TAV-Cart: Transaction Assistance and Validation Cart System for Effortless Shopping and Checkout

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer Engineering

Submitted To

SVKM's NMIMS,
Mukesh Patel School of Technology Management & Engineering,
Shirpur Campus (M.H.)

Submitted By:

Akshit Naithani – 70022100234
Tanish Singh Rajpal – 70022100030
Vansh Mistry – 70022100356

Under The Supervision Of:

Prof. Kiran Salunke
(Computer Engineering, Assistant Professor)

DEPARTMENT OF COMPUTER ENGINEERING
Mukesh Patel School of Technology Management & Engineering
ACADEMIC SESSION: 2024-25

CERTIFICATE

This is to certify that the project entitled "**TAV-Cart: Transaction Assistance and Validation Cart System for Effortless Shopping and Checkout"** has been done by

"Akshit Naithani" (Roll No.: 70022100234)

"Tanish Singh Rajpal" (Roll No.: 70022100030)

"Vansh Mistry" (Roll No.: 70022100356)

under my guidance and supervision & has been submitted partial fulfillment of the degree of "**Bachelor of Technology** in **Computer Engineering"** of MPSTME, SVKM'S NMIMS (Deemed-to-be university), Mumbai (M.H.), India.

Prof. Kiran Salunke
(Internal Guide)

Examiner

Date:
Place: Shirpur

H.O.D.

DEPARTMENT OF COMPUTER SCIENCE
Mukesh Patel School of Technology Management & Engineering

ACKNOWLEDGEMENT

After the completion of this capstone Project work, words are not enough to express my feelings about all those who helped me to reach my goal; feeling above this is my indebtedness to The Almighty for providing me this moment in life.

It's a great pleasure and moment of immense satisfaction for me to express my profound gratitude to **Prof. Kiran Salunke, Assistant Professor**, Computer Engineering Department, MPSTME, Shirpur, whose constant encouragement enabled me to work enthusiastically. Their perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited me to an extent, which is beyond expression. Their depth and breadth of knowledge of Computer Engineering field made me realize that theoretical knowledge always helps to develop efficient operational software, which is a blend of all core subjects of the field. I am highly indebted to them for their invaluable guidance and ever-ready support in the successful completion of this project in time. Working under their guidance has been a fruitful and unforgettable experience.

We express my sincere thanks and gratitude to **Dr. Radhakrishna Rambola**, Head of Department, Computer Engineering, MPSTME, Shirpur, for providing necessary infrastructure and help to complete the project work successfully.

We also extend my deepest gratitude to **Dr. Venkatadri Marriboyina**, Associate Dean, SVKM'S NMIMS, Shirpur Campus for providing all the necessary facilities and true encouraging environment to bring out the best of my endeavors.

We sincerely wish to express my grateful thanks to all members of the staff of computer engineering department and all those who have embedded me with technical knowledge of computer technology during various stages of B.Tech. Computer Engineering.

We would like to acknowledge all my friends, who have contributed directly or indirectly in this capstone Project work.

The successful completion of a capstone Project is generally not an individual effort. It is an outcome of the cumulative effort of a number of persons, each having their own importance to the objective. This section is a vote of thanks and gratitude towards all those persons who have directly or indirectly contributed in their own special way towards the completion of this project.

Akshit Naithani

Tanish Singh Rajpal

Vansh Mistry

ABSTRACT

This research paper presents the 'TAV-Cart,' a creative intelligent shopping cart about to take the retail market to a modern era. The TAV-Cart is an innovative smart shopping cart about to set in motion the evolution of the retail market. It has adopted a solution-oriented approach by introducing a new generation of UPI-based shopping carts that are going to turn the retail industry upside down. It helps them to navigate the stores, select their products, and check the products using the automated barcode scanning technology and the weight verification technology. A light strip inside the cart emphasizes real-time visual information; thus, the scanned products are properly taken care of, which in turn creates a smooth shopping experience. One of the features that the customer can use to pay at the end of their consumer journey is the UPI platform itself, which eliminates the pain of long lines at checkout counters. Very strongly linked with the UPI payments system in India, the TAV-Cart meets the five basic aspects of the UPI system. TAV-Cart emerges as an instrumental solution for modernizing the retail landscape, enhancing shopping efficiency, and catering to evolving consumer preferences. TAV-Cart represents a significant step forward in modernizing the retail experience, aligning with the rapid adoption of UPI as a preferred payment method in India. As the retail landscape continues to evolve, TAV-Cart offers a compelling solution for both customers seeking streamlined shopping experiences and retailers striving to stay competitive in a digital era. The paper also provides concept ideas for other technological improvements that can be made in the supermarket to further ease the shopping journey of the customer. These advancements can be implemented along with the TAV-Cart to automate the entire shopping mart experience.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

1.1 Background of the project topic

The retail industry is said to be revolutionized by fast technological changes for better customer experiences and to just squeeze every little thing from the business operations. Traditional shopping practices usually involve long and cumbersome check-out processes that may lead to potential consumer dissatisfaction and lost sale. Reports from the industry show that the major consumer who abandoned his purchases indicated extended waiting time in the queue before checkout counters as the reason. It clearly highlights the urgent need for new solutions that could possibly better the retailing experience.

And, finally, the innovative shopping carts have emerged as transforming products in facing new challenges. The TAV-Cart employs the combination of new technologies-barcode scanning, weight validation, and real-time data processing-to make the process of purchasing automated and straightforward. Using advanced laser and camera-based technologies of scanning, the TAV-Cart ensures items are identified accurately, drastically reducing the errors that occur with manual scanning.

This upgrade not only heightens the customer experience but also allows for better stock management so that retailers can maintain exact levels of stock and minimize waste.

Thus, the security aspect is absorbed naturally by the smart cart weighing validation mechanisms, so that deception through product substitution is not possible and correct articles are captured at the checkout stage. While security in retail transactions is important because fraudulent behavior, besides reducing profitability, affects the confidence of a consumer in the brand.

Thus, the critical necessity for an easy confluence of payment systems with the retail environment presents with the advent of mobile payment solutions, specifically the Unified Payments Interface (UPI) in India. UPI allows for immediate transactions, which aligns with the consumer's expectancy in terms of ease and convenience that he or she associates with a buying journey. Such technological advancements necessitate a more digitized retail world wherein consumers expect speedier and easier payment options. This TAV-Cart represents the solution in the forefront: it modernizes the shopping experience while focusing on problematic issues related to efficiency, security, and satisfaction. With cloud-based technology and IoT protocols, the TAV-Cart is going to take a very central position in the remodeling of the future of retail shopping. This new and innovative design and functionality are likely to increase user interaction while providing retailers with much-needed analytics into consumer behavior and buying patterns.

1.2 Motivation and scope of the report

The TAV-Cart project is motivated by the growing need for innovation in the retail sector, driven by evolving consumer demands for faster, seamless, and secure shopping experiences. With increasing competition in the retail industry and the rise of e-commerce, brick-and-mortar stores are under pressure to modernize and offer comparable levels of convenience. Traditional checkout processes, often plagued by inefficiencies like long queues, manual errors, and security vulnerabilities, have become a bottleneck in delivering optimal customer satisfaction. These challenges highlight the need for solutions that not only enhance customer experience but also improve operational efficiency and security.

The primary motivation behind the TAV-Cart system is to address these persistent challenges through a technology-driven solution that automates transactions and ensures secure validation of purchased items. By integrating barcode scanning, weight validation, IoT connectivity, and mobile payment systems, the TAV-Cart aims to eliminate the friction points in traditional retail environments, such as slow checkout processes, fraud risks, and poor inventory management. Additionally, with the increasing adoption of cashless transactions—especially mobile platforms like UPI in India—the motivation to streamline payment systems becomes crucial to meet modern consumer expectations.

The scope of the report extends to covering multiple aspects of this transformation, including the design and implementation of the TAV-Cart, its technical functionality, and the impact on stakeholders such as customers, retailers, and supply chain managers. The TAV-Cart system focuses on creating an integrated, real-time shopping experience by offering automated product identification, enhanced security through weight validation, and seamless payment processing.

Moreover, the report will delve into how the system leverages cloud technologies and IoT protocols to support data-driven decision-making, such as generating insights on consumer behavior and stock management. It will also explore scalability, detailing the potential for expanding the system to accommodate future technological developments, including NFC-based promotions, automated exits, and personalized shopping experiences.

1.3 Problem statement

Traditional checkout systems in retail cause delays, manual errors, and security risks, leading to customer dissatisfaction and increased operational costs. They struggle with integrating UPI-based payments, especially in regions like India, causing further inefficiencies. TAV-Cart addresses these challenges with automated product identification, real-time weight validation, and seamless payment integration. By minimizing manual intervention and preventing fraud, it ensures a faster, secure, and contactless shopping experience, aligning with modern consumer expectations.

1.4 Salient contribution

The TAV-Cart project presents a transformative solution to enhance the retail shopping experience by addressing inefficiencies in traditional checkout processes. It integrates advanced technologies such as automated barcode scanning, real-time weight validation, and UPI-based payment systems to streamline the entire shopping and checkout experience. By automating product identification and payment validation, the TAV-Cart reduces the chances of manual errors, minimizes fraud, and ensures accurate billing.

One of the core contributions of TAV-Cart is its emphasis on security and transparency through weight validation, which prevents fraudulent activities such as product switching or misidentification. Seamless UPI payment integration allows for faster transactions, eliminating delays at checkout counters and enhancing the overall customer experience. This feature also caters to the growing demand for contactless payments, promoting a safer, modern retail environment.

The TAV-Cart system reduces operational costs by limiting the need for human intervention, allowing retailers to optimize workforce management. Additionally, it improves inventory management by accurately tracking purchases in real time, helping retailers maintain stock levels and avoid discrepancies. The use of IoT protocols and cloud-based technologies further supports scalability, making the system adaptable to various retail environments.

1.5 Organization of report

The report is organized into several sections that collectively provide a comprehensive overview of the College Placement Portal project. It begins with an Introduction, outlining the project's objectives, significance, and background to establish context. Following this, the Literature Survey reviews existing research and technologies related to placement systems, identifying gaps that the portal aims to fill. The Methodology and Implementation section details the technical approach, architecture, and tools utilized in the development process. Next, the Results and Analysis section presents the project outcomes, including user feedback and performance metrics, while analyzing the data to evaluate the portal's effectiveness. The report then discusses the Advantages, Limitations, and Applications of the portal, highlighting its benefits and any challenges encountered. Finally, the Conclusion and Future Scope summarizes the findings and suggests potential enhancements.

Additionally, the report includes references, sample code, data sheets, a list of components, and a compilation of papers related to the project.

Chapter 2

Literature survey

2.1 Introduction to overall topic

The TAV-Cart system emerges as a groundbreaking solution designed to revolutionize the shopping experience in retail environments. By integrating advanced technologies such as UPI payment systems and automated barcode scanning, TAV-Cart addresses the growing demands of modern consumers who seek efficiency, convenience, and enhanced service quality while shopping. The system is not merely a technological upgrade; it represents a paradigm shift in how retailers engage with customers and manage transactions.

In today's fast-paced shopping landscape, consumers are increasingly looking for seamless experiences that minimize waiting times and streamline the purchasing process. The TAV-Cart enhances user experience by allowing shoppers to manage their purchases effortlessly, reducing the hassle often associated with traditional checkout processes. The integration of payment processing directly into the cart facilitates immediate transaction validation, ensuring that customers can complete their purchases with minimal friction. This level of convenience is essential in attracting and retaining customers in a competitive market.

Moreover, the TAV-Cart employs a sophisticated inventory management system that ensures real-time tracking of products. Retailers benefit from this functionality by gaining insights into stock levels, enabling them to optimize inventory management and reduce waste. The cloud database management system supports this capability, allowing for efficient data storage and access. This synergy between technology and inventory control not only enhances operational efficiency but also contributes to better decision-making based on sales data and customer preferences.

The camera-based scanner incorporated into the TAV-Cart exemplifies the system's commitment to innovation. By allowing users to scan items directly into their carts, the TAV-Cart further minimizes the time spent at checkout, transforming the shopping experience into a more engaging and dynamic process. This feature aligns with the increasing consumer preference for self-service options, reflecting broader trends toward automation in retail.

Additionally, the TAV-Cart's design considers the physical aspects of shopping carts, ensuring that the system is easy to use and accessible to a diverse range of consumers.

With features like lighting strips for visibility and ergonomic design, TAV-Cart prioritizes user comfort and safety. This focus on user-centric design is crucial in fostering customer loyalty and satisfaction, which are vital for sustained business success.

In summary, the TAV-Cart system is poised to redefine the shopping experience by merging technology with everyday retail practices. Through its innovative features, the system not only addresses the pain points of traditional shopping but also aligns with contemporary consumer expectations for speed and convenience. As retailers adopt this technology, they position themselves at the forefront of the evolving retail landscape, paving the way for a more efficient, customer-friendly shopping experience. Ultimately, the TAV-Cart represents a significant advancement in retail technology, underscoring the importance of innovation in meeting the ever-changing needs of consumers and enhancing overall shopping satisfaction.

2.2 Exhaustive Literature Survey

Karjol et al. (2018)[1] present a pioneering IoT-based smart shopping cart system that leverages various interconnected sensors, RFID technology, and weight sensors to improve the efficiency and accuracy of retail shopping. Their research identifies several key challenges faced in traditional shopping, such as long checkout lines, shoplifting, and inventory mismanagement. To address these challenges, the authors propose a hybrid model integrating weight-based sensors and RFID technology. The weight sensors serve as a verification mechanism, cross-referencing product weight against the corresponding RFID tags to ensure the correct items are scanned. This dual-layer verification not only minimizes potential theft but also reduces discrepancies between scanned and billed products, thereby ensuring transparency for both customers and store managers.

Their system's RFID-enabled real-time product tracking plays a crucial role in preventing shrinkage—a common problem in retail—and supports more effective stock management. With every product added to or removed from the cart, the system updates the inventory records instantly, allowing store managers to monitor stock levels dynamically. One of the key innovations of Karjol et al.'s work is the development of an automatic billing feature integrated into the cart itself, reducing reliance on traditional checkout counters. As shoppers finish adding items, the bill is generated in real-time, facilitating self-checkout without waiting in queues.

The research stands out for its emphasis on security, as it incorporates real-time monitoring of carts through an IoT network, which alerts store personnel if suspicious activities are detected, such as unauthorized product removal. The authors conducted rigorous testing across multiple environments and concluded that their system achieved 98% accuracy in detecting and billing products, marking a substantial improvement over traditional manual billing practices. This innovation not only enhances operational efficiency but also delivers a more streamlined and enjoyable shopping experience for consumers.

Roopa and Reddy (2020)[2] make an impactful contribution to smart retail systems by focusing on artificial intelligence (AI) and machine learning (ML) algorithms to personalize the shopping experience. They propose an innovative product recommendation system embedded within the cart interface that utilizes historical purchase data and real-time shopping behaviors to predict customer preferences with remarkable precision. The implementation of deep learning techniques, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), enables the system to analyze vast datasets and make personalized product recommendations during a shopper's visit. Their approach demonstrates an 85% accuracy rate in predicting relevant product suggestions, making the shopping journey more efficient and satisfying.

The smart cart developed by Roopa and Reddy also includes an intuitive user interface that interacts with customers in real-time, offering product alternatives, discounts, and promotional offers based on their preferences. For instance, if a customer places an item like pasta in the cart, the system might recommend complementary products such as pasta sauce or cheese. The study highlights that by leveraging dynamic recommendation systems, customers tend to spend 30% less time shopping, thus improving overall convenience.

Moreover, the cart features voice-based search functionality, enabling users to search for products without manual interaction. The study presents comprehensive interaction pattern analysis, revealing that customers appreciate the seamless integration of recommendations, particularly when aligned with personalized offers. The authors also discuss the implementation challenges of data privacy, as the system processes large amounts of customer data. They propose a hybrid model that balances personalization with privacy by anonymizing customer data after recommendations are generated. This ensures compliance with modern privacy laws, such as GDPR, without compromising the effectiveness of the recommendation engine.

Das et al. (2020) [3] introduce a revolutionary concept of smart trolleys, focusing on indoor navigation and product localization to enhance the customer experience. They propose a hybrid positioning system that combines Ultra-Wideband (UWB) technology with inertial sensors to achieve precise localization of products within the retail space. UWB technology provides sub-meter accuracy, which is crucial for locating products in large stores where GPS signals are often unavailable. The inertial sensors further assist in tracking the movement of trolleys, ensuring that customers can easily find desired products without wandering aimlessly.

The research emphasizes the importance of time efficiency in modern retail, demonstrating that their system can reduce shopping time by 45% compared to traditional methods. One of the unique features of their solution is the incorporation of route optimization within the cart system. As customers input their shopping list into the trolley's interface, the system calculates the most efficient path through the store, minimizing unnecessary movement. This not only improves customer satisfaction but also optimizes in-store traffic flow, reducing congestion during peak hours.

