'Tinsukia', 'Dibrugarh', 'Sibsagar', 'Jorhat', 'Golaghat', 'Karbi Anglong', 'North Cachar Hills', 'Cachar', 'Karimganj',
'Hailakandi'];
} else if (a === '03') {
var array = ['Pashchim Champaran', 'Purba Champaran', 'Sheohar *', 'Sitamarhi', 'Madhubani', 'Supaul *', 'Araria', 'Kishanganj', 'Purnia', 'Katihar', 'Madhepura', 'Saharsa', 'Darbhanga', 'Muzaffarpur', 'Gopalganj', 'Siwan', 'Saran', 'Vaishali', 'Samastipur', 'Begusarai', 'Khagaria', 'Bhagalpur', 'Banka *', 'Munger', 'Lakhisarai *', 'Sheikhpura *', 'Nalanda', 'Patna', 'Bhojpur', 'Buxar *', 'Kaimur (Bhabua) *', 'Rohtas', 'Jehanabad ', 'Aurangabad', 'Gaya', 'Nawada', 'Jamui *'];
} else if (a === '04') {
var array = ['Kachchh', 'Banas Kantha', 'Patan *', 'Mahesana', 'Sabar Kantha', 'Gandhinagar', 'Ahmadabad', 'Surendranagar', 'Rajkot', 'Jamnagar', 'Porbandar *', 'Junagadh', 'Amreli', 'Bhavnagar', 'Anand *', 'Kheda', 'Panch Mahals', 'Dohad *', 'Vadodara', 'Narmada *', 'Bharuch', 'Surat', 'The Dangs', 'Navsari *', 'Valsad'];
} else if (a === '05') {
var array = ['Panchkula *', 'Ambala', 'Yamunanagar', 'Kurukshetra', 'Kaithal', 'Karnal', 'Panipat', 'Sonipat', 'Jind', 'Fatehabad *', 'Sirsa', 'Hisar', 'Bhiwani', 'Rohtak',
'Jhajjar *', 'Mahendragarh', 'Rewari', 'Gurgaon', 'Faridabad'];
} else if (a === '06') {
var array = ['Chamba', 'Kangra', 'Lahul & Spiti', 'Kullu', 'Mandi', 'Hamirpur', 'Una', 'Bilaspur', 'Solan', 'Sirmaur', 'Shimla', 'Kinnaur'];
} else if (a === '07') {
var array = ['Kupwara', 'Baramula', 'Srinagar', 'Badgam', 'Pulwama', 'Anantnag', 'Leh (Ladakh)', 'Kargil', 'Doda', 'Udhampur', 'Punch', 'Rajauri', 'Jammu', 'Kathua'];
} else if (a === '08') {
var array = ['Belgaum', 'Bagalkot *', 'Bijapur', 'Gulbarga', 'Bidar', 'Raichur', 'Koppal *', 'Gadag *', 'Dharwad', 'Uttara Kannada', 'Haveri *', 'Bellary', 'Chitradurga', 'Davangere*', 'Shimoga', 'Udupi *', 'Chikmagalur', 'Tumkur', 'Kolar', 'Bangalore', 'Bangalore Rural', 'Mandya', 'Hassan', 'Dakshina Kannada', 'Kodagu', 'Mysore', 'Chamrajnagar*'];
} else if (a === '09') {
var array = ['Kasaragod', 'Kannur', 'Wayanad', 'Kozhikode', 'Malappuram', 'Palakkad', 'Thrissur', 'Ernakulam', 'Idukki', 'Kottayam', 'Alappuzha', 'Pathanamthitta', 'Kollam', 'Thiruvananthapuram'];
} else if (a === '10') {
var array = ['Sheopur *', 'Morena', 'Bhind', 'Gwalior', 'Datia', 'Shivpuri', 'Guna', 'Tikamgarh', 'Chhatarpur', 'Panna', 'Sagar', 'Damoh', 'Satna', 'Rewa', 'Umaria *', 'Shahdol', 'Sidhi', 'Neemuch *', 'Mandsaur', 'Ratlam', 'Ujjain', 'Shajapur',

'Dewas', 'Jhabua', 'Dhar', 'Indore', 'West Nimar', 'Barwani *', 'East Nimar', 'Rajgarh', 'Vidisha', 'Bhopal', 'Sehore', 'Raisen', 'Betul', 'Harda *', 'Hoshangabad', 'Katni *', 'Jabalpur', 'Narsimhapur', 'Dindori *', 'Mandla', 'Chhindwara', 'Seoni', 'Balaghat'];
} else if (a === '11') {
