QKart - Checkout Simplified

A midterm capstone project report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electronics and Computers Engineering / Electronics and Communication Engineering

Submitted By

Abhimanyu Kumar (Regn. No: 102215021)

Arushi Gupta (Regn. No: 102206190)

Khushi Kaushal (Regn. No: 102206269)

Tanishka Bhatia (Regn. No: 102206217)

Uttkarsh Singh Pathania (Regn. No: 102215291)

(CPG- 36)

Under Supervision of

Dr. Geetika Dua (Assistant Professor-II, ECED)

Department of Electronics and Communication Engineering

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, PATIALA, PUNJAB

August 2025

DECLARATION

We hereby declare that the capstone project group report title “QKart - Checkout Simplified“ is an authentic record of our own work carried out at “Thapar Institute of Engineering and Technology, Patiala” as a Capstone Project in seventh semester of B.E. (Electronics & Communication Engineering), under the guidance of “**Dr. Geetika Dua**”, during January to July, 2025. We confirm that all information and concepts presented herein are accurate and a true reflection of our proposed project. We have cited all external sources and references appropriately.

Date: 11-08-2025

Abhimanyu Kumar	102215021	Signature
Arushi Gupta	102206190	Signature
Khushi Kaushal	102206269	Signature
Tanishka Bhatia	102206217	Signature
Uttkarsh Singh Pathania	102215291	Signature

ACKNOWLEDGEMENT

We wish to express our profound gratitude to all those who have supported us throughout this capstone project. We are particularly indebted to our esteemed mentor, Dr. Geetika Dua, Assistant Professor at the Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala. Her guidance, insightful feedback, and constant encouragement have been invaluable in shaping our project from a proposal into a tangible reality.

Our sincere thanks also go to the Department of Electronics and Communication Engineering and the management of Thapar Institute of Engineering and Technology for providing us with the necessary resources, a conducive learning environment, and the opportunity to undertake a project of this scale. The knowledge gained from relevant course subjects, including Computer Vision & Image Processing, Embedded Systems, and Machine Learning & AI, was foundational to our work.

Finally, we would like to acknowledge the hard work, dedication, and collaborative spirit of our team members— Abhimanyu Kumar, Uttkarsh Singh Pathania, Tanishka Bhatia, Arushi Gupta, and Khushi Kaushal. This project, "QKart: Checkout Simplified," is a testament to our collective efforts and shared vision. We are proud to have worked together to create a solution that addresses real-world challenges in the retail industry.

Date: 11-08-25

Roll No.	**Name**
102215021	Abhimanyu Kumar
102206190	Arushi Gupta
102206269	Khushi Kaushal
102215291	Uttkarsh Singh Pathania
102206217	Tanishka Bhatia

ABSTRACT

This capstone project proposal introduces **QKart**, an intelligent shopping cart meticulously engineered to redefine the retail checkout landscape. Traditional retail environments are often hampered by systemic inefficiencies, including protracted checkout queues, manual scanning inaccuracies, and inadequate inventory management, all of which contribute to customer dissatisfaction and operational bottlenecks. QKart presents a sophisticated solution to these pervasive challenges by seamlessly integrating cutting-edge computer vision (CV) and Internet of Things (IoT) technologies to enable automated item detection, real-time billing, and dynamic inventory synchronization.

The core of the QKart prototype is the **Raspberry Pi 5 (4GB RAM)**, which serves as the primary processing unit. This is complemented by a highly optimized, lightweight **YOLOv4-tiny model**, specifically chosen for its efficacy on edge devices to perform real-time object detection as items are placed into or removed from the cart. To ensure unparalleled accuracy and mitigate potential errors, the system incorporates weight sensors equipped with **HX711 modules** to provide a crucial layer of cross-verification for detected items. All transactional and inventory data are meticulously synchronized in real-time via a **Firebase cloud database** to a dedicated web application. This empowers customers with a live view of their cart total while simultaneously providing store management with a seamless and accurate inventory management system.

The project's innovative approach lies in its dual-validation mechanism, which fuses computer vision with weight-based data. This integration is designed to overcome the limitations of a single-sensor system, ensuring a higher degree of reliability and security. By actively cross-referencing visual data with precise weight measurements, QKart can accurately identify items even under challenging conditions, such as occlusions or varying lighting. This redundancy not only improves the system's accuracy but also serves as a robust anti-theft measure by flagging discrepancies between the detected items and their combined weight. The focus on this fused data approach is central to the project's objective of delivering a dependable and trustworthy solution for retailers.

By completely eliminating the need for manual item scanning, QKart is poised to significantly reduce checkout times, enhance overall operational efficiency, and elevate the customer experience. Furthermore, it offers a distinct advantage over existing, more costly solutions like RFID-based carts, which necessitate expensive product tagging. The project's technical objectives include ensuring robust item detection under varying conditions, seamlessly integrating CV with weight-based validation, and maintaining low-latency data synchronization. The ultimate goal is to deliver a robust, cost-effective, and scalable system that democratizes smart cart technology, making it accessible to a broader range of retailers and providing a competitive edge in a modern, fast-paced market.