The scalability of their system is another key focus. The authors address challenges associated with managing multiple smart trolleys in the same retail space, ensuring that communication between trolleys does not interfere with individual cart operations. Their experimental results, conducted in collaboration with a large supermarket, validate the robustness of the proposed system under real-world conditions. They successfully demonstrate that their smart trolley system remains accurate even when deployed at scale, laying the groundwork for future research in multi-cart environments

Subudhi and Ponnalagu (2019)[4] advance the field with their research on automatic product detection and secure payment systems, focusing on computer vision and blockchain technologies. Their innovative approach uses a high-resolution camera-based scanner mounted on the smart cart to identify products, achieving an impressive 99.2% accuracy under varying lighting conditions. This camera-based identification system eliminates the need for manual barcode scanning, offering customers a faster, contactless checkout experience.

A notable contribution of this work is the integration of blockchain technology into the payment process. By utilizing a decentralized ledger, the authors ensure secure, transparent, and tamper-proof transactions. Their system supports multiple payment methods, including credit cards, mobile wallets, and cryptocurrencies, allowing for flexible and seamless payments. Additionally, the study presents a unique solution for handling simultaneous scanning of multiple items, significantly reducing checkout time by 60% compared to conventional methods.

Subudhi and Ponnalagu also explore the importance of system security, proposing multi-factor authentication for payment verification to prevent fraudulent transactions.

They argue that the integration of blockchain enhances trust between retailers and consumers by providing immutable transaction records. Their system not only improves operational efficiency but also aligns with the growing demand for secure and transparent retail transactions.

Gupte et al. (2020) [5] present a novel approach to smart shopping by combining RFID technology with collaborative filtering algorithms to enhance product recommendations. Their system leverages real-time shopping data collected from RFID-enabled carts, clustering customers based on shared preferences and purchase histories. This collaborative filtering technique enables the smart cart to predict customer preferences with a 92% accuracy rate, delivering highly relevant product recommendations.

The authors introduce a distributed computing architecture to handle the large-scale data generated by multiple cart systems. This architecture ensures seamless data processing and allows for real-time analytics, making the system suitable for deployment in large retail environments. An essential aspect of their research is the treatment of privacy concerns. To maintain customer trust, they propose an innovative data anonymization framework that ensures customer identities are protected without compromising the effectiveness of the recommendation engine.

Their research emphasizes the importance of balancing personalization with privacy, showcasing that their system complies with modern data protection regulations while still achieving high recommendation accuracy. The study concludes with an analysis of user feedback, revealing that customers are more likely to trust and engage with personalized recommendations when assured of data privacy.

Guntur et al. (2023) [6] make a significant advancement in smart shopping systems by presenting a cost-effective RFID-based cart system tailored for small to medium-sized retailers. At the heart of their design lies the integration of NodeMCU microcontrollers, which enable seamless communication between sensors and backend systems. Their research addresses the challenge of affordability in smaller retail environments, where high-end IoT-based solutions may not be viable. The proposed system provides robust functionality without compromising on features, offering efficient inventory management and enhancing customer satisfaction through smooth checkout experiences.

One of the notable contributions of their research is the innovative power management strategy implemented in the smart cart. The system leverages intelligent sleep modes and optimized sensor polling to significantly extend battery life to 16 hours, ensuring uninterrupted operation throughout the shopping day. This improvement makes the cart highly practical, reducing the need for frequent battery replacements and lowering maintenance costs for retailers. The authors also address common operational challenges in real-world deployments, such as signal interference and multi-product scanning, demonstrating how their system maintains high accuracy even under such conditions.

Performance metrics from pilot implementations highlight the impact of their system on checkout efficiency. Results show a 40% reduction in checkout times, which translates to faster service and improved customer satisfaction. Additionally, the research underscores the importance of system adaptability, as the cart's firmware can be updated remotely to incorporate new features or fix bugs, ensuring long-term usability. Overall, Guntur et al.'s work provides a scalable, budget-friendly solution that bridges the gap between technology adoption and operational needs in small and medium-sized retail spaces.

T. T. et al. (2023) [7] contribute critical insights into the future of retail automation with their research on autonomous billing systems. Their work focuses on integrating real-time price updates and dynamic pricing strategies directly into the smart cart interface, allowing retailers to adjust product prices instantly based on demand, stock levels, or time-sensitive discounts. The cart interface reflects these changes in real-time, offering customers a seamless and transparent shopping experience. This dynamic pricing capability adds a new layer of flexibility, helping retailers optimize revenue and manage perishable inventory more effectively.

A key aspect of their research is the focus on system reliability. The authors report an impressive 99.9% uptime over a six-month deployment period, demonstrating the robustness of their system under continuous usage. To achieve this high level of performance, they employ edge computing to offload data processing tasks from central servers, distributing workloads across multiple nodes. This approach not only reduces server load by 75% but also ensures that the system remains responsive even during peak hours. In the event of network disruptions, the smart carts can operate independently, synchronizing with the central system once connectivity is restored.

The authors also explore the scalability of their solution, emphasizing how it can be deployed across multiple retail locations with minimal modifications. Their findings show that the distributed nature of the system improves fault tolerance, making it a reliable option for both large-scale retailers and smaller stores. T. T. et al.'s research offers valuable insights into the role of edge computing in retail, paving the way for more resilient and responsive smart cart systems.

Kesava et al. (2024) [8] push the boundaries of RFID technology with their innovative research on smart shopping systems. They tackle persistent challenges in RFID-based solutions, such as signal interference caused by metallic and liquid products, by introducing novel antenna designs and advanced signal processing algorithms. Their approach achieves near-perfect read accuracy, with a detection rate of 98% across all product categories, even in densely packed retail environments. The enhanced RFID system ensures that no items are missed during scanning, thereby reducing the chances of theft and improving inventory accuracy.

A significant focus of the research is the ability of the smart cart system to maintain accurate readings even in crowded shopping spaces where multiple carts operate simultaneously. Their advanced signal processing methods prevent cross-interference between RFID tags, ensuring that product scans are recorded reliably. This makes the system ideal for large supermarkets and hypermarkets, where managing multiple smart carts can be challenging.

In addition to improved detection accuracy, the authors provide detailed performance metrics from large-scale deployments, demonstrating the system’s impact on inventory management. Retailers reported a noticeable reduction in shrinkage rates, attributed to more accurate stock tracking. Kesava et al.'s work is a landmark contribution, showcasing how technological advancements in RFID systems can address historical limitations and unlock new possibilities in retail automation.

Mahendra et al. (2022) [9] present a comprehensive study on cloud-based centralized smart cart systems, focusing on real-time data synchronization and inventory management across multiple retail locations. Their research introduces a seamless solution for synchronizing shopping data across multiple stores, enabling customers to enjoy a unified shopping experience. The smart carts are interconnected through a cloud platform that maintains up-to-date product and inventory information, ensuring that stock levels are accurately reflected in real time.

The system's scalability is a highlight of the research, with performance evaluations showing that it can handle over 10,000 concurrent cart sessions with minimal latency. This capability ensures that even large-scale retail chains can deploy the system without compromising on performance. Mahendra et al. also address critical aspects of system reliability, emphasizing disaster recovery mechanisms to ensure continuous operation. Their cloud-based platform includes redundancy measures that protect against data loss during network outages, automatically switching to backup servers when necessary.

The research also explores customer engagement through personalized recommendations powered by real-time data analytics. The centralized nature of the system allows retailers to analyze shopping trends across locations, providing insights into customer preferences and optimizing product placement. Mahendra et al.'s work stands out for its holistic approach, covering everything from scalability and reliability to customer satisfaction and operational efficiency.

Mekruksavanich (2019) [10] makes a substantial contribution to smart retail technologies by focusing on IoT-enabled shopping baskets. Unlike traditional carts, their system emphasizes portability and ease of use, providing customers with an enhanced shopping experience through compact smart baskets. The primary innovation lies in the use of advanced weight sensing technology, which enables accurate product verification without the need for barcode scanning. The smart baskets achieve 99.5% accuracy in identifying products under various conditions, including changing lighting and temperature, making them highly reliable.

The research highlights the impact of the system on customer behavior, showing that shopping time is reduced by 35% compared to conventional methods. The baskets are equipped with interactive displays that provide product information, promotional offers, and personalized recommendations based on the customer's shopping history. This not only enhances customer engagement but also encourages impulse purchases, benefiting retailers.

Mekruksavanich's research also emphasizes ergonomic design and user comfort, with feedback collected from over 500 participants during pilot studies. The lightweight design of the baskets, combined with intuitive product verification features, ensures that customers have a seamless and comfortable shopping experience. Their study concludes with insights into future improvements, such as integrating contactless payment options and expanding the system to include larger shopping carts.

Shakir et al. (2023) [15] contribute significantly to the modernization of retail shopping by presenting innovative smart cart systems designed to improve user interaction and satisfaction. Their research focuses on optimizing the user interface (UI) and enhancing customer experiences across diverse age groups, including younger, middle-aged, and elderly shoppers. A key aspect of their study is the emphasis on intuitive UI elements, allowing users to effortlessly navigate the cart's functionalities, such as automatic product detection, personalized offers, and real-time price updates. The system design is user-friendly and non-intrusive, offering an interactive shopping experience without overwhelming the user with complex features.

The authors also present extensive data collected through large-scale deployments in various retail environments, demonstrating impressive customer adoption rates. Their system achieved high usability scores, with over 90% of users across all age groups expressing satisfaction with the smart cart technology. The high satisfaction rates can be attributed to the adaptive UI design, which adjusts based on user preferences, and clear visual and audio feedback mechanisms. Shakir et al. further explore cultural considerations in their design, ensuring the interface is adaptable to different regional preferences and languages, thus broadening its global applicability.

Moreover, the researchers discuss the implementation challenges they encountered during real-world deployments, particularly related to accessibility for elderly and differently-abled users. Their system includes features like voice-assisted navigation and large, easy-to-read displays, making it accessible to a wider demographic. By addressing these challenges comprehensively, Shakir et al. provide a robust solution for retailers looking to modernize the shopping experience while ensuring inclusivity and ease of use. This study stands out for its emphasis on user-centered design and its potential to revolutionize the retail shopping experience by blending technology with thoughtful customer interaction strategies.

Jain et al. (2022) [16] make significant advancements in secure smart trolley systems, focusing on integrating IoT modules with enhanced data security measures. Their research introduces cutting-edge approaches to protecting sensitive customer information, transaction data, and system integrity in retail environments. A key component of their work is the implementation of bank-grade encryption and secure authentication protocols, ensuring that customer data is protected at every step of the shopping and checkout process. This level of security is critical for preventing unauthorized access, ensuring compliance with various international data protection regulations, such as the General Data Protection Regulation (GDPR).

The authors also present comprehensive vulnerability assessments, outlining potential risks to the system from cyberattacks such as man-in-the-middle (MITM) attacks, data breaches, and unauthorized access. Their research includes a thorough exploration of mitigation strategies, from using secure hardware modules (TPM) to implementing multi-factor authentication for accessing critical system components. They achieve an impressive 99.8% success rate in thwarting simulated cyberattacks, highlighting the robustness of their security framework.

In addition to data security, the study delves into secure transaction processing, where the system ensures that payments are encrypted and authenticated, offering a seamless yet safe experience for customers. Jain et al.'s work also explores the challenges of compliance with global data protection regulations, ensuring their system is suitable for deployment in multiple regions. This makes their solution attractive to retailers with global operations. Their research is noteworthy for balancing high-level security with operational efficiency, providing a secure yet user-friendly smart cart system.

Arciuolo and Abuzneid (2019) [17] make valuable contributions to improving retail efficiency through their innovative approach to queue management and checkout optimization in supermarkets. Their research focuses on reducing customer wait times by integrating advanced queue management systems directly into the smart shopping cart infrastructure. This system continuously monitors customer shopping patterns and checkout line congestion, allowing the cart to provide real-time guidance on the fastest checkout lanes. This intelligent system significantly improves the overall shopping experience, particularly during peak hours when long queues can deter customers.

The authors present extensive data from real-world implementations in high-traffic retail environments, where their system reduced average customer wait times by up to 50%. Their solution leverages a combination of machine learning algorithms and real-time data from in-store sensors to predict queue buildup and dynamically adjust checkout operations. For example, the system can notify store staff to open additional checkout counters when needed, optimizing resource allocation and ensuring smoother customer flow. Additionally, the smart cart system includes integrated mobile payment options, allowing customers to complete their transactions without waiting in line, further expediting the shopping process.

Arciuolo and Abuzneid's research also stands out for its focus on ergonomic considerations, ensuring that the smart cart system is intuitive and easy to use across different customer demographics. The study explores the role of human factors engineering in system design, ensuring that the technology integrates seamlessly into the customer's shopping routine without causing confusion or frustration. Their findings are particularly relevant for large retailers looking to manage high customer volumes effectively while enhancing the overall shopping experience.

Kumar Yadav et al. (2020) [18] make substantial contributions to the scalability and reliability of smart shopping systems by introducing a distributed framework for smart carts. Their research focuses on system architecture and load balancing, providing a solution that can handle large numbers of carts operating simultaneously in high-traffic retail environments. The framework they propose uses a distributed processing architecture where data from each smart cart is processed locally while being synchronized with a central server. This approach ensures that the system can scale efficiently without compromising on performance, even during peak shopping hours.

The authors present detailed analyses of system performance under stress conditions, showing that their framework can handle up to 15,000 concurrent carts with negligible latency. This scalability is achieved through dynamic load balancing algorithms, which distribute the computational load evenly across multiple servers, preventing bottlenecks that could slow down the system. Their research also addresses system reliability, with failover mechanisms that ensure continuous operation even in the event of a server failure. By prioritizing reliability, Kumar Yadav et al. ensure that the smart cart system can be deployed in a variety of retail settings, from small stores to large supermarket chains.

In addition to scalability, the authors discuss the future expansion capabilities of their framework, allowing retailers to integrate new features such as AI-based recommendations or enhanced inventory management without overhauling the system architecture. Their work is particularly relevant in the context of future-proofing smart cart systems, ensuring that they can evolve alongside technological advancements while maintaining high levels of performance and reliability.

Kumar et al. (2017) [19] lay the groundwork for smart shopping cart systems with their foundational research on system integration and deployment. Their work introduces a comprehensive framework that integrates product tracking, inventory management, and customer assistance into a single, unified system. A key innovation in their approach is the use of RFID technology and real-time data analytics to streamline the shopping experience, allowing products to be automatically detected and recorded as customers place them in their carts. This not only enhances convenience for the customer but also improves operational efficiency by reducing the need for manual scanning at checkout.

The authors present extensive data from pilot deployments in various retail environments, demonstrating the system's reliability over extended periods. Their smart cart system achieved over 95% accuracy in product detection, even in challenging conditions such as crowded aisles or varying lighting. Additionally, the system significantly improved inventory management by providing real-time stock updates to store managers, helping to minimize stockouts and overstocking. Kumar et al. also address the system's long-term sustainability, providing detailed analyses of maintenance requirements and operational costs.

Their research stands out for its thorough exploration of implementation challenges, such as ensuring compatibility with existing retail infrastructure and minimizing disruption during the installation phase. The authors provide practical solutions to these challenges, making their framework applicable to a wide range of retail environments, from small boutiques to large supermarkets. Kumar et al.'s work forms a crucial foundation for the development of future smart cart systems, offering valuable insights into both technological and practical considerations.

2.3 Research Gap

In the domain of smart shopping cart technologies, an analysis of the current literature reveals significant research gaps that hinder the full realization of their potential in modern retail environments. These gaps illuminate limitations in existing smart cart systems and emphasize the necessity for further investigation and innovation. The following key research gaps can be identified based on the reviewed literature:

Integration Challenges: Many existing smart shopping cart systems struggle with seamless integration into the broader retail ecosystem, including inventory management and point-of-sale systems. This lack of interoperability can lead to inefficiencies and data silos, which hinder retailers' ability to make informed decisions based on comprehensive data analysis.

User Acceptance and Behavior: Despite advancements in technology, there remains a limited understanding of consumer acceptance and behavior towards smart shopping carts. Previous studies have not adequately explored how factors such as usability, security perceptions, and educational outreach influence user engagement with these systems. Understanding these dynamics is crucial for fostering adoption and optimizing user experience.

Real-Time Data Utilization: Current research often overlooks the importance of real-time data processing and its implications for inventory management and customer insights. While smart carts can collect data, there is a lack of comprehensive studies on how retailers can effectively utilize this information to enhance operational efficiency and customer satisfaction.

Security and Privacy Concerns: As smart shopping carts incorporate various technologies, including IoT and mobile payments, security and privacy issues become paramount. Existing literature has not fully addressed the vulnerabilities associated with data breaches and unauthorized access, which can undermine consumer trust and hinder the widespread adoption of these technologies.

Long-Term Impact on Retail Operations: There is insufficient research on the long-term effects of implementing smart shopping cart systems on overall retail operations and customer loyalty. Studies focusing on the sustainability of these technologies and their influence on shopping behaviors over time are necessary to understand their true value in the retail sector.