var array = ['Nandurbar *', 'Dhule', 'Jalgaon', 'Buldana', 'Akola', 'Washim *', 'Amravati', 'Wardha', 'Nagpur', 'Bhandara', 'Gondiya *', 'Gadchiroli', 'Chandrapur', 'Yavatmal', 'Nanded', 'Hingoli *', 'Parbhani', 'Jalna', 'Aurangabad', 'Nashik', 'Thane', 'Mumbai (Suburban) *', 'Mumbai', 'Raigarh', 'Pune', 'Ahmadnagar', 'Bid', 'Latur', 'Osmanabad', 'Solapur', 'Satara', 'Ratnagiri', 'Sindhudurg', 'Kolhapur', 'Sangli'];
} else if (a == 12) {
var array = ['Senapati', 'Tamenglong', 'Churachandpur', 'Bishnupur', 'Thoubal', 'Imphal West', 'Imphal East *', 'Ukhrul', 'Chandel'];
} else if (a === '18') {
var array = ['Thiruvallur', 'Chennai', 'Kancheepuram', 'Vellore', 'Dharmapuri', 'Tiruvannamalai', 'Viluppuram', 'Salem', 'Namakkal *', 'Erode', 'The Nilgiris', 'Coimbatore', 'Dindigul', 'Karur *', 'Tiruchirappalli', 'Perambalur *', 'Ariyalur *', 'Cuddalore', 'Nagapattinam *', 'Thiruvarur', 'Thanjavur', 'Pudukkottai', 'Sivaganga', 'Madurai', 'Theni *', 'Virudhunagar', 'Ramanathapuram', 'Thoothukkudi', 'Tirunelveli ', 'Kanniyakumari'];
} else if (a === '39') {
var array = ['Yanam', 'Pondicherry', 'Mahe', 'Karaikal'];
} else if (a === '37') {
var array = ['Lakshadweep'];
} else if (a === '42') {
var array = ['North Goa ', 'South Goa'];
} else if (a === '34') {
var array = ['Dadra & Nagar Haveli'];
} else if (a === '36') {
var array = ['Diu', 'Daman'];
} else if (a === '23') {
var array = ['Koriya *', 'Surguja', 'Jashpur *', 'Raigarh', 'Korba *', 'Janjgir - Champa*', 'Bilaspur', 'Kawardha *', 'Rajnandgaon', 'Durg', 'Raipur', 'Mahasamund *', 'Dhamtari *', 'Kanker *', 'Baster', 'Dantewada*'];
} else if (a === '24') {
var array = ['Garhwa *', 'Palamu', 'Chatra *', 'Hazaribag', 'Kodarma *', 'Giridih', 'Deoghar', 'Godda', 'Sahibganj', 'Pakaur *', 'Dumka', 'Dhanbad', 'Bokaro *', 'Ranchi', 'Lohardaga', 'Gumla', 'Pashchimi Singhbhum', 'Purbi Singhbhum', 'ORISSA',

```
'Bargarh, 'Jharsuguda *', 'Sambalpur', 'Debagarh *', 'Sundargarh', 'Kendujhar',
'Mayurbhanj', 'Baleshwar', 'Bhadrak *', 'Kendrapara *',
'Jagatsinghapur *', 'Cuttack', 'Jajapur *', 'Dhenkanal', 'Anugul *', 'Nayagarh *',
'Khordha *', 'Puri', 'Ganjam', 'Gajapati *', 'Kandhamal', 'Baudh *',
'Sonapur *', 'Balangir', 'Nuapada *', 'Kalahandi', 'Rayagada *', 'Nabarangapur *',
'Koraput', 'Malkangiri *'];
} else if (a === '21') {
    var array = ['Darjiling ', 'Jalpaiguri ', 'Koch Bihar ', 'Uttar Dinajpur', 'Dakshin
Dinajpur *', 'Maldah ', 'Murshidabad ', 'Birbhum', 'Barddhaman ', 'Nadia ', 'North
Twenty Four Parganas', 'Hugli ', 'Bankura ', 'Puruliya', 'Medinipur ', 'Haora ', 'Kolkata',
'South Twenty Four Parganas'];
} else if (a === '13') {
    var array = ['West Garo Hills', 'East Garo Hills', 'South Garo Hills *', 'West
Khasi Hills', 'Ri Bhoi *', 'East Khasi Hills', 'Jaintia Hills'];
} else if (a === '22') {
    var array = ['North ', 'West', 'South', 'East'];
} else if (a === '20')
    var array = ['Ganganagar', 'Hanumangarh *', 'Bikaner', 'Churu', 'Jhunjhunun',
'Alwar', 'Bharatpur', 'Dhaulpur', 'Karauli *', 'Sawai Madhopur',
'Dausa *', 'Jaipur', 'Sikar', 'Nagaur', 'Jodhpur', 'Jaisalmer', 'Barmer', 'Jalor',