TABLE OF CONTENTS

DECLARATION
ACKNOWLEDGEMENT
ABSTRACT
TABLE OF CONTENT
LIST OF FIGURES
CHAPTER 1- INTRODUCTION
1.1 PROJECT OVERVIEW 1
1.2 MOTIVATION 2
1.3 ASSUMPTIONS AND CONSTRAINTS 3
1.4 NOVELTY OF WORK 4
CHAPTER 2- LITERATURE SURVEY 5
2.1 LITERATURE SURVEY 5
2.2 RESEARCH GAPS 8
2.3 PROBLEM DEFINITION AND SCOPE 9
CHAPTER 3- DIAGRAMATIC ANALYSIS 10
3.1 SYSTEM ARCHITECTURE 10
3.2 MODEL ANALYSIS 11
3.3 TOOLS AND TECHNOLOGY USED 11
CHAPTER 4- PROJECT DESIGN AND DESCRIPTION 12
4.1 DESCRIPTION 12
4.2 U.G. SUBJECTS 13
4.3 STANDARDS USED 13
CHAPTER 5- IMPLEMENTATION AND EXPERIMENTAL RESULTS 14
5.1 SIMULATION RESULTS 14
5.2 HARDWARE PROGRESS 16
CHAPTER 6- OUTCOME AND PROSPECTIVE LEARNING 17
6.1 SCOPE AND OUTCOMES 17
6.2 PROSPECTIVE LEARNINGS 18
6.3 CONCLUSION 18
CHAPTER 7- PROJECT TIMELINE
7.1 WORK BREAKDOWN AND GANTT CHART 19
7.2 PROJECT TIMELINE 19

REFERENCES **20**

LIST OF FIGURES

Figure No.	Caption	Page No.
Figure No. 3.1	Flow Chart for System Architecture	1
Figure No. 5.1	Model Precision Metrics	14
Figure No. 5.2	Device Performance Statistics	14
Figure No. 5.3	Model Training Graph	15
Figure No. 5.4	Prototype for QKart	16
Figure No. 5.5	Hardware Setup for QKart	16
Figure No. 7.1	Gantt Chart	19

1. INTRODUCTION

1.1 PROJECT OVERVIEW

QKart is an intelligent shopping cart designed to revolutionize the checkout experience in grocery and supermarket stores. Traditional checkout systems are inefficient, suffering from long queues, manual scanning errors, and inventory mismanagement. QKart addresses these challenges by integrating computer vision (CV) and the Internet of Things (IoT) to automate item detection, billing, and real-time inventory updates. This innovative approach aims to enhance operational efficiency, reduce labor costs, and improve customer satisfaction. The core of the QKart system is a Raspberry Pi 5 (5GB RAM) which has been selected and is now in use as the central processing unit. The system employs a lightweight YOLO model, which has been trained and optimized for edge devices, to detect items as they are placed into or removed from the cart. To ensure accuracy and reduce errors, weight sensors with HX711 modules have been integrated and are used to cross-verify the items detected by the camera. This synchronization allows customers to view their cart total instantly, providing full transparency over their spending. The Flask-based web UI and a QR-based UPI payment gateway have been successfully developed and integrated, completing the checkout flow. This work demonstrates the feasibility of creating a robust system that improves checkout speed, minimizes errors, and supports dynamic inventory management.

The seamless operation of the QKart system is due to its meticulously designed architecture. At the heart of the system is the Raspberry Pi 5, which acts as the central hub, processing data from multiple sensors. The computer vision component, powered by the optimized YOLO model, is responsible for real-time object detection. Furthermore, the IoT aspect of QKart extends beyond the cart itself. The real-time data synchronization with Firebase is a key feature, enabling a dynamic and interactive shopping experience. As items are added or removed, the total bill and an itemized list are updated instantly and displayed on the customer's smartphone or a dedicated screen on the cart. This transparency empowers customers by allowing them to monitor their budget as they shop, preventing any checkout surprises. The integrated Flask-based web UI and QR-based UPI payment gateway make the final step of payment quick and contactless, allowing customers to simply scan a QR code to complete the transaction in seconds.

1.2 MOTIVATION

The modern retail industry faces increasing pressure to innovate in response to evolving customer expectations for convenience and efficiency. Traditional grocery stores that rely on manual checkouts suffer from significant inefficiencies, including long queues, billing errors, and poor inventory management. These operational shortcomings have a direct impact on profitability and customer loyalty. A 2022 McKinsey report highlighted a critical issue: over 30% of shoppers abandon their purchases due to long checkout lines, a problem that results in billions of dollars in lost revenue annually. Furthermore, retailers struggle with real-time inventory tracking, leading to frequent stock discrepancies, which can cause both overstocking and stockouts. These issues collectively degrade the customer experience and increase labor costs, which can account for approximately 15% of grocery store expenses.

The work completed in this midterm report directly addresses these consumer and retailer pain points by providing a cost-effective and automated solution. For shoppers, the system we have developed eliminates the frustration of long queues by automatically detecting items as they are placed in the cart and updating a real-time bill via a web-app. This gives customers full transparency over their spending and provides information on available inventory. For retailers, QKart automates the billing process and synchronizes stock data to the cloud, reducing the need for manual labor and minimizing human errors. The system also enables real-time stock alerts, allowing for proactive restocking to prevent lost sales.