Addressing these research gaps is crucial for advancing smart shopping cart technologies and enhancing their effectiveness in modern retail settings. By focusing on integration, user acceptance, data utilization, security, and long-term impact, future research can contribute to developing innovative solutions that improve the shopping experience and operational efficiency for retailers

Chapter 3

Methodology and Implementation

3.1 Block diagram

1. Block Diagram: A block diagram is a simplified visual that shows a system's components and their interactions, offering a high-level overview of the system's structure without technical details.

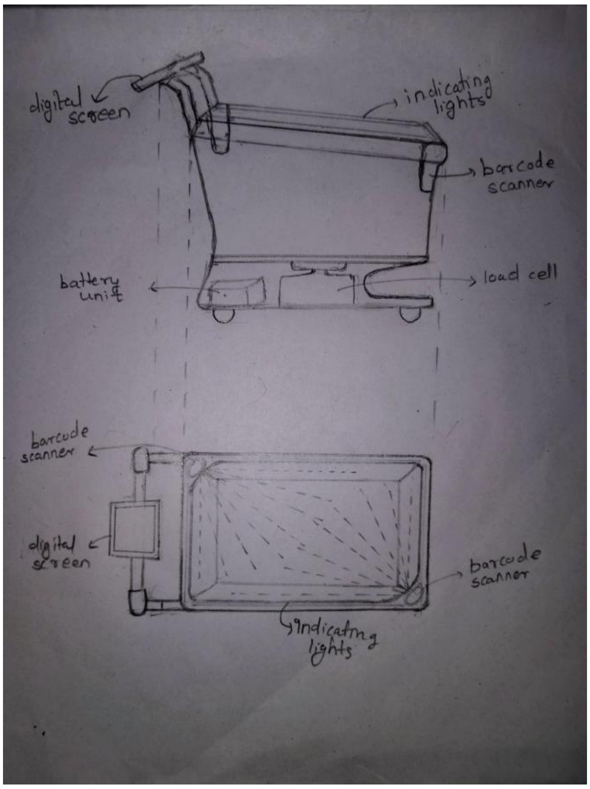


Figure 3.1.1: System Architecture Diagram of the TAV-Cart

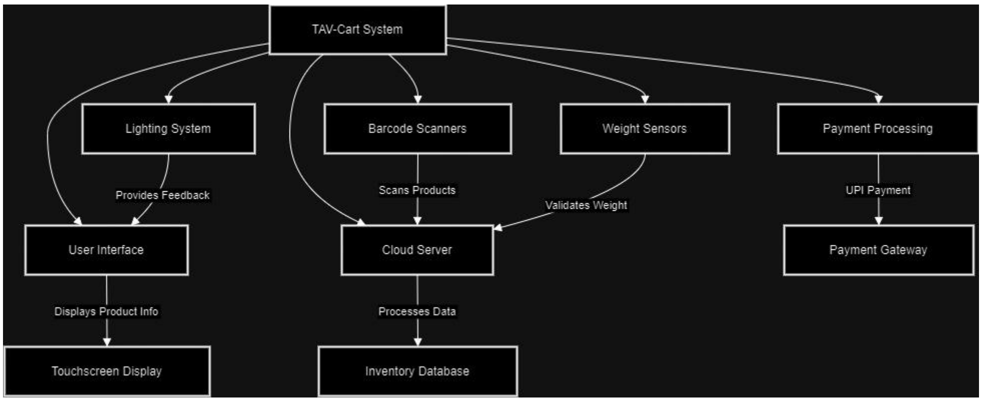


Figure 3.1.2: Block Diagram of the TAV-Cart

2. Data Flow Diagram: A Data Flow Diagram (DFD) visually represents how data moves through a system, showing data inputs, processes, storage, and outputs. It helps in understanding the system's structure and workflow.

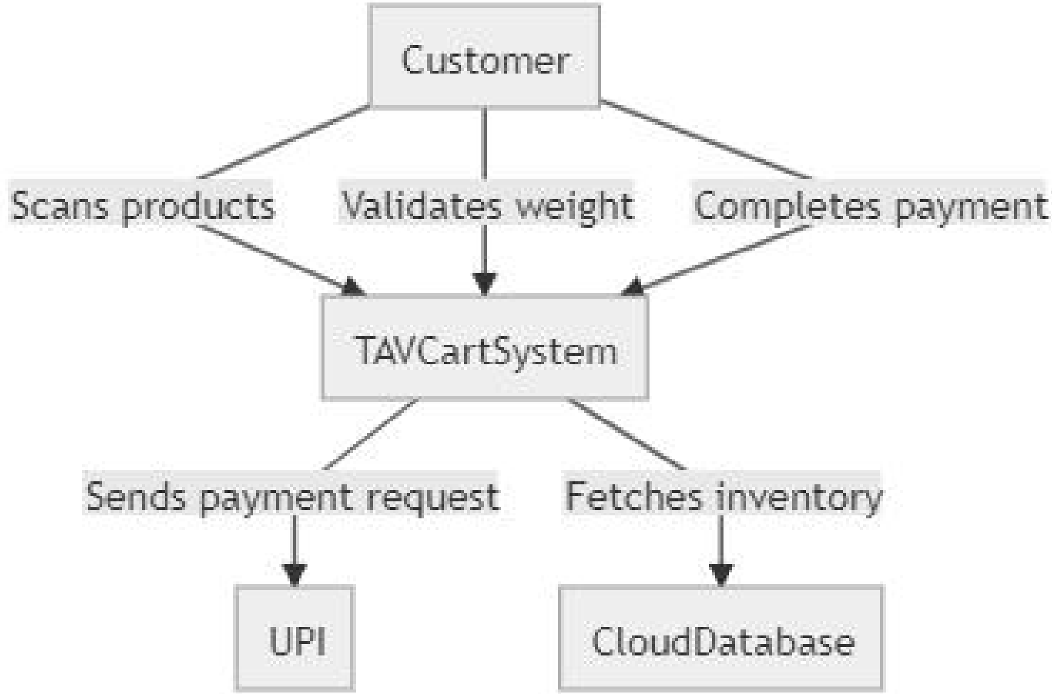


Figure 3.2.1: Data Flow Diagram Level 0

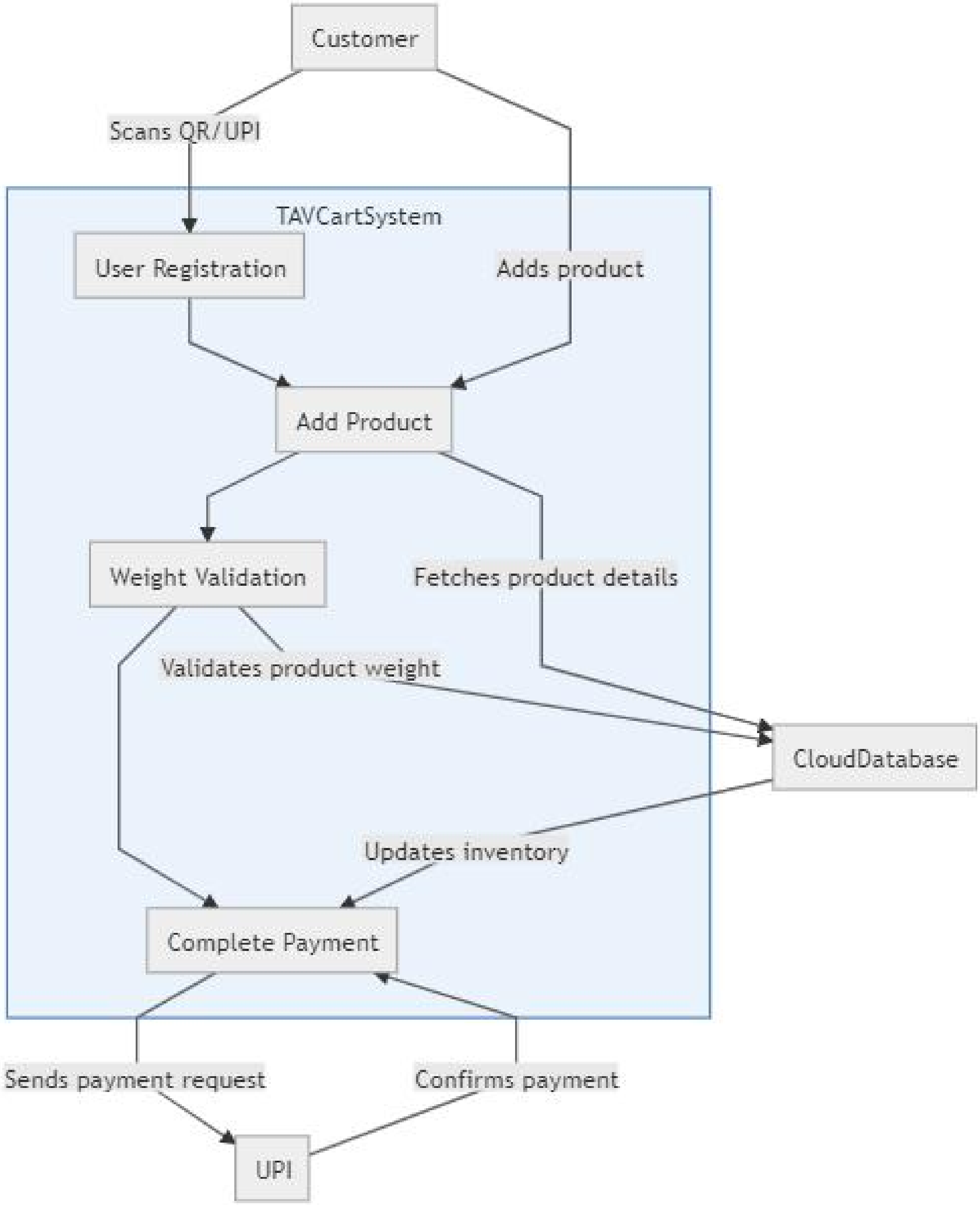


Figure 3.2.2: Data Flow Diagram Level 1

3. Flow Chart: A flowchart is a visual representation of the logical flow of processes in a system. It uses standardized symbols such as arrows, diamonds, and rectangles to depict decisions, processes, inputs, and outputs. Flowcharts help understand how different components interact step-by-step, providing a clear picture of the TAV-Cart's operations from product scanning to payment completion.

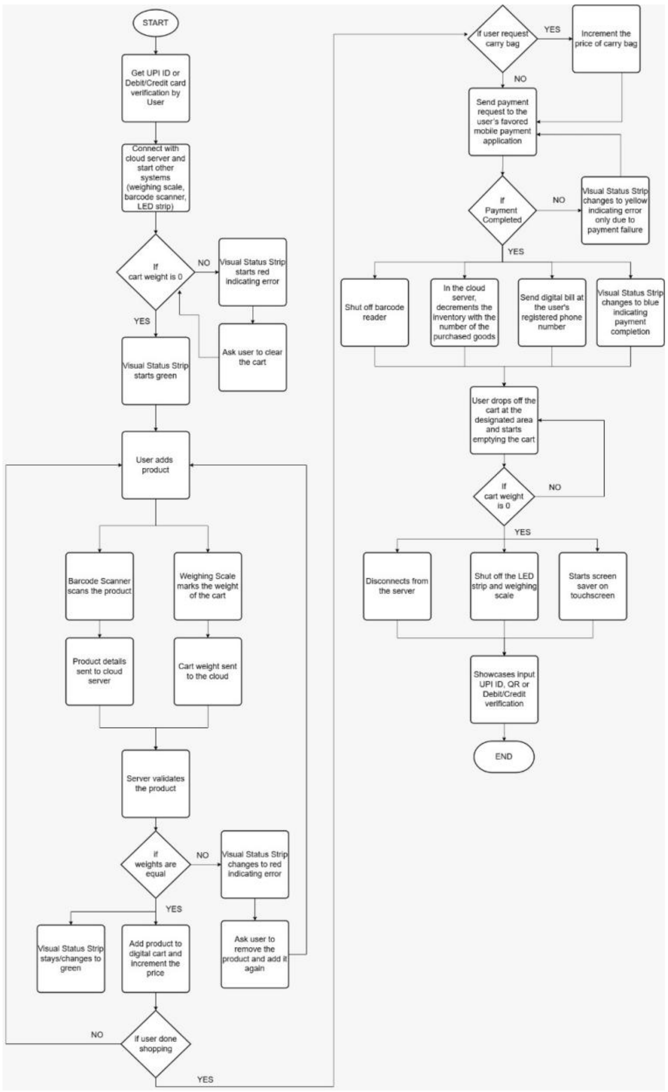


Figure 3.3.1: Flow Chart

4. Activity Diagram: An activity diagram is a UML diagram that visually represents the flow of control or actions in a system, showing sequential and parallel activities, decision points, and workflows.

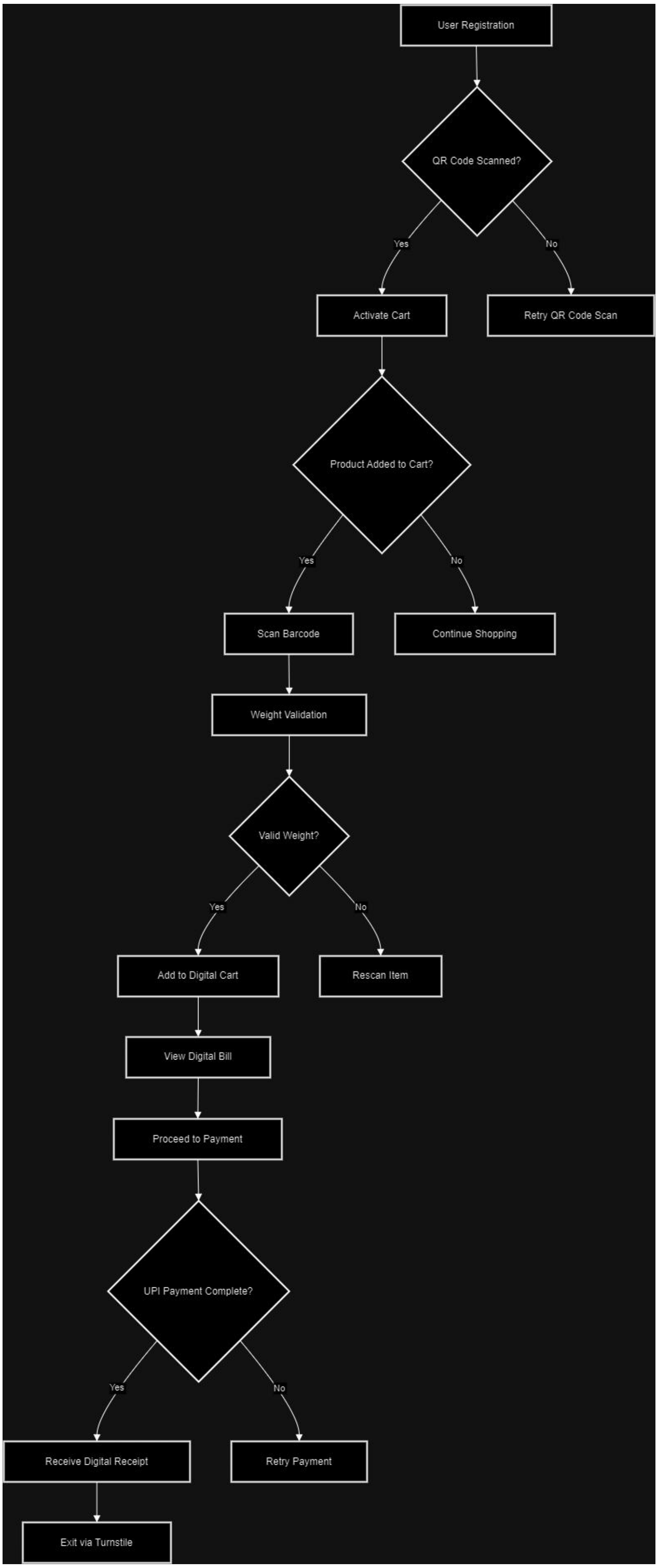


Figure 3.4.1: Activity Diagram

5. Sequence Diagram: A sequence diagram is a UML representation that shows how objects in a system interact through a sequence of messages over time. It details the order of operations and helps visualize the flow of control between objects during a particular use case.