'Sirohi', 'Pali','Ajmer', 'Tonk', 'Bundi', 'Bhilwara', 'Rajsamand *',
'Udaipur', 'Dungarpur', 'Banswara', 'Chittaurgarh',
Kota', 'Baran *','Jhalawar'];
} else if (a === '16') {
    var array = ['Gurdaspur', 'Amritsar', 'Kapurthala', 'Jalandhar', 'Hoshiarpur',
'Nawanshahr *', 'Rupnagar', 'Fatehgarh Sahib *', 'Ludhiana', 'Moga *',
'Firozpur', 'Muktsar *', 'Faridkot', 'Bathinda', 'Mansa *', 'Sangrur',
'Patiala'];
} else if (a === '14') {
    var array = ['Mon', 'Tuensang', 'Mokokchung', 'Zunheboto', 'Wokha', 'Dimapur
*', 'Kohima', 'Phek'];
} else if (a === '19') {
    var array = ['West Tripura ', 'South Tripura ', 'Dhalai *', 'North Tripura '];
} else if (a === '38') {
    var array = ['Mamit *', 'Kolasib *', 'Aizawl', 'Champhai *', 'Serchhip *',
'Lunglei', 'Lawngtlai', 'Saiha *'];
} else if (a === '32') {
    var array = ['Tawang', 'West Kameng', 'East Kameng', 'Papum Pare *', 'Lower
Subansiri', 'Upper Subansiri', 'West Siang', 'East Siang', 'Upper
```

```
Siang *', 'Dibang Valley', 'Lohit', 'Changlang', 'Tirap'];
} else if (a === '33') {
        var array = ['Chandigarh'];
} else if (a === '35') {
        var array = ['North West   *', 'North   *', 'North East   *', 'East   *', 'New Delhi',
'Central  *', 'West   *', 'South West   *',                    'South  *'];
} else if (a === '40') {
        var array = ['Uttarkashi', 'Chamoli', 'Rudraprayag *', 'Tehri Garhwal',
'Dehradun', 'Garhwal', 'Pithoragarh', 'Bageshwar',                    'Almora',
'Champawat', 'Nainital', 'Udham Singh Nagar *', 'Hardwar'];
}
var string = "";
for (let i = 0; i < array.length; i++) {
string = string + "<option>" + array[i] + "</option>";
}
string = "<select nmae = 'lol'>" + string + "</select>"
document.getElementById('district').innerHTML = string;
}
</script>
<script>
var a;
function display() {
if (a == 0) {
document.getElementById("majic").style.visibility = "hidden";
document.getElementById("show").style.visibility = "visible";
return a = 1;
} else {
document.getElementById("majic").style.visibility = "visible";
document.getElementById("show").style.visibility = "hidden";
// document.getElementById("show").style. visibility= "hidden";
return a = 0;
}
}
</script>
</script> -->
</head>
<body>
<nav class="navbar navbar-expand-xl ">
<div class=" flex-row-reverse left ">
```

```
<div class="p-2">
<div class="icon2">
<a href="CartPage.php"> <i class="fa" style="font-size:30px; color:green ;margin-top:2px;">&#61562;</i></a>
<span id="icon" style="color:green"> <?php echo totalItems(); ?> </span>
</div>
</div>
<div class="p-2 ml-5"><i class='far fa-user-circle' style='font-size:30px; color: green;margin-top:2px;'></i></div>
<a class="float-left" href="bhome.php">
<img src="agro.png" class="float-left mr-5 ml-0 " alt="Logo" style="height:50px;">
</a>
</div>
<button class="navbar-toggler" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"><i class="fas fa-bars p-1 " style="color:green;margin-right:-9%;font-size:28px;"></i></span>