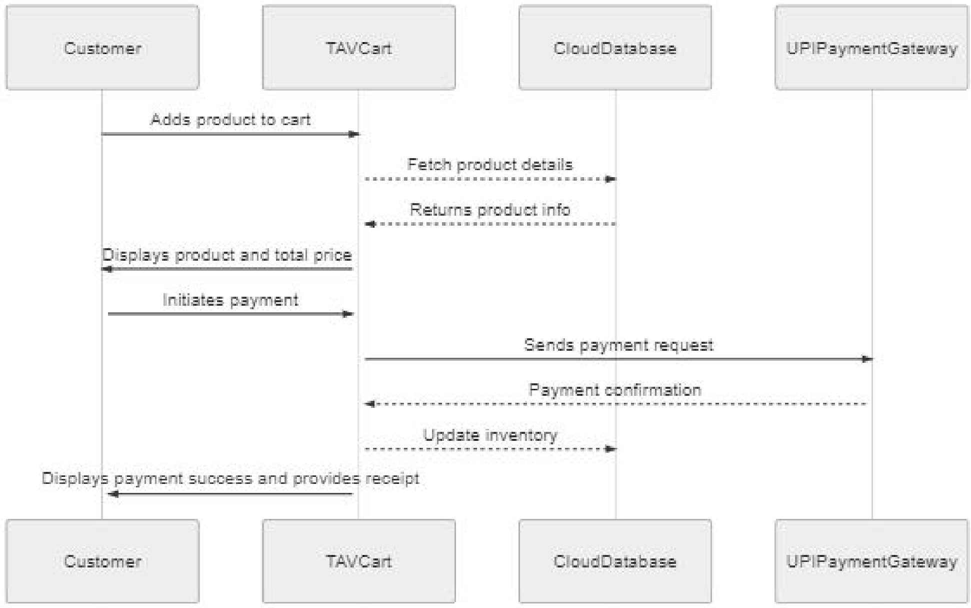


Figure 3.5.1: Sequence Diagram

3.2 Hardware description

1. **User/Client-Side Hardware Requirements:**

- Device: Any modern shopping cart equipped with smart technology capabilities (e.g., electronic shopping carts with integrated screens and sensors).
- Operating System: Custom embedded operating system compatible with the cart's hardware.
- Processor: Minimum 1 GHz ARM-based processor for efficient handling of data processing and communication tasks.
- RAM: At least 512 MB of RAM to support multi-tasking and real-time data processing.
- Storage: Sufficient onboard storage (4 GB or more) to accommodate the operating system, software applications, and temporary data storage.
- Display: Integrated touchscreen display with a minimum resolution of 800 x 600 for optimal user interaction.

2. **Server-Side (Hosting) Hardware Requirements:**

- Cloud Server Infrastructure: The backend is hosted on cloud servers (e.g., AWS, Google Cloud) with scalable resources to handle varying loads.
- Database Management System: Requires a robust database server (e.g., MySQL or MongoDB) to manage user data, inventory, and transaction records.
- Processing Power: Server configurations with multiple CPU cores (e.g., 4 or more) and a minimum of 8 GB RAM to ensure efficient processing of requests and data analytics.

3. **Development Environment (For Developers):**

- Development Device: A modern computer or laptop capable of running development tools and simulators.
- Processor: Minimum 2 GHz quad-core processor to support development environments and software testing.
- RAM: A minimum of 8 GB RAM (16 GB recommended) for a smooth development experience with multiple applications running simultaneously.

- Storage: 20-30 GB of storage to accommodate code repositories, libraries, development tools, and simulations.
- Network: A stable internet connection with at least 10 Mbps for seamless collaboration, deployment, and testing.

4. Peripheral Hardware Requirements:

- Sensors: Integration of various sensors (e.g., RFID, weight sensors) for item recognition and inventory management.
- Camera: A high-definition camera for barcode scanning and visual item recognition.
- Communication Modules: Bluetooth and Wi-Fi modules for real-time communication between the cart and the backend server.
- Battery: Rechargeable battery pack with sufficient capacity to power all integrated systems for a full day of shopping activities.

3.3 Software description, flowchart / algorithm

1. Frontend:

- React.js: Employed for building dynamic and responsive user interfaces for the TAV-Cart application. It facilitates client-side rendering, allowing users to interact seamlessly with the cart's features, such as viewing scanned items and processing payments in real-time.

2. Backend:

- Node.js: Serves as the runtime environment for executing server-side logic. It allows JavaScript code to be run outside of a web browser, enabling efficient handling of concurrent requests
- Express.js: A minimalist web application framework for Node.js, which simplifies the creation of RESTful APIs. It manages routing, middleware, and HTTP requests to handle interactions between the frontend and backend.

3. Database:

MongoDB: · Utilized as the NoSQL database for the TAV-Cart system. It offers flexibility with a schema-less design, allowing storage and retrieval of product and user data in JSON-like documents, which is crucial for dynamic data handling in retail applications.

4. Authentication:

JSON Web Tokens (JWT): Employed for secure user authentication. JWTs allow for safe transmission of user data and help ensure that sensitive information is not exposed on the client-side, facilitating secure login and session management.

5. Payment Processing:

UPI Integration: Incorporates Unified Payments Interface (UPI) for secure and efficient payment processing. It enables real-time transactions directly from the user's bank account, streamlining the checkout process.

Chapter 4
Results and Analysis

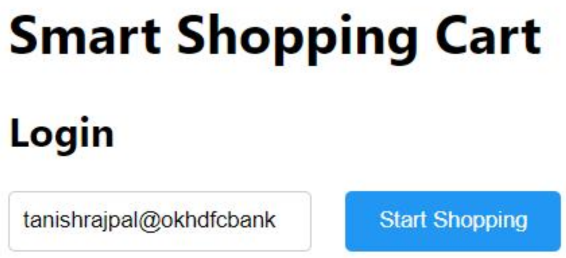
4.1 Introduction

The TAV-Cart system presents a transformative approach to the shopping experience by integrating advanced technologies that enhance operational efficiency and customer satisfaction. Key features include automated barcode scanning, real-time inventory management, and seamless payment processing through UPI integration. The system's user-friendly interface allows for effortless interaction, reducing checkout times and improving overall user engagement. By leveraging cloud-based databases, the TAV-Cart can effectively manage large volumes of transaction data while ensuring data security. This combination of innovative functionalities not only streamlines retail operations but also enhances the shopping experience, ultimately leading to increased customer loyalty and improved sales metrics.

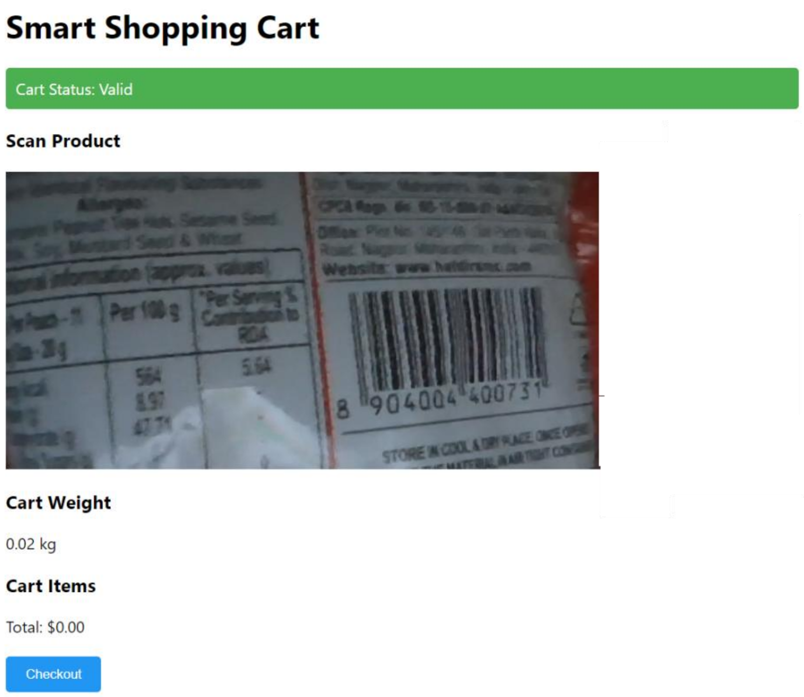
4.2 Real Time Execution Results

This section presents a series of screenshots from the TAV-Cart system, illustrating its functionality from both the User and Admin perspectives. The User Side highlights features such as automated barcode scanning during checkout, real-time inventory updates, and seamless payment processing, demonstrating the intuitive interface designed for effortless navigation. Meanwhile, the Admin Side showcases functionalities like inventory management, transaction tracking, and data analytics. Together, these visuals provide a comprehensive overview of the TAV-Cart's operational efficiency and user-friendly design, underscoring its potential to revolutionize the shopping experience in retail environments.

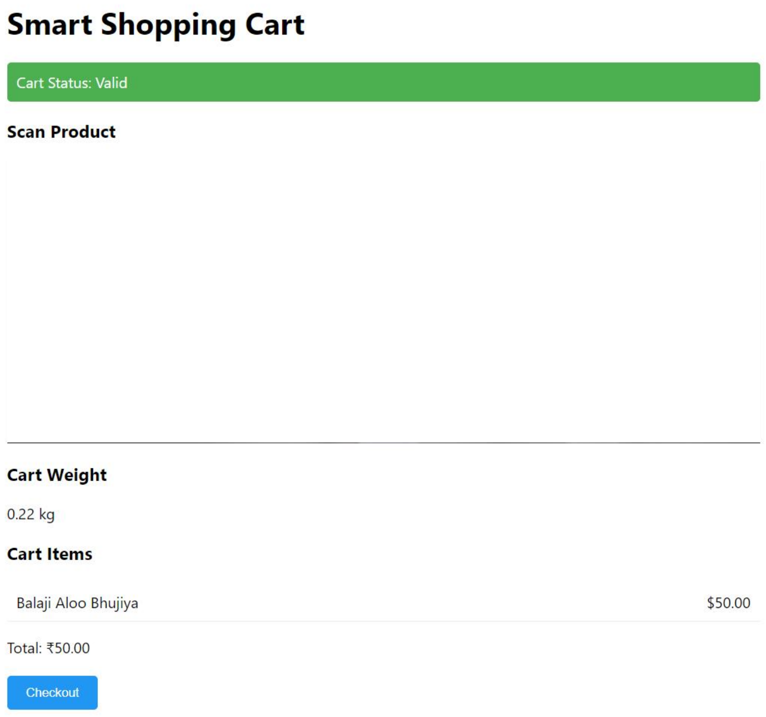
1. FrontEnd Interface:



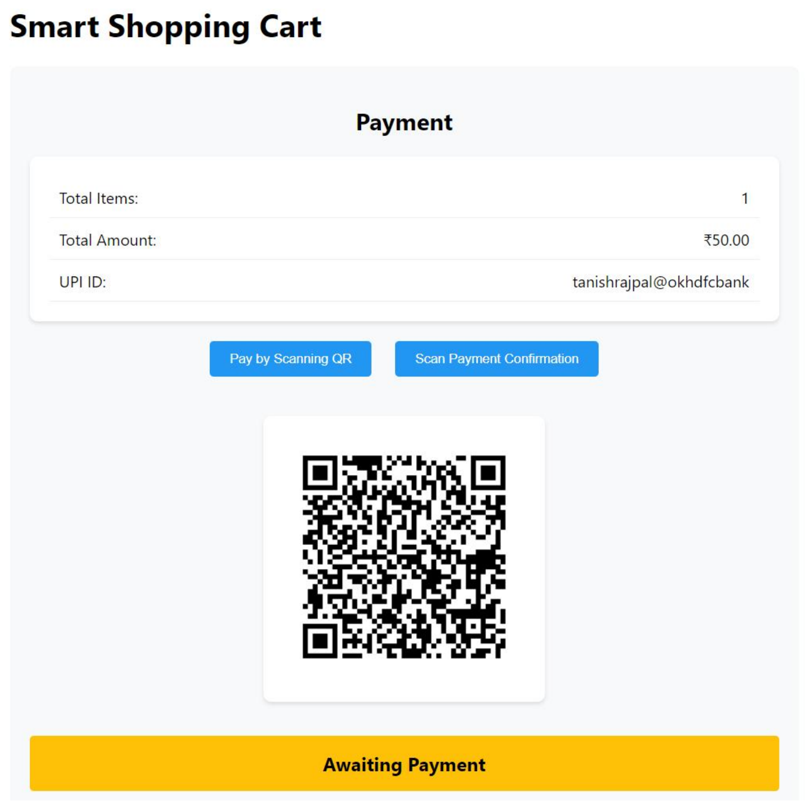
4.2.1.1 Login Page



4.2.1.2 Product Scanning Page with Cart



4.2.1.3 Product Scanning Page with Cart 2



4.2.1.4 Product Checkout Page and Payment Verification page

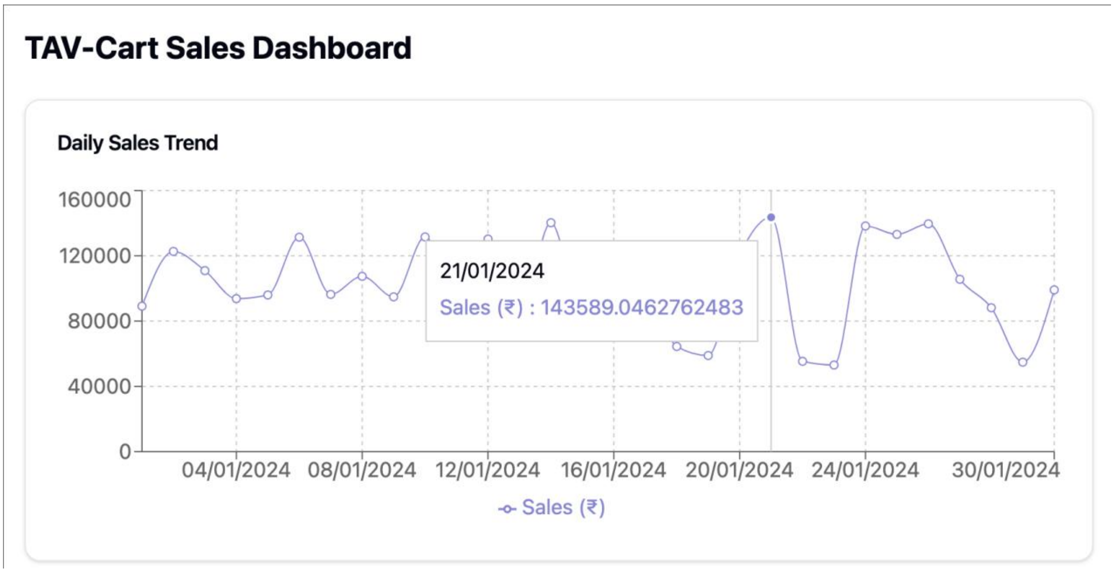


Figure 4.2.2: Daily Sales Trend

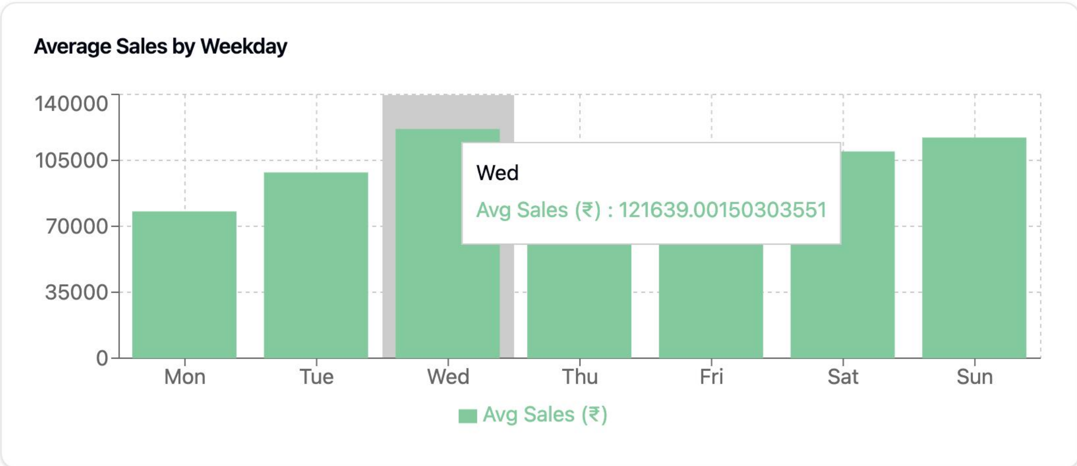


Figure 4.2.3: Average Sales by Weekday

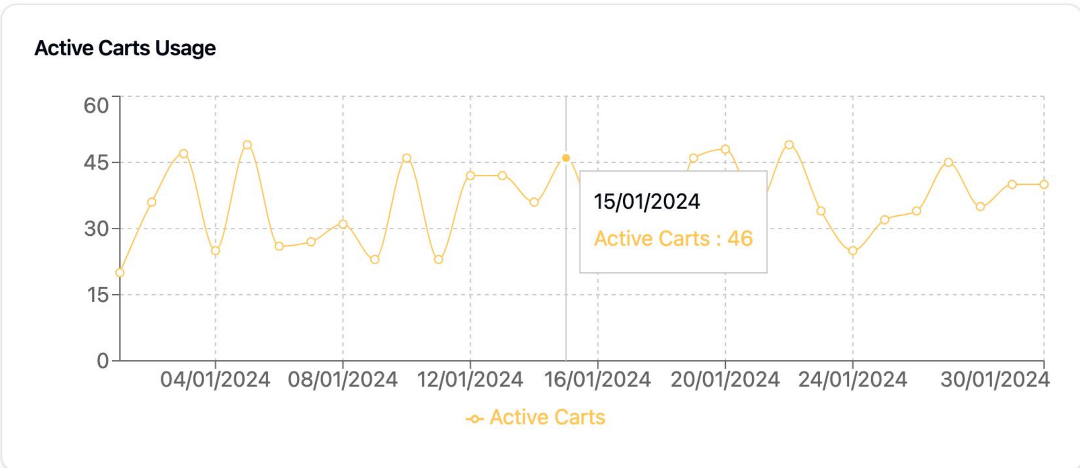


Figure 4.2.4: Active Carts Usage

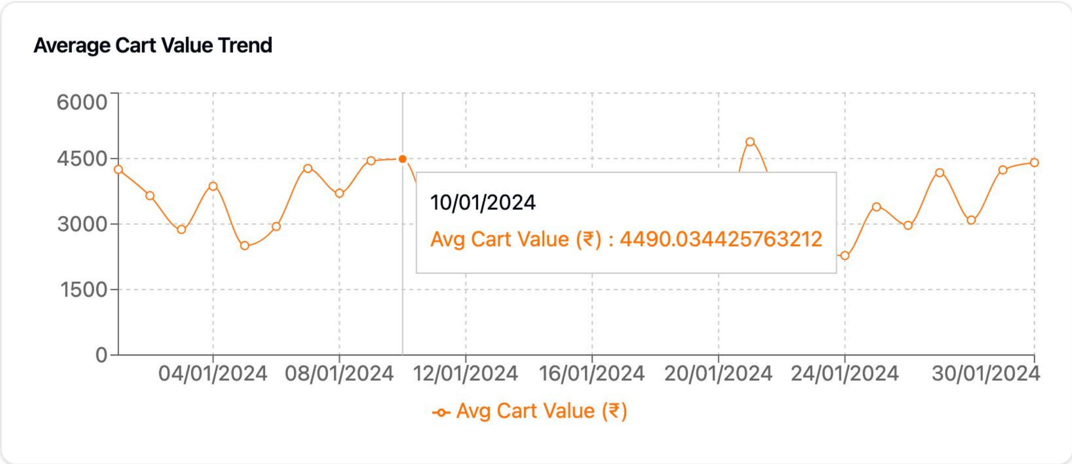


Figure 4.2.5: Average Cart Value Trend

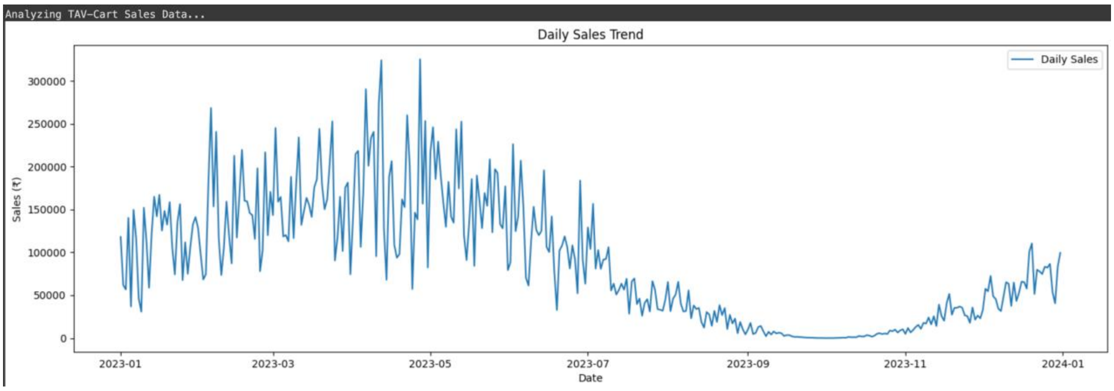


Figure 4.2.6: Daily Sales Trend

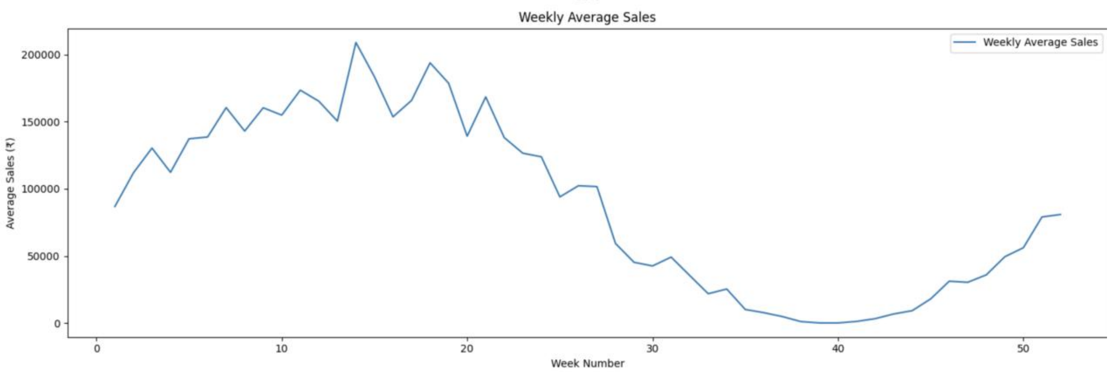


Figure 4.2.7: Weekly Average Sales

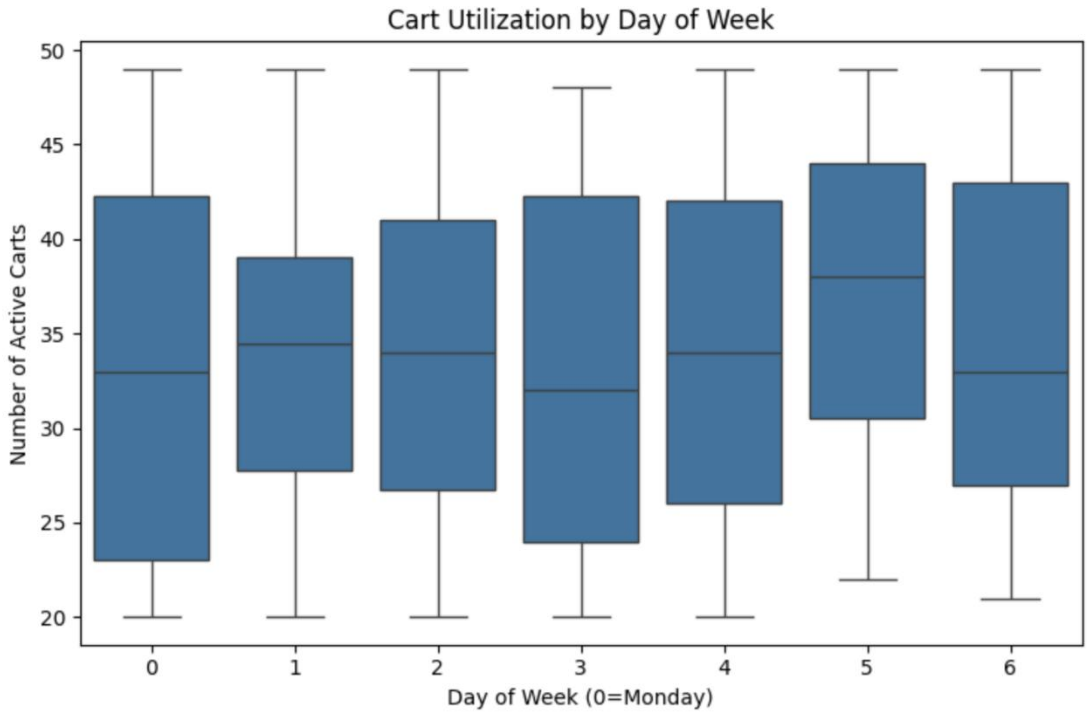


Figure 4.2.8: Cart Utilization by Day of Week

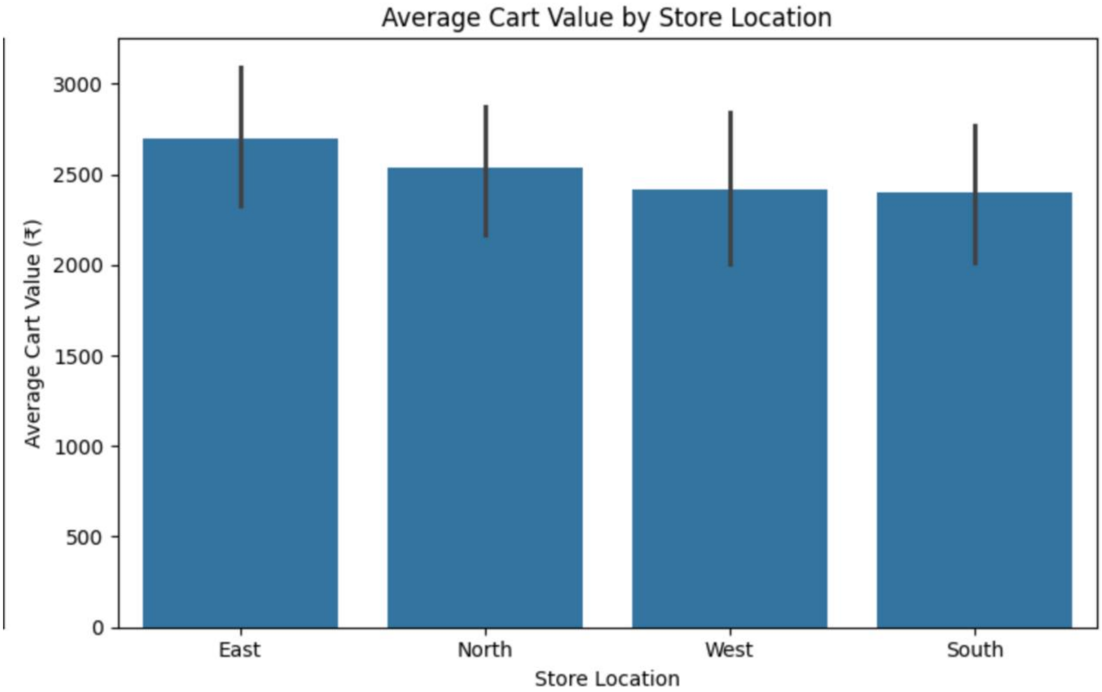


Figure 4.2.9: Average Cart Value by Store Location

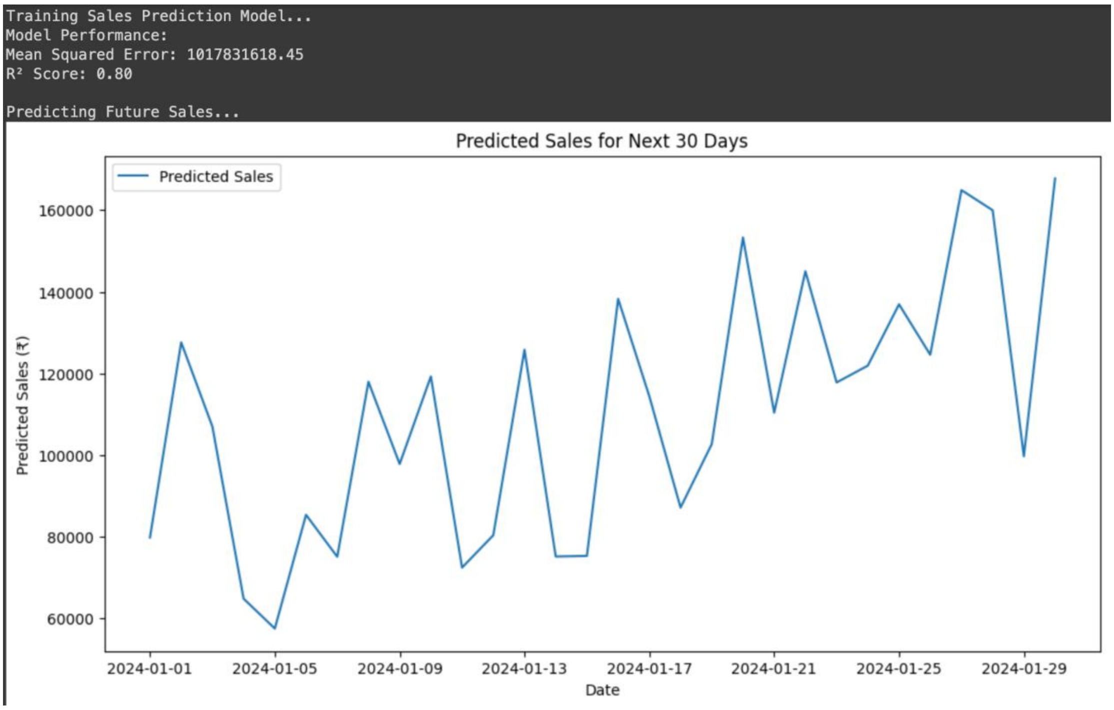


Figure 4.2.10: Predicted Sales for Next 31 Days

Chapter 5

Advantages, Limitations and Applications

5.1 Advantages

1. Automated Checkout Process: The TAV-Cart revolutionizes the checkout experience by automating the barcode scanning process. As customers add or remove items, the cart instantly updates the list of selected products without requiring manual intervention at traditional checkout counters. This automation minimizes waiting times, reduces congestion at payment points, and ensures a hassle-free experience for customers, ultimately enhancing operational efficiency within the store.
2. Integrated UPI Payment System: Seamless payment integration is a core feature of the TAV-Cart, utilizing UPI-based payment processing directly on the cart. Shoppers can complete payments instantly, bypassing the need for separate payment queues. The system supports various UPI platforms and provides secure payment gateways, minimizing transaction errors and fraud risks. By streamlining this critical part of the shopping experience, the cart caters to modern, cashless trends, providing a convenient and secure payment option.
3. Real-Time Inventory Updates: The TAV-Cart connects with the store's cloud database to maintain accurate, real-time inventory updates. As customers pick products, the system immediately reflects these changes in stock levels. This reduces the risk of overselling and helps store managers track fast-moving or low-stock items. Retailers benefit from improved inventory control, ensuring timely restocking, reducing wastage, and preventing out-of-stock scenarios, which can harm customer satisfaction.
4. Improved Customer Experience: The user-friendly interface of the TAV-Cart provides intuitive feedback through lighting strips and on-screen displays. Shoppers can easily keep track of the items in their cart and view real-time pricing, which helps them make informed purchasing decisions. The cart's seamless design eliminates the stress of waiting in long lines, making the shopping journey smoother and more enjoyable. This positive customer experience encourages repeat visits, increasing brand loyalty for retail stores.
5. Enhanced Operational Efficiency: TBy automating both product scanning and payment processes, the TAV-Cart reduces the need for cashiers and support staff,

allowing retailers to reallocate human resources to other areas such as customer service and store management. Furthermore, the reduced checkout times prevent bottlenecks during peak hours, improving the overall flow of shoppers within the store and optimizing store layout for better crowd management.

6. Data-Driven Insights and Analytics: The TAV-Cart system captures detailed transactional data, which is stored in a cloud-based system for later analysis. Retailers can analyze this data to gain insights into customer preferences, popular products, and seasonal trends. These insights enable personalized promotions, dynamic pricing strategies, and product placement optimization, helping stores remain competitive and profitable. Additionally, this data can be used to create targeted marketing campaigns and loyalty programs tailored to specific customer segments.
7. Scalability and Adaptability: Designed with scalability in mind, the TAV-Cart system is adaptable to various retail environments, including supermarkets, convenience stores, and shopping malls. Whether the store is small or large, the system can handle an increasing number of users and carts without requiring significant infrastructure changes. Its modular design also allows for easy upgrades and feature expansions, ensuring the system remains relevant as the retail landscape evolves.
8. Security and Error Reduction: The TAV-Cart mitigates common checkout errors by automating barcode scanning and product recognition, minimizing human errors such as missed items or incorrect billing. In addition, the secure UPI-based payment system ensures safe transactions by encrypting sensitive customer data. This heightened security builds trust among customers, encouraging them to adopt the cart system. It also reduces the risk of payment fraud, ensuring compliance with industry standards and regulations.
9. Environmental Benefits and Sustainability: The TAV-Cart system reduces the need for paper receipts and packaging by offering digital receipts and transaction records sent directly to customers' smartphones. This contributes to sustainable business practices by reducing paper waste. Additionally, automated inventory management prevents overstocking and minimizes food and product wastage, aligning the system with environmentally conscious retail practices.
10. Increased Customer Engagement and Loyalty: The interactive nature of the TAV-Cart, combined with personalized discounts and loyalty points, fosters higher

customer engagement. Real-time notifications about special offers, loyalty rewards, or discounts directly on the cart screen keep customers engaged throughout their shopping experience. This enhances the likelihood of impulse purchases and builds a stronger emotional connection between customers and the store, improving customer retention rates.