</button>
<a class="float-left" href="bhome.php">
<img src="agro.png" class="float-left mr-2 moblogo" alt="Logo" style="height:50px;">
</a>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<div class="input-group mb-1 ml-2 searchbox">
<div class="input-group-prepend">
<div class="input-group-text"><i class="fas fa-search" style="font-size:20px;color:green; "></i></div>
</div>
<form action="SearchResult.php" method="get" enctype="multipart/form-data">
<input type="text" class="form-control " id="inlineFormInputGroup" name="search" placeholder="Search for fruits,vegetables or crops " style="width:500px;">
</form>
</div>
<?php
getUsername();
?>
<div class="list-group moblists">
<?php
```

```
if (isset($_SESSION['phonenumber'])) {
echo "<a href='BuyerProfile.php' class='list-group-item list-group-item-action'
style='background-color:#292b2c;text-align:center;color:goldenrod'>Profile</a>";
echo "<a href= 'Transaction.php' class='list-group-item list-group-item-action'
style='background-color:#292b2c;text-
align:center;color:goldenrod'>Transactions</a>";
echo "<a href='../Includes/logout.php' class='list-group-item list-group-item-action '
style='background-color:#292b2c;text-align:center;color:goldenrod'>Logout</a>";
} else {
echo "<a href='../auth/BuyerLogin.php' class='list-group-item list-group-item-action '
style='background-color:#292b2c;text-align:center;color:goldenrod'>Login</a>";
}
?>
</div>
</div>
<div class=" flex-row-reverse right ">
<div class="p-2 cart">
<div class="icon2">
<a href="CartPage.php"> <i class="fa" style="font-size:30px;
color:green">&amp;#61562;</i></a>
<span id="icon" style="color:green"> <?php echo totalItems(); ?> </span>
</div>
</div>
<div class="dropdown p-2 settings ">
<button class="btn  dropdown-toggle text-success" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
Settings
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
<?php
if (isset($_SESSION['phonenumber'])) {
echo "<a href='BuyerProfile2.php' class='dropdown-item  ' style='padding-right:-
20px;'>Profile</a>";
echo "<a href='Transaction.php' class='dropdown-item ' style='padding-right:-
20px;'>Transactions</a>";
// echo "<a href='#' class='dropdown-item'  style='padding-right:-
20px;'>Subscriptions</a>";
// echo "<a href='saveforlater.php' class='dropdown-item' style='padding-right:-
```

```
20px;'>Save For Later</a>";
echo "<a href='farmers.php' class='dropdown-item' style='padding-right:-20px;'
>Farmers</a>";
echo "<a href='../Includes/logout.php' class='dropdown-item ' style='padding-right:-
20px;'>Logout</a>";
} else {
echo "<a href='../auth/BuyerLogin.php' class='dropdown-item ' style='padding-
right:-20px;'>Login</a>";
}
?>
</div>
</div>
<div class="text-success  login">Login</div>
</div>
</nav>
<div class="container">
<div class="d-flex justify-content-around bg-white mb-3">
<div class="p-2 ">
<div class="dropdown">
<button class="btn btn-green mybtn dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
Fruits
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
<?php
getFruits();
?>
</div>
</div>
</div>
<div class="p-2">
<div class="dropdown">
<button class="btn btn-green mybtn dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
<?php
```

```
getVegetables();
<div class="p-2 ">
<div class="dropdown">
<button class="btn btn-green mybtn dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
Crops
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
<?php
getCrops();
?>
</div>
</div>
</div>
</div>
</div>
<div class="container"> <img src="b4.jpeg" class="img-fluid firstimage d-block
mx-auto" alt="Responsive image">
</div>
<br>
<br>
<div class="container">
<div class="text-center">
<!-- <h2 id="headings" class="destext">Fresh fruits</h2> -->
<h1 id="headings" class="guard"><span><b>Fresh Fruits </b></span>
</h1>
</div>
<hr>
<div class="row BigBox">
<?php
getFruitsHomepage();
?>
<hr>
</div>
<hr>
</div>
<br><br>
<div class="container">
```

```
<div class="text-center">
<!-- <h2 id="headings" class="destext">Fresh fruits</h2> -->
<h1 id="headings" class="guard"><span><b>Fresh Vegetables </b></span>
</h1>
</div>
<hr>
<div class="row BigBox">
<?php
getVegetablesHomepage();
<br><br>
<div class="container">
<div class="text-center">
<h1 id="headings" class="longguard"><span><b>Best Selling Products All Over
India </b></span>
</h1>
</div>
<br>
<div class="row">
<?php
cart();
getProducts();
?>
</div>
<br><br>
</div>
</div>
<!-- footer -->
<section id="footer" class="myfooter">
<div class="container">
<div class="row text-center text-xs-center text-sm-left text-md-left">
<div class="col aligncenter">
<br>
<h5>Payment Option</h5>
<img src="../Images/Website/paytm1.jpg" alt="paytm">
<img src="../Images/Website/cod.jpg" alt="paytm" style="height:37px">
</div>
</div>
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12 mt-2 mt-sm-5">
```

```
<ul class="list-unstyled list-inline social text-center">
<li class="list-inline-item"><a href="javascript:void();"><i class="fa fa-facebook"></i></a></li>
<li class="list-inline-item"><a href="javascript:void();"><i class="fa fa-twitter"></i></a></li>
<li class="list-inline-item"><a href="javascript:void();"><i class="fa fa-instagram"></i></a></li>
<li class="list-inline-item"><a href="javascript:void();"><i class="fa fa-google-plus"></i></a></li>
<li class="list-inline-item"><a href="javascript:void();" target="_blank"><i class="fa fa-envelope"></i></a></li>
</ul>
</div>
</hr>
</div>
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12 mt-2 mt-sm-2 text-center">
<p><u><a href="https://www.chandru.wuaze.com/">AgroCraft Corporation</a></u> is a Multitrading Company for farmers ang traders</p>
<p class="h6">Copy All right Reversed.<a class="text-green ml-2" href="https://www.google.com" target="_blank">Agrotech</a></p>
</div>
</hr>
</div>
</div>
</section>
<!-- ./Footer a ,myfooter,aligncenter-->
</body>
</html>
```

REFERENCES