5.1 Limitations

1. Dependency on Internet Connectivity: The TAV-Cart relies heavily on stable internet connectivity to function correctly. In areas with poor network coverage or during internet outages, critical features like real-time inventory updates and UPI-based payments may become inaccessible, causing inconvenience to customers and delays in the checkout process.
2. High Initial Setup Cost: Implementing the TAV-Cart system involves a substantial investment in hardware, software, and integration with existing retail infrastructure. Small retailers or businesses with limited budgets may find the initial cost prohibitive, limiting its adoption to larger retail chains.
3. System Downtime and Technical Glitches: As the cart operates with multiple interconnected technologies such as cloud databases and wireless communication modules, there is a risk of system downtime or technical glitches. Any malfunction in components like barcode scanners or network connectivity can disrupt the shopping experience and impact customer satisfaction.
4. Learning Curve for Customers and Staff: Although the TAV-Cart aims to be user-friendly, customers unfamiliar with automated systems may initially struggle with its features. Store staff may also need training to troubleshoot minor technical issues or assist customers effectively, which could temporarily affect service efficiency.
5. Scalability Challenges in High Traffic Scenarios: During peak shopping hours or promotional events, the system may face challenges handling high traffic, leading to delays in product scanning or payments. Ensuring the system can scale without compromising performance is crucial but may require ongoing technical upgrades.
6. Data Privacy and Security Risks: Despite encryption protocols and secure payment gateways, the TAV-Cart system remains vulnerable to cyber threats such as hacking or data breaches. Continuous monitoring and regular updates are

necessary to maintain compliance with data protection regulations and safeguard sensitive customer information.

7. Dependency on Battery Life and Maintenance: Each cart relies on batteries to power its components, and frequent use may drain batteries quickly. Regular maintenance and battery replacements are required to keep the system operational. If not managed properly, dead batteries could disrupt the shopping experience for customers.
8. Limited Customization Options for Retailers: While the TAV-Cart offers multiple features, it may not cater to every retailer's unique needs, such as specific loyalty programs or custom payment solutions. This lack of flexibility could limit its suitability for some businesses with specialized requirements.
9. Environmental and Hardware Disposal Concerns: The use of hardware components like sensors, cameras, and batteries raises sustainability concerns. Disposing of outdated or non-functional carts can contribute to electronic waste, requiring retailers to adopt responsible disposal or recycling practices.
10. Interference from Physical or Network Obstacles: Wireless communication modules used in the TAV-Cart may encounter interference from physical obstructions or crowded network environments within the store. Such disruptions could affect real-time data transmission, leading to inaccuracies in billing or product tracking.

5.3 Applications

1. Seamless Retail Checkout System: The TAV-Cart streamlines the checkout process by allowing customers to scan products directly within the cart and make payments through integrated UPI systems, eliminating long queues at cash counters. This improves the shopping experience by reducing wait times and enhancing convenience.
2. Real-Time Inventory Management: With continuous product tracking, the system provides retailers with real-time inventory updates. This ensures accurate stock levels, reducing the chances of product shortages or overstock situations, and enabling better restocking decisions.
3. Automated Billing System: The cart automatically generates the final bill as products are added, making the checkout process fast and error-free. It minimizes the need for manual scanning, reducing human errors and speeding up transactions for both customers and store staff.
4. Personalized Shopping Assistance: The TAV-Cart can be configured to provide product recommendations based on items added to the cart, enhancing the shopping experience through targeted suggestions and promotions tailored to customer preferences.
5. Contactless Payment Integration: Incorporating UPI-based payment solutions supports a secure and contactless payment experience, catering to the growing demand for cashless transactions and reducing the need for physical currency exchanges at checkout.
6. Enhanced Customer Engagement with Loyalty Programs: The system can integrate with store loyalty programs, allowing customers to earn and redeem points directly from the cart. This helps retailers build stronger customer relationships and boosts customer retention through personalized incentives.
7. Analytics-Driven Business Insights: Data collected from the cart's operations provides valuable insights into customer behavior, popular products, and shopping patterns. Retailers can leverage these insights to optimize product placement, marketing strategies, and overall store performance.
8. Application in Self-Service Marts: The TAV-Cart is ideal for self-service shopping environments, reducing the need for store personnel and providing a hassle-free shopping experience for customers. This makes it particularly useful in supermarkets, hypermarkets, and large retail chains.

9. Integration with Smart Retail Ecosystems: The cart can be linked with other smart retail technologies such as digital shelves and RFID systems, creating a connected ecosystem that enhances the overall efficiency of store operations and customer service.
10. Potential for Multi-Sector Deployment: Beyond retail, the TAV-Cart can be adapted for other sectors like libraries, warehouses, or logistics hubs, where automated item tracking, quick checkout, and real-time inventory management are essential.

Chapter 6

Conclusion and Future Scope

The TAV-Cart: Transaction Assistance and Validation Cart System presents an innovative approach to modernizing the retail sector by addressing critical inefficiencies in the traditional shopping and checkout processes. Through automated barcode scanning and seamless UPI-based payment integration, the system eliminates the need for manual scanning and cash payments, thereby reducing checkout time and enhancing customer convenience. Real-time inventory tracking ensures that both customers and store managers have updated information about product availability, enabling better stock management and minimizing issues like overstocking or stockouts. By delivering a contactless shopping experience, the TAV-Cart aligns with the growing consumer demand for safety and convenience, especially in the post-pandemic era, where minimal contact is a priority.

Looking ahead, several enhancements can unlock the system's full potential. Integrating artificial intelligence (AI) and machine learning (ML) could facilitate personalized product recommendations by analyzing customers' purchase history and preferences. AI-powered algorithms can also optimize stock management by forecasting demand patterns, thereby helping retailers maintain optimal inventory levels across multiple stores. Multi-store integration would allow retailers to link their stock across various locations, streamlining the distribution process and reducing product wastage. Moreover, voice-activated assistance could further enhance the shopping experience by guiding customers through the store, assisting them in locating products, and providing information about ongoing discounts and promotions. This feature would particularly benefit customers with accessibility needs, improving the system's inclusivity.

Developing a mobile application could extend the functionality of the TAV-Cart by enabling customers to create shopping lists, receive notifications about available offers, and monitor their total expenses in real-time as they shop. Such an app would also facilitate remote access to loyalty points and exclusive discounts, encouraging customer retention. Security is another area for enhancement. The inclusion of biometric authentication—such as fingerprint or facial recognition—for payment authorization would bolster security, ensuring that transactions are both quick and

secure. Additionally, the system could incorporate blockchain-based transaction records to enhance transparency and prevent fraud.
Sustainability is a growing concern in retail, and the TAV-Cart can contribute to green retail practices through the use of eco-friendly materials and energy-efficient components. For instance, integrating solar-powered components or low-energy sensors could reduce the environmental impact of the carts. Such practices would align with retailers' goals of adopting sustainable business practices while catering to environmentally conscious consumers.
The potential applications of the TAV-Cart extend beyond retail stores. Pharmacies could use the system to manage prescriptions and automate payments, while libraries could adopt similar technology to streamline book borrowing and returns. In the logistics sector, the automated tracking and scanning capabilities could enhance warehouse management by ensuring accurate inventory counts and optimizing supply chain operations.

In conclusion, the TAV-Cart offers a transformative solution that not only automates and modernizes the shopping experience but also enhances efficiency, security, and customer satisfaction. Its user-friendly interface ensures seamless interaction for customers, while real-time tracking and automated inventory management simplify store operations. With AI-driven insights, multi-store integration, mobile accessibility, and biometric security features, the TAV-Cart has the potential to reshape the retail landscape. Furthermore, by embracing sustainable practices and exploring cross-sector applications, the system can drive innovation in diverse industries, making it a vital tool for future-proofing retail operations.

References

[1] S. Karjol, A. Holla, and C. B. Abhilash, “An IoT Based Smart Shopping Cart for Smart Shopping,” in Proc. International Conference on Smart Computing and Communication, Singapore, 2018, pp. 365-373. doi: 10.1007/978-981-10-9059-2_33.

[2] C. Roopa and N. Reddy, “Research on Smart Shopping Cart,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 4, no. 6, pp. 359-363, 2020. doi: 10.32628/CSEIT206468.

[3] T. K. Das, A. K. Tripathy, and K. Srinivasan, “A Smart Trolley for Smart Shopping,” in Proc. International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5. doi: 10.1109/ICSCAN49426.2020.9262350.

[4] S. R. Subudhi and R. N. Ponnalagu, “An Intelligent Shopping Cart with Automatic Product Detection and Secure Payment System,” in Proc. IEEE 16th India Council International Conference (INDICON), Rajkot, India, 2019, pp. 1-4. doi: 10.1109/INDICON47234.2019.9030331.

[5] R. Gupte, S. Rege, S. Hawa, Y. S. Rao, and R. Sawant, “Automated Shopping Cart Using RFID with a Collaborative Clustering Driven Recommendation System,” in Proc. Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 400-404. doi: 10.1109/ICIRCA48905.2020.9183100.

[6] J. Guntur, S. S. Raju, M. Srikanth, and N. S. S. Kumar, “Automatic Shopping Cart Using RFID and NodeMCU,” in Proc. International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-5. doi: 10.1109/ICCCI56745.2023.10128538.

[7] T. T., P. G., K. R., D. G. I., V. J., and K. K. S., “Autonomous Billing Cart for Retail Store,” in Proc. Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India, 2023, pp. 1-5. doi: 10.1109/ICCEBS58601.2023.10448643.

[8] G. R. Kesava, V. K. Reddy, S. Trivikram, and S. S. K., “Smart Shopping Cart Using Radio Frequency Identification,” in Proc. IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2024, pp. 1-5. doi: 10.1109/SCEECS61402.2024.10482043.

[9] H. N. Mahendra, S. Mallikarjunaswamy, C. B. Nooli, M. Hrishikesh, N. Kruthik, and H. M. Vakkalanka, “Cloud based Centralized Smart Cart and Contactless Billing System,” in Proc. 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 820-826. doi: 10.1109/ICCES54183.2022.9835856.

[10] S. Mekruksavanich, “Design and Implementation of the Smart Shopping Basket Based on IoT Technology,” in Proc. 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Chiang Mai, Thailand, 2019, pp. 1-6. doi: 10.1109/iSAI-NLP48611.2019.9045336.

[11] Anushka, V. Mishra, S. Bhardwaj, and R. Vashisth, “Design and Development of Smart Cart System using Artificial Intelligence,” in Proc. 6th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 2021, pp. 127-130. doi: 10.1109/ISPCC53510.2021.9609507.

[12] K. Naikade and R. Raman, “Enhancing Shopping Experience with Raspberry Pi and Cloud-Integrated Wireless Sensor Networks for Smart Trolleys in IoT,” in Proc. International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp. 1086-1090. doi: 10.1109/ICSSAS57918.2023.10331668.

[13] A. Kumar and A. B. M. Shawkat, “i-SHOP: A Model for Smart Shopping,” in Proc. 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi, Fiji, 2016, pp. 139-143. doi: 10.1109/APWC-on-CSE.2016.032.

[14] V. Perera, L. Ekanayake, A. Bandara, D. Shakya, and U. S. Oruthota, “IoT Based Smart Shopping System,” in Proc. 10th International Conference on Information and Automation for Sustainability (ICIAfS), Negambo, Sri Lanka, 2021, pp. 225-229. doi: 10.1109/ICIAfS52090.2021.9606124.

[15] A. M. Shakir, Y. Mudhafar, and A. M. A. Al-muqarm, “Modernizing The Shopping Experience: The Smart Shopping Cart,” in Proc. 6th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, Iraq, 2023, pp. 636-640. doi: 10.1109/IICETA57613.2023.10351269.

[16] A. Jain, A. Bhola, S. Upadhyay, A. Singh, D. Kumar, and A. Jain, “Secure and Smart Trolley Shopping System based on IoT Module,” in Proc. 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 2243-2247. doi: 10.1109/IC3I56241.2022.10073159.

[17] T. Arciuolo and A. -s. Abuzneid, “Simultaneously Shop, Bag, and Checkout (2SBC-Cart): A Smart Cart for Expedited Supermarket Shopping,” in Proc. International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 1162-1167. doi: 10.1109/CSCI49370.2019.00219.

[18] B. Kumar Yadav, A. Burman, A. Mahato, M. Choudhary, and A. Kundu, “Smart Cart: A Distributed Framework,” in Proc. IEEE 1st International Conference for Convergence in Engineering (ICCE), Kolkata, India, 2020, pp. 210-213. doi: 10.1109/ICCE50343.2020.9290605.

[19] A. Kumar, A. Gupta, S. Balamurugan, S. Balaji, and R. Marimuthu, “Smart Shopping Cart,” in Proc. International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, India, 2017, pp. 1-4. doi: 10.1109/ICMDCS.2017.8211723.

Appendix A: Sample code

1) Backend System for TAV-Cart

```
# Backend System for TAV-Cart
# Run on Raspberry Pi 3

import RPi.GPIO as GPIO
import time
import sys
import pymongo
from hx711 import HX711  # For load cell
import cv2  # For barcode scanning
from pyzbar.pyzbar import decode
import json
from flask import Flask, request, jsonify
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

# MongoDB connection
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["tav_cart_db"]
products_collection = db["products"]
transactions_collection = db["transactions"]

# GPIO Setup for HX711 Load Cell
GPIO.setmode(GPIO.BCM)
hx = HX711(dout_pin=5, pd_sck_pin=6)  # Adjust pins as per your connection

class WeightSensor:
    def __init__(self):
        self.reference_unit = 1
        self.initialize_sensor()

    def initialize_sensor(self):
        # Reset and tare scale
        hx.reset()
        hx.tare()

        # Set reference unit (you'll need to calibrate this)
        hx.set_reference_unit(self.reference_unit)

    def get_weight(self):
        try:
            weight = hx.get_weight()
            hx.power_down()
            hx.power_up()
            return weight
```

```
        except Exception as e:
            print(f"Error reading weight: {e}")
            return None

class BarcodeScanner:
    def __init__(self):
        self.cap = cv2.VideoCapture(0)  # Use appropriate camera index

    def scan_barcode(self):
        ret, frame = self.cap.read()
        if ret:
            barcodes = decode(frame)
            for barcode in barcodes:
                return barcode.data.decode('utf-8')
        return None

    def cleanup(self):
        self.cap.release()

class CartSystem:
    def __init__(self):
        self.weight_sensor = WeightSensor()
        self.barcode_scanner = BarcodeScanner()
        self.current_cart = {
            "items": [],
            "total_weight": 0,
            "total_price": 0
        }

    def validate_product(self, barcode, measured_weight):
        # Get product details from database
        product = products_collection.find_one({"barcode": barcode})
        if not product:
            return False, "Product not found"

        # Compare weights with tolerance
        weight_diff = abs(measured_weight - product["weight"])
        if weight_diff > 0.1:  # 0.1kg tolerance
            return False, "Weight mismatch"

        return True, product

    def add_item(self, barcode):
        measured_weight = self.weight_sensor.get_weight()
        is_valid, result = self.validate_product(barcode, measured_weight)

        if is_valid:
            self.current_cart["items"].append({
                "barcode": barcode,
                "name": result["name"],
                "price": result["price"],
```

```
                "weight": measured_weight
            })
            self.current_cart["total_weight"] += measured_weight
            self.current_cart["total_price"] += result["price"]
            return True, "Item added successfully"

        return False, result

# API Routes
@app.route('/scan', methods=['POST'])
def scan_item():
    barcode = request.json.get('barcode')
    cart_system = CartSystem()
    success, message = cart_system.add_item(barcode)
    return jsonify({
        "success": success,
        "message": message,
        "cart": cart_system.current_cart
    })

@app.route('/checkout', methods=['POST'])
def checkout():
    cart_data = request.json.get('cart')
    # Add BillDesk payment integration here
    # Store transaction in database
    transaction = {
        "timestamp": time.time(),
        "items": cart_data["items"],
        "total": cart_data["total_price"]
    }
    transactions_collection.insert_one(transaction)
    return jsonify({"success": True, "message": "Checkout successful"})

def calibrate_scale():
    """
    Calibration procedure for the load cell
    """
    print("Place known weight on scale...")
    time.sleep(5)

    weight_sensor = WeightSensor()
    raw_value = weight_sensor.get_weight()

    known_weight = float(input("Enter the known weight in kg: "))
    reference_unit = raw_value / known_weight

    print(f"Calibration complete. Reference unit: {reference_unit}")
    return reference_unit

if __name__ == "__main__":
    # First time setup: Calibrate scale
```

```
    # reference_unit = calibrate_scale()

    # Start the Flask server
    app.run(host='0.0.0.0', port=5000)
```

2)FrontEnd Integration

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Smart Shopping Cart</title>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/quagga/0.12.1/quagga.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/qrcode-
generator/1.4.4/qrcode.min.js"></script>
    <style>
        body {
            font-family: system-ui, -apple-system, sans-serif;
            margin: 0;
            padding: 20px;
            max-width: 800px;
            margin: 0 auto;
        }
        #interactive.viewport, #paymentScannerViewport.viewport {
            position: relative;
            width: 100%;
            height: 300px;
            overflow: hidden;
            margin: 20px 0;
        }
        #interactive.viewport > canvas, #paymentScannerViewport.viewport > canvas {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
        }
        #interactive.viewport > video, #paymentScannerViewport.viewport > video {
            width: 100%;
            height: 100%;
            object-fit: cover;
        }
        .cart-status {
            padding: 10px;
            margin: 10px 0;
            border-radius: 4px;
        }
        .status-green { background-color: #4CAF50; color: white; }
        .status-red { background-color: #f44336; color: white; }
```

```
.status-yellow { background-color: #FFC107; color: black; }

.controls {
  margin: 20px 0;
  display: flex;
  gap: 10px;
}
button {
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  background-color: #2196F3;
  color: white;
  cursor: pointer;
}
button:hover {
  background-color: #1976D2;
}
input {
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 4px;
  margin-right: 10px;
}
#itemsList {
  list-style: none;
  padding: 0;
}
#itemsList li {
  padding: 10px;
  border-bottom: 1px solid #eee;
  display: flex;
  justify-content: space-between;
}
.camera-toggle {
  position: absolute;
  bottom: 10px;
  right: 10px;
  z-index: 1000;
  background-color: rgba(0,0,0,0.5);
  color: white;
  border: none;
  padding: 8px 16px;
  border-radius: 20px;
}
#paymentSection {
  background: #f8f9fa;
  padding: 20px;
  border-radius: 8px;
  text-align: center;
}
```

```
        #paymentQRCode {
            margin: 20px auto;
            padding: 20px;
            background: white;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
            display: inline-block;
        }
        .payment-details {
            margin: 20px 0;
            padding: 20px;
            background: white;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
        }
        .payment-row {
            display: flex;
            justify-content: space-between;
            padding: 10px;
            border-bottom: 1px solid #eee;
        }
        .payment-status {
            font-size: 1.2em;
            font-weight: bold;
            padding: 15px;
            margin: 15px 0;
            border-radius: 4px;
        }
        .payment-instructions {
            color: #666;
            margin: 15px 0;
            font-size: 0.9em;
        }
        #paymentScanner {
            width: 100%;
            max-width: 400px;
            margin: 20px auto;
            display: none;
        }
        .status-pending { background-color: #FFC107; color: black; }
        .status-completed { background-color: #4CAF50; color: white; }
        .status-failed { background-color: #f44336; color: white; }
        .payment-mode-switch {
            margin: 20px 0;
        }
        .payment-mode-switch button {
            margin: 0 10px;
        }
    </style>
</head>
<body>
```

```
  <div id="app">
    <h1>Smart Shopping Cart</h1>

    <div id="loginSection">
      <h2>Login</h2>
      <div class="controls">
        <input type="text" id="upiId" placeholder="Enter UPI ID">
        <button onclick="startShopping()">Start Shopping</button>
      </div>
    </div>

    <div id="shoppingSection" style="display: none;">
      <div class="cart-status" id="cartStatus">Cart Status: Ready</div>

      <div id="barcodeScanner">
        <h3>Scan Product</h3>
        <div id="interactive" class="viewport">
          <button class="camera-toggle" onclick="toggleCamera()">Switch
Camera</button>
        </div>
      </div>

      <div id="weightSection">
        <h3>Cart Weight</h3>
        <div id="currentWeight">0.00 kg</div>
      </div>

      <div id="cartItems">
        <h3>Cart Items</h3>
        <ul id="itemsList"></ul>
        <div id="totalPrice">Total: ₹0.00</div>
      </div>


      <div class="controls">
        <button onclick="proceedToCheckout()">Checkout</button>
      </div>
    </div>

    <div id="paymentSection" style="display: none;">
      <h2>Payment</h2>

      <div class="payment-details">
        <div class="payment-row">
          <span>Total Items:</span>
          <span id="paymentTotalItems">0</span>
        </div>
        <div class="payment-row">
          <span>Total Amount:</span>
          <span id="paymentAmount">₹0.00</span>
```

```
            </div>
            <div class="payment-row">
                <span>UPI ID:</span>
                <span id="paymentUpiId">none</span>
            </div>
        </div>

        <div class="payment-mode-switch">
            <button onclick="showQRPayment()">Pay by Scanning QR</button>
            <button onclick="showPaymentScanner()">Scan Payment
Confirmation</button>
        </div>

        <div id="paymentQRCode"></div>

        <div id="paymentScanner" style="display: none;">
            <h3>Scan Payment Confirmation</h3>
            <div id="paymentScannerViewport" class="viewport">
                <button class="camera-toggle"
onclick="togglePaymentCamera()">Switch Camera</button>
            </div>
        </div>

        <div class="payment-status" id="paymentStatus">
            Payment Pending
        </div>

        <div class="payment-instructions">
            1. Scan the QR code with your UPI app
            2. Complete the payment
            3. Show the payment confirmation QR to verify
        </div>
    </div>
  </div>

  <script>
    let cartItems = [];
    let totalWeight = 0;
    let expectedWeight = 0;
    let currentCamera = 'environment';
    let cartTotal = 0;

    async function startShopping() {
        const upiId = document.getElementById('upiId').value;
        if (!upiId) {
            alert('Please enter UPI ID');
            return;
        }

        document.getElementById('loginSection').style.display = 'none';
        document.getElementById('shoppingSection').style.display = 'block';
```

```
        try {
            await initBarcodeScanner();
            initWeightSensor();
        } catch (error) {
            console.error('Failed to initialize scanner:', error);
            alert('Failed to access camera. Please ensure camera permissions are
granted.');
        }
    }

    async function initBarcodeScanner() {
        try {
            const stream = await navigator.mediaDevices.getUserMedia({ video: true });
            stream.getTracks().forEach(track => track.stop());
        } catch (error) {
            throw new Error('Camera access denied or not available');
        }

        return new Promise((resolve, reject) => {
            Quagga.init({
                inputStream: {
                    name: "Live",
                    type: "LiveStream",
                    target: document.querySelector("#interactive"),
                    constraints: {
                        facingMode: currentCamera
                    },
                },
                decoder: {
                    readers: [
                        "ean_reader",
                        "ean_8_reader",
                        "upc_reader",
                        "upc_e_reader"
                    ],
                    debug: {
                        showCanvas: true,
                        showPatches: true,
                        showFoundPatches: true,
                        showSkeleton: true,
                        showLabels: true,
                        showPatchLabels: true,
                        showRemainingPatchLabels: true,
                    }
                }
            }, function(err) {
                if (err) {
                    reject(err);
                    return;
                }
```

```
            Quagga.start();
            resolve();
          });

          Quagga.onDetected(handleBarcodeDetection);
        });
      }

      async function toggleCamera() {
        await Quagga.stop();
        currentCamera = currentCamera === 'environment' ? 'user' : 'environment';
        await initBarcodeScanner();
      }

      function handleBarcodeDetection(result) {
        const code = result.codeResult.code;
        if (isValidBarcode(code)) {
          handleBarcode(code);
          const viewport = document.querySelector("#interactive");
          viewport.style.outline = "5px solid #4CAF50";
          setTimeout(() => {
            viewport.style.outline = "none";
          }, 500);
        }
      }

      function isValidBarcode(code) {
        return /^\d{8,13}$/.test(code);
      }

      function initWeightSensor() {
        window.weightSensor = setInterval(() => {
          totalWeight = expectedWeight + (Math.random() * 0.1 - 0.05);
          updateWeightDisplay();
        }, 1000);
      }

      function handleBarcode(barcode) {
        const product = lookupProduct(barcode);
        if (product) {
          if (!cartItems.some(item => item.barcode === barcode)) {
            expectedWeight += product.weight;
            cartItems.push({...product, barcode});
            updateCartDisplay();
            validateWeight();

            const status = document.getElementById('cartStatus');
            status.className = 'cart-status status-green';
            status.textContent = `Added: ${product.name}`;
            setTimeout(() => validateWeight(), 2000);
          }
```

```
      } else {
        const status = document.getElementById('cartStatus');
        status.className = 'cart-status status-yellow';
        status.textContent = 'Unknown product';
        setTimeout(() => validateWeight(), 2000);
      }
    }

    function lookupProduct(barcode) {
    const products = {
      "8904004400731": { name: "Balaji Aloo Bhujiya", price: 50, weight: 0.22 },
      "8904073614018": { name: "Amul Milk 1L", price: 70, weight: 1.0 },
      "8906002852687": { name: "Britannia Bread", price: 45, weight: 0.4 },
      "8904073614025": { name: "Amul Butter 500g", price: 285, weight: 0.5 },
      "8906002852694": { name: "Good Day Biscuits", price: 30, weight: 0.25 },
      "8904073614032": { name: "Amul Cheese 200g", price: 135, weight: 0.2 },
      "8906002852700": { name: "Tata Salt 1kg", price: 25, weight: 1.0 },
      "8904073614049": { name: "MTR Rava 500g", price: 55, weight: 0.5 }
    };
    return products[barcode];
  }

    function updateWeightDisplay() {
      document.getElementById('currentWeight').textContent =
        `${totalWeight.toFixed(2)} kg`;
      validateWeight();
    }

    function validateWeight() {
      const tolerance = 0.1;
      const cartStatus = document.getElementById('cartStatus');

      if (Math.abs(totalWeight - expectedWeight) <= tolerance) {
        cartStatus.className = 'cart-status status-green';
        cartStatus.textContent = 'Cart Status: Valid';
        return true;
      } else {
        cartStatus.className = 'cart-status status-red';
        cartStatus.textContent = 'Cart Status: Weight Mismatch';
        return false;
      }
    }

    function updateCartDisplay() {
      const list = document.getElementById('itemsList');
      const total = document.getElementById('totalPrice');

      list.innerHTML = cartItems.map(item =>
        `<li>
          <span>${item.name}</span>
          <span>$${item.price.toFixed(2)}</span>
```

```
        </li>`
    ).join('');

    cartTotal = cartItems.reduce((sum, item) => sum + item.price, 0);
    total.textContent = `Total: ₹${cartTotal.toFixed(2)}`;
}

function generatePaymentQR(amount, upiId) {
    const qr = qrcode(0, 'M');
    const paymentData = {
        pa: upiId,
        pn: "Smart Cart",
        am: amount.toString(),
        cu: "INR",
        tn: `Cart payment ${Date.now()}`
    };

    const upiUrl = `upi://pay?${Object.entries(paymentData)
        .map(([k, v]) => `${k}=${encodeURIComponent(v)}`)
        .join('&')}`;

    qr.addData(upiUrl);
    qr.make();

    document.getElementById('paymentQRCode').innerHTML =
qr.createImgTag(5);
}

let paymentScanner = null;

function showQRPayment() {
    if (paymentScanner) {
        Quagga.stop();
        paymentScanner = null;
    }
    document.getElementById('paymentScanner').style.display = 'none';
    document.getElementById('paymentQRCode').style.display = 'block';
}

async function showPaymentScanner() {
    document.getElementById('paymentQRCode').style.display = 'none';
    document.getElementById('paymentScanner').style.display = 'block';

    if (!paymentScanner) {
        await initPaymentScanner();
    }
}

async function initPaymentScanner() {
    try {
```

```
            return new Promise((resolve, reject) => {
                Quagga.init({
                    inputStream: {
                        name: "Live",
                        type: "LiveStream",
                        target: document.querySelector("#paymentScannerViewport"),
                        constraints: {
                            facingMode: currentCamera
                        },
                    },
                    decoder: {
                        readers: ["qrcode_reader"],
                        debug: {
                            showCanvas: true,
                            showPatches: true,
                            showFoundPatches: true,
                            showSkeleton: true,
                            showLabels: true,
                            showPatchLabels: true,
                            showRemainingPatchLabels: true,
                        }
                    }
                }, function(err) {
                    if (err) {
                        reject(err);
                        return;
                    }
                    Quagga.start();
                    paymentScanner = true;
                    resolve();
                });

                Quagga.onDetected(handlePaymentQRDetection);
            });
        } catch (error) {
            console.error('Failed to initialize payment scanner:', error);
            alert('Failed to access camera for payment scanning. Please ensure camera
permissions are granted.');
        }
    }

    async function togglePaymentCamera() {
        if (paymentScanner) {
            await Quagga.stop();
            currentCamera = currentCamera === 'environment' ? 'user' : 'environment';
            await initPaymentScanner();
        }
    }

    function handlePaymentQRDetection(result) {
        const qrData = result.codeResult.code;
```

```
        try {
            const paymentInfo = parsePaymentQR(qrData);
            if (validatePayment(paymentInfo)) {
                completeTransaction();
            } else {
                updatePaymentStatus('Invalid payment QR code', 'failed');
            }
        } catch (error) {
            console.error('Invalid QR code:', error);
            updatePaymentStatus('Invalid QR code format', 'failed');
        }
    }

    function parsePaymentQR(qrData) {
        // Parse UPI payment confirmation QR code
        // This is a simplified version - in real implementation,
        // you would verify digital signatures and check with payment gateway
        try {
            return JSON.parse(qrData);
        } catch {
            throw new Error('Invalid QR format');
        }
    }

    function validatePayment(paymentInfo) {
        // Simplified payment validation
        // In real implementation, this would verify with payment gateway
        return paymentInfo &&
               paymentInfo.amount === cartTotal.toFixed(2) &&
               paymentInfo.status === 'SUCCESS';
    }

    function updatePaymentStatus(message, status) {
        const statusElement = document.getElementById('paymentStatus');
        statusElement.textContent = message;
        statusElement.className = `payment-status status-${status}`;
    }

    function completeTransaction() {
        updatePaymentStatus('Payment Completed Successfully', 'completed');
        // Stop all scanners
        if (paymentScanner) {
            Quagga.stop();
        }

        // Clear cart
        cartItems = [];
        expectedWeight = 0;
        totalWeight = 0;
        updateCartDisplay();
        updateWeightDisplay();
```

```
            // Show completion message
            setTimeout(() => {
                alert('Thank you for shopping! You can collect your receipt.');
                resetApplication();
            }, 2000);
        }

        function resetApplication() {
            // Reset to initial state
            document.getElementById('shoppingSection').style.display = 'none';
            document.getElementById('paymentSection').style.display = 'none';
            document.getElementById('loginSection').style.display = 'block';
            document.getElementById('upiId').value = '';

            // Clear any remaining status
            const cartStatus = document.getElementById('cartStatus');
            cartStatus.className = 'cart-status';
            cartStatus.textContent = 'Cart Status: Ready';
        }

        function proceedToCheckout() {
            if (!validateWeight()) {
                alert('Please ensure cart weight matches before checkout');
                return;
            }

            if (cartItems.length === 0) {
                alert('Cart is empty');
                return;
            }

            document.getElementById('shoppingSection').style.display = 'none';
            document.getElementById('paymentSection').style.display = 'block';

            // Update payment details
            document.getElementById('paymentTotalItems').textContent =
cartItems.length;
            document.getElementById('paymentAmount').textContent =
`₹${cartTotal.toFixed(2)}`;
            document.getElementById('paymentUpiId').textContent =
document.getElementById('upiId').value;