1. Kiran Shinde et al., “Web Based Recommendation System for Farmers”, March 2015.
2. Vikas Kumar et al., “Krishi Mantra: Agricultural Recommendation System”, Jan 2013.
3. Mansi Shinde et al., “Crop Recommendation and Fertilizer Purchase System”, 2016.
4. Vikas Kumar et al., “Crop Cultivation Information System on Mobile”, 2013.
5. Cheng-Lin Liu et al., “Handwritten digit recognition: benchmarking of state-of-the-art techniques”, Pattern Recognition, Elsevier, 2003.
6. Mahmoud M. Abu Ghosh et al., “A Comparative Study on Handwriting Digit Recognition Using Neural Networks”, IEEE, 2017.
7. Sharath D M and Rohan M G, “Disease Detection in Pomegranate using Image Processing”, International Conference on Trends in Electronics and Informatics, 2020.
8. M. Pushpavalli, “Image Processing Technique for Fruit Grading”, International Journal of Engineering and Advanced Technology (IJEAT), 2019.
9. Neha. Building crop ontology for farmers. Masters thesis, Banasthali University, Rajasthan, 2012.
10. T. Sasipraba R. Jeberson Retna Raj. Disaster management system based on gis web services. In Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010.
11. Md. Rokunuzzaman, H. P. W. Jayasuriya," Development of a low cost machine vision system for sorting of tomatoes", CIGR Journal, Vol. 15, No. 1,pp. 173-180,2013. 51
12. V. Leemans, H. Mageinb,M. -F. Destain," On-line Fruit Grading according to their External Quality using Machine Vision", Journal of Automation and

Emerging Technologies, Belgium. Biosystems Engineering, pp. 397 - 404, 2002.

13. Mustafa, N. B. A.; Ahmed, S. K.; Ali, Z.; Yit, W. B.; Abidin, A. A. Z.; Sharrif, Z. A. M., "Agricultural Produce Sorting and Grading using Support Vector Machines and Fuzzy Logic," Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on Signal and Image Processing, vol.18, pp. 391,396, Nov. 2009.
14. Woo Chaw Seng and Seyed Hadi Mirisaee, "A New Method for Fruits Recognition System," Electrical Engineering and Informatics, vol. 01, pp. 130-134, August 2009.
15. Hongshe Dang, Jinguo Song, Qin Guo, "A Fruit Size Detecting and Grading System Based on Image Processing", 2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 2, pp. 83-86, August 2010.
16. M. Khojastehnazhand, M. Omid, and A. Tabatabaeefar, "Development of a lemon sorting system based on color and size," African Journal of Plant Science, vol. 4(4), pp. 122-127, April 2010.
17. Yousef Al Ohali, "Computer vision based date fruit grading system: Design and implementation," Journal of King Saud University - Computer and Information Sciences, vol. 23, no. 1, pp. 29-39, January 2011.
18. K.Sakthivel, R.Abinaya, I.Nivetha, R.Arun Kumar,"Region Based Image Retrieval using k-means and Hierarchical Clustering Algorithms",International Journal of Innovative Research in Science, Engineering and Technology,pp.1255-1260,Volume 3, Special Issue 1, February 2014.
19. Devendra. A. Itole, Madhura. R. Lodam, Sneha. N. Panse, Vaishavi. S. Kulkarni," Fruit Quality 10. Management and Sorting System", International 52 Journal of Science, Engineering and Technology,Volume 3 Issue 2,pp. 394- 396, 2015.

20. Rafel C Gonzalez. Digital image processing, 3rd edition, Pearson Education India, 2009.

21. R. Venkata Ramana Chary,Dr. D. Rajya Lakshmi and Dr. K.V.N Sunitha, “Feature Extraction Methods for Color Image Similarity” Advanced Computing: An International Journal (ACIJ), Vol.3, No.2, pp.147 - 157,March 2012.