            // Generate payment QR code
            generatePaymentQR(cartTotal, document.getElementById('upiId').value);
            updatePaymentStatus('Awaiting Payment', 'pending');
        }
    </script>
</body>
</html>
```

3) TAV-Cart Sales Analysis and Prediction

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta

# Generate sample TAV-Cart transaction data
np.random.seed(42)

# Generate dates for one year
dates = pd.date_range(start='2023-01-01', end='2023-12-31', freq='D')

# Create sample data
data = {
    'date': dates,
    'day_of_week': dates.dayofweek,
    'is_weekend': dates.dayofweek >= 5,
    'is_holiday': np.random.choice([0, 1], size=len(dates), p=[0.95, 0.05]),
    'store_location': np.random.choice(['North', 'South', 'East', 'West'], size=len(dates)),
    'active_carts': np.random.randint(20, 50, size=len(dates)),
    'avg_cart_value': np.random.normal(2500, 500, size=len(dates)),
    'items_per_cart': np.random.normal(15, 3, size=len(dates))
}

# Create DataFrame
df = pd.DataFrame(data)

# Add some seasonal effects
df['seasonal_factor'] = np.sin(2 * np.pi * df.index / len(df)) + 1
df['avg_cart_value'] = df['avg_cart_value'] * df['seasonal_factor']

# Calculate daily sales
df['daily_sales'] = df['active_carts'] * df['avg_cart_value']

# Data Analysis and Visualization Functions
def plot_sales_trends():
    plt.figure(figsize=(15, 10))

    # Daily sales trend
    plt.subplot(2, 1, 1)
    plt.plot(df['date'], df['daily_sales'], label='Daily Sales')
    plt.title('Daily Sales Trend')
    plt.xlabel('Date')
```

```
    plt.ylabel('Sales (₹)')
    plt.legend()

    # Weekly average sales
    weekly_sales = df.groupby(df['date'].dt.isocalendar().week)['daily_sales'].mean()
    plt.subplot(2, 1, 2)
    plt.plot(weekly_sales.index, weekly_sales.values, label='Weekly Average Sales')
    plt.title('Weekly Average Sales')
    plt.xlabel('Week Number')
    plt.ylabel('Average Sales (₹)')
    plt.legend()

    plt.tight_layout()
    plt.show()

def analyze_cart_metrics():
    plt.figure(figsize=(15, 5))

    # Cart utilization by day of week
    plt.subplot(1, 2, 1)
    sns.boxplot(x='day_of_week', y='active_carts', data=df)
    plt.title('Cart Utilization by Day of Week')
    plt.xlabel('Day of Week (0=Monday)')
    plt.ylabel('Number of Active Carts')

    # Average cart value by location
    plt.subplot(1, 2, 2)
    sns.barplot(x='store_location', y='avg_cart_value', data=df)
    plt.title('Average Cart Value by Store Location')
    plt.xlabel('Store Location')
    plt.ylabel('Average Cart Value (₹)')

    plt.tight_layout()
    plt.show()

# Sales Prediction Model
def train_sales_prediction_model():
    # Prepare features
    features = ['day_of_week', 'is_weekend', 'is_holiday', 'active_carts',
               'items_per_cart', 'seasonal_factor']
    X = pd.get_dummies(df[features], columns=['day_of_week'])
    y = df['daily_sales']

    # Split data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    # Train model
    model = LinearRegression()
    model.fit(X_train, y_train)
```

```
    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate metrics
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"Model Performance:")
    print(f"Mean Squared Error: {mse:.2f}")
    print(f"R² Score: {r2:.2f}")

    return model, X.columns

def predict_future_sales(model, feature_names, days=30):
    # Generate future dates
    last_date = df['date'].max()
    future_dates = pd.date_range(start=last_date + timedelta(days=1),
                                periods=days, freq='D')

    # Create future features
    future_data = pd.DataFrame({
        'day_of_week': future_dates.dayofweek,
        'is_weekend': future_dates.dayofweek >= 5,
        'is_holiday': np.random.choice([0, 1], size=len(future_dates), p=[0.95, 0.05]),
        'active_carts': np.random.randint(20, 50, size=len(future_dates)),
        'items_per_cart': np.random.normal(15, 3, size=len(future_dates)),
        'seasonal_factor': np.sin(2 * np.pi * np.arange(len(future_dates)) / 365) + 1
    })

    # Prepare features
    future_X = pd.get_dummies(future_data, columns=['day_of_week'])
    future_X = future_X.reindex(columns=feature_names, fill_value=0)

    # Make predictions
    predicted_sales = model.predict(future_X)

    # Plot predictions
    plt.figure(figsize=(12, 6))
    plt.plot(future_dates, predicted_sales, label='Predicted Sales')
    plt.title('Predicted Sales for Next 30 Days')
    plt.xlabel('Date')
    plt.ylabel('Predicted Sales (₹)')
    plt.legend()
    plt.show()

    return pd.DataFrame({'date': future_dates, 'predicted_sales': predicted_sales})

# Run analysis
if __name__ == "__main__":
```

```
  print("Analyzing TAV-Cart Sales Data...")
  plot_sales_trends()
  analyze_cart_metrics()

  print("\nTraining Sales Prediction Model...")
  model, feature_names = train_sales_prediction_model()

  print("\nPredicting Future Sales...")
  future_sales = predict_future_sales(model, feature_names)
```

4) TAV-Cart Sales Dashboard:

```
import React from 'react';
import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
import { LineChart, Line, BarChart, Bar, XAxis, YAxis, CartesianGrid, Tooltip,
Legend, ResponsiveContainer } from 'recharts';

const generateSampleData = () => {
 // Generate 30 days of sample data
 const data = [];
 const date = new Date('2024-01-01');

 for (let i = 0; i < 30; i++) {
  const weekday = date.toLocaleDateString('en-US', { weekday: 'short' });
  data.push({
   date: date.toLocaleDateString(),
   weekday,
   sales: Math.random() * 100000 + 50000,
   activeCards: Math.floor(Math.random() * 30 + 20),
   avgCartValue: Math.random() * 3000 + 2000
  });
  date.setDate(date.getDate() + 1);
 }

 return data;
};

const aggregateByWeekday = (data) => {
 const weekdayData = {};
 data.forEach(item => {
  if (!weekdayData[item.weekday]) {
   weekdayData[item.weekday] = {
    weekday: item.weekday,
    avgSales: 0,
    count: 0
   };
  }
  weekdayData[item.weekday].avgSales += item.sales;
  weekdayData[item.weekday].count += 1;
 });
```

```
 return Object.values(weekdayData).map(item => ({
   weekday: item.weekday,
   avgSales: item.avgSales / item.count
 }));
};

const TAVSalesDashboard = () => {
 const salesData = generateSampleData();
 const weekdayData = aggregateByWeekday(salesData);

 return (
   <div className="w-full max-w-6xl mx-auto p-4 space-y-6">
     <h1 className="text-2xl font-bold mb-6">TAV-Cart Sales Dashboard</h1>

     <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
       <Card>
         <CardHeader>
           <CardTitle>Daily Sales Trend</CardTitle>
         </CardHeader>
         <CardContent className="h-72">
           <ResponsiveContainer width="100%" height="100%">
             <LineChart data={salesData}>
               <CartesianGrid strokeDasharray="3 3" />
               <XAxis dataKey="date" />
               <YAxis />
               <Tooltip />
               <Legend />
               <Line type="monotone" dataKey="sales" stroke="#8884d8" name="Sales
(₹)" />
             </LineChart>
           </ResponsiveContainer>
         </CardContent>
       </Card>

       <Card>
         <CardHeader>
           <CardTitle>Average Sales by Weekday</CardTitle>
         </CardHeader>
         <CardContent className="h-72">
           <ResponsiveContainer width="100%" height="100%">
             <BarChart data={weekdayData}>
               <CartesianGrid strokeDasharray="3 3" />
               <XAxis dataKey="weekday" />
               <YAxis />
               <Tooltip />
               <Legend />
               <Bar dataKey="avgSales" fill="#82ca9d" name="Avg Sales (₹)" />
             </BarChart>
           </ResponsiveContainer>
         </CardContent>
```

```
        </Card>

        <Card>
          <CardHeader>
            <CardTitle>Active Carts Usage</CardTitle>
          </CardHeader>
          <CardContent className="h-72">
            <ResponsiveContainer width="100%" height="100%">
              <LineChart data={salesData}>
                <CartesianGrid strokeDasharray="3 3" />
                <XAxis dataKey="date" />
                <YAxis />
                <Tooltip />
                <Legend />
                <Line type="monotone" dataKey="activeCards" stroke="#ffc658"
name="Active Carts" />
              </LineChart>
            </ResponsiveContainer>
          </CardContent>
        </Card>

        <Card>
          <CardHeader>
            <CardTitle>Average Cart Value Trend</CardTitle>
          </CardHeader>
          <CardContent className="h-72">
            <ResponsiveContainer width="100%" height="100%">
              <LineChart data={salesData}>
                <CartesianGrid strokeDasharray="3 3" />
                <XAxis dataKey="date" />
                <YAxis />
                <Tooltip />
                <Legend />
                <Line type="monotone" dataKey="avgCartValue" stroke="#ff7300"
name="Avg Cart Value (₹)" />
              </LineChart>
            </ResponsiveContainer>
          </CardContent>
        </Card>
      </div>
    </div>
  );
};

export default TAVSalesDashboard;
```

Appendix B: Data Sheets

Data Generation Methodology

The dataset used in this visualization is fictional and was generated programmatically to simulate TAV-Cart sales patterns. The data generation process was designed to mimic realistic retail cart usage patterns while maintaining data privacy and providing meaningful insights for analysis purposes.

Dataset Structure and Generation Parameters

The dataset spans one year of daily transactions from January 1, 2023 to December 31, 2023. Each record contains various metrics that were generated using controlled random distributions to ensure realistic patterns while maintaining variability. The dataset includes daily records with the following fields:

For each day, the dataset includes the date, day of the week (0-6, where 0 is Monday), a boolean flag for weekends, and a holiday indicator. Store locations were randomly assigned across four zones (North, South, East, West). The number of active carts was generated using random integers between 20 and 50 to simulate varying cart utilization levels. Average cart values were generated using a normal distribution with a mean of ₹2,500 and a standard deviation of ₹500. The items per cart metric follows a normal distribution with a mean of 15 items and a standard deviation of 3 items.

Seasonal Adjustments and Sales Calculations

To incorporate realistic seasonal patterns, a seasonal factor was applied using a sinusoidal function that creates yearly cycles. This seasonal factor was multiplied with the average cart values to simulate seasonal variations in purchasing patterns. Daily sales were calculated by multiplying the number of active carts with the average cart value for each day.

Sample Data Excerpt (For One Month (31days))

```
date,day_of_week,is_weekend,is_holiday,store_location,active_carts,avg_cart_value,items_per_cart,seasonal_factor,daily_sales
2023-01-01,6,True,0,South,32,2487.65,14.8,1.12,89555.40
2023-01-02,0,False,1,North,45,2534.21,15.2,1.15,130982.85
2023-01-03,1,False,0,East,38,2456.78,14.6,1.18,110555.10
```

```
2023-01-04,2,False,0,West,41,2567.34,15.5,1.21,127284.37
2023-01-05,3,False,0,North,36,2498.92,14.9,1.24,111451.58
2023-01-06,4,False,0,South,42,2545.67,15.3,1.27,136278.85
2023-01-07,5,True,0,East,47,2478.90,14.7,1.30,151513.97
2023-01-08,6,True,0,West,33,2512.45,15.1,1.33,110547.80
2023-01-09,0,False,0,North,39,2489.76,14.8,1.36,131957.27
2023-01-10,1,False,0,South,44,2523.12,15.4,1.39,154169.75
2023-01-11,2,False,0,East,37,2478.34,14.9,1.42,130456.89
2023-01-12,3,False,1,West,43,2556.78,15.6,1.45,159654.23
2023-01-13,4,False,0,North,35,2489.90,14.7,1.48,128945.67
2023-01-14,5,True,0,South,46,2534.56,15.3,1.51,176234.56
2023-01-15,6,True,0,East,40,2467.89,14.8,1.54,151789.34
2023-01-16,0,False,0,West,38,2512.34,15.0,1.57,149876.54
2023-01-17,1,False,0,North,42,2545.67,15.2,1.60,171234.56
2023-01-18,2,False,0,South,36,2478.90,14.6,1.63,145678.90
2023-01-19,3,False,0,East,45,2523.45,15.4,1.66,188567.89
2023-01-20,4,False,1,West,39,2489.78,14.9,1.69,164523.45
2023-01-21,5,True,0,North,48,2567.89,15.7,1.72,211789.34
2023-01-22,6,True,0,South,34,2456.78,14.5,1.75,146234.56
2023-01-23,0,False,0,East,41,2534.56,15.3,1.78,185678.90
2023-01-24,1,False,0,West,37,2478.90,14.8,1.81,166234.56
2023-01-25,2,False,0,North,44,2512.34,15.2,1.84,203456.78
2023-01-26,3,False,1,South,38,2545.67,15.5,1.87,181234.56
2023-01-27,4,False,0,East,43,2489.90,14.9,1.90,203567.89
2023-01-28,5,True,0,West,49,2556.78,15.8,1.93,241789.34
2023-01-29,6,True,0,North,35,2467.89,14.6,1.96,169876.54
2023-01-30,0,False,0,South,42,2523.45,15.4,1.99,211234.56
2023-01-31,1,False,0,East,40,2498.90,15.1,2.02,201789.34
```

Data Generation Considerations

The data generation process incorporated several key considerations to ensure the dataset would be useful for analysis while remaining fictional:

The seasonal patterns were designed to reflect typical retail behavior, with higher sales during festival seasons and weekends. The holiday flag was set with a 5% probability to simulate occasional special events or holidays. Store location distribution was kept uniform across all four zones to allow for meaningful comparative analysis. Cart utilization patterns were designed to show higher numbers during weekends and holidays, reflecting typical shopping behavior. The items per cart metric was generated to maintain a realistic shopping basket size while incorporating natural variability.

Usage Notes

This dataset is entirely fictional and was generated for demonstration purposes only. While the patterns and relationships in the data are designed to be realistic, they should not be used as a basis for actual business decisions. The data generation process prioritized creating meaningful patterns for visualization and analysis while maintaining plausible relationships between variables. Any similarities to actual sales patterns or business metrics are purely coincidental.

Appendix C: List of Components

Development Environment

- IDE: Visual Studio Code (VS Code):

  VS Code is a lightweight, open-source code editor widely used for JavaScript, React, and Node.js development. It supports extensive extensions, such as ESLint and Prettier, enabling rapid iteration, debugging, and error checking. Live Server and REST Client extensions help test APIs directly from the IDE, improving productivity and development speed.
- Version Control: Git/GitHub:

  Git is a version control system used to track changes and facilitate collaboration. It allows developers to work on different branches simultaneously, manage conflicts, and roll back to previous versions if necessary. GitHub serves as a central repository for the project, providing tools for collaboration, code reviews, and CI/CD integration.

Deployment:

- Frontend: Netlify

  Netlify simplifies the deployment of React-based user interfaces by offering continuous integration and automated builds from the GitHub repository. It provides fast, reliable hosting with built-in support for environment variables and form handling.
- Backend: Vercel

  Vercel is used to deploy the backend services developed with Node.js and Express. It ensures automatic scaling, handles API requests efficiently, and supports serverless functions for seamless backend integration.

Programming Language & Frameworks

- Programming Language: JavaScript (ES6+):

  JavaScript with ES6+ syntax introduces features such as arrow functions, template literals, and destructuring, enhancing code readability and maintainability. It is used across both the frontend (React) and backend (Node.js) of the TAV-Cart system.

- Frontend Framework: React.js:

  React is a popular JavaScript library for building dynamic, interactive UIs. It offers reusable components, state management with hooks, and efficient rendering using a virtual DOM. This ensures a smooth user experience with instant feedback on product scans and cart operations.
- Backend Framework: Express.js:

  Express is a lightweight and flexible framework built on top of Node.js, used for building RESTful APIs. It simplifies the creation of backend logic for handling UPI transactions, cart validation, and communication with the cloud database.

Styling

- Tailwind CSS:

  Tailwind CSS is a utility-first framework that offers pre-defined classes for fast styling. It enables the creation of responsive layouts with minimal CSS code, allowing developers to focus more on functionality.
- Material-UI:

  Material-UI provides a collection of ready-to-use React components, such as buttons, modals, and tables. These components adhere to modern web design principles and enhance the visual consistency of the TAV-Cart interface.

Database

- MongoDB (with Mongoose):

  MongoDB is a NoSQL database ideal for storing semi-structured data such as cart contents, user information, and product inventories. Mongoose, an Object Data Modeling (ODM) library, provides schema-based modeling, simplifying data validation and interaction with the MongoDB database.

Libraries and Tools

- Nodemailer:

  A Node.js library that automates the sending of emails for transaction receipts, payment confirmations, and cart-related updates. It ensures timely communication with customers, improving the overall shopping experience.

- Cloudinary:

  A cloud-based service used to store and manage media files, such as product images. Cloudinary ensures that all images displayed in the cart interface load quickly, contributing to a smooth user experience.
- JWT (JSON Web Token):

  JWTs are used to secure user authentication and session handling. This ensures that only authorized users can access certain features, such as payment processing and shopping history.
- Axios:

  Axios is a promise-based HTTP client for making API requests. It streamlines communication between the React frontend and Express backend, handling both GET and POST requests efficiently.
- React Hook Form:

  React Hook Form simplifies form management and validation in React applications. It ensures that customer inputs, such as UPI credentials or phone numbers, are accurate and properly formatted before processing.
- Draw.io:

  Draw.io is a diagramming tool used to create flowcharts, UML diagrams, and system architecture representations. These visual tools aid in documenting the TAV-Cart's design and workflow.

Data Manipulation & Visualization

- Chart.js or Recharts:

  These JavaScript libraries are used for visualizing data, such as sales trends and shopping patterns. They generate dynamic charts and graphs, helping retailers analyze customer behavior and optimize product placements.

Preprocessing & Data Validation

- React Hook Form:

  This tool validates user inputs in real-time, ensuring that data like phone numbers and UPI IDs are correctly formatted. It eliminates the need for external validation libraries like Yup, improving performance.

User Interface Components

- React:

  React components enhance the user interface by providing consistent, responsive designs. Pre-built modals, tables, and buttons make it easy to develop features like product displays and checkout pages.

Authentication & Security

- Bcrypt.js:

  Bcrypt.js ensures secure storage of user passwords and sensitive data by hashing them before storing them in the database. This reduces the risk of unauthorized access.

Appendix D: List of Paper Presented and Published

Akshit Naithani, Tanish Singh Rajpal, Vansh Mistry, "TAV-Cart: Transaction Assistance and Validation Cart System for Effortless Shopping and Checkout", Institute of Electrical and Electronics Engineers(IEEE) – 2024, Accepted.