The market relevance of QKart is underscored by the rapid growth of the global smart cart market, which is projected to expand at a 21% CAGR from 2023 to 2030, driven by the increasing demand for automation. Unlike expensive and complex existing solutions, QKart is designed to be accessible. It leverages lightweight AI models (YOLO) and affordable hardware (Raspberry Pi 5), making it a viable option for budget-conscious retailers. This approach is particularly advantageous when compared to RFID-based systems, which are cost-prohibitive and require every item to be equipped with an expensive tag. Our developed model eliminates the need for these tags, which can reduce implementation costs by 60-70%.

In a post-pandemic world, where retailers must compete with the convenience offered by online giants, the completed QKart prototype provides a critical, future-ready solution. By democratizing smart cart technology, it empowers small-to-mid-sized retailers to modernize their operations without significant capital investment. The project's progress to date confirms its ultimate goal is to not only improve efficiency and reduce costs but also to enhance customer retention and provide a competitive edge in a fast-paced market.

1.3 ASSUMPTIONS AND CONSTRAINTS

The development of QKart has been based on a number of key assumptions and has been subject to specific constraints to ensure its feasibility and scope.

Assumptions

- Availability of Power: We assume a continuous and stable power source for the charging of the smart cart's internal battery.
- Network Connectivity: The project assumes reliable Wi-Fi or internet connectivity within the retail store to facilitate real-time data synchronization with Firebase.
- Hardware Availability: All necessary hardware components, including the Raspberry Pi 5, camera module, weight sensors, and HX711 modules, were procured successfully.
- Dataset for Training: A sufficiently large and diverse dataset of store products was created to effectively train the YOLO.
- Operating Conditions: We assume the cart will be operated in a controlled indoor environment with consistent lighting, though the system is designed to handle some variation.

Constraints

- Cost-Effectiveness: A primary constraint is the project's focus on remaining a low-cost solution, which we have adhered to by utilizing affordable hardware and open-source software.
- Processing Power: The project is constrained by the processing limitations of the Raspberry Pi 5. The YOLO model was chosen and successfully implemented to ensure real-time performance on this edge device.
- Model Accuracy: While the system aims for high accuracy, the performance of the computer vision model may be affected by varying lighting and occlusion conditions.
- Scalability: The current prototype is designed for a limited number of products. Scaling the system to a large-scale retail environment would require a larger, more comprehensive dataset and robust backend infrastructure.
- Physical Limitations: The design is constrained by the physical size and weight capacity of a standard shopping cart, which was modified to accommodate the system's components without compromising its core functionality.

1.4 NOVELTY OF WORK

The QKart project represents a significant and novel advancement in retail automation by addressing the key limitations of existing smart cart solutions and making sophisticated technology both accessible and cost-effective. The primary novelty of our work is rooted in the following core innovations:

- **Enabling a Seamless Self-Checkout Experience:** The system's holistic design is centered around providing a completely automated and hands-free self-checkout experience for the customer. By integrating computer vision for automated product identification and real-time billing via a web application, QKart eliminates the need for manual scanning and human intervention at the checkout counter. This not only dramatically reduces wait times and operational bottlenecks but also empowers retailers to significantly cut down on labor costs associated with traditional cashiers.

- **Simultaneous Multi-Item Detection:** Unlike conventional systems that require individual item scanning, the project employs a lightweight YOLO model to detect multiple items simultaneously within a single camera frame. This capability allows customers to place several items into the cart at once, accelerating the shopping process and enhancing overall efficiency. The model's optimization for edge devices ensures this multi-item detection is performed in real-time on the Raspberry Pi 5, providing a rapid and accurate response.

- **Integrated Multi-Sensor Verification for Fraud Prevention:** The QKart system incorporates a unique and robust fraud prevention mechanism by integrating a computer vision model with weight-based verification using HX711 modules. This dual-verification approach ensures high accuracy by cross-referencing the detected item with its expected weight, significantly enhancing the reliability of the billing process and preventing unbilled items from leaving the store.

- **Cost-Effective and Accessible Technology:** QKart provides a market-disrupting solution by leveraging affordable hardware, such as the Raspberry Pi 5, and open-source software. This contrasts sharply with expensive proprietary systems and RFID-based solutions, which require costly product tags. This cost-conscious design makes sophisticated smart cart technology accessible to small-to-mid-sized retailers, giving them a competitive edge against larger establishments without significant investment.

- **Intuitive User Interface and Seamless Payment Integration:** The project offers a refined and user-friendly experience through a Flask-based web UI or mobile application that provides real-time billing. The interface allows users to review and even remove items before checkout. A key feature is the seamless QR-based UPI payment system, which enables quick and secure transactions, further enhancing customer convenience and reducing the time spent at checkout.

2.1 LITERATURE REVIEW

The retail industry is undergoing a significant transformation driven by the adoption of smart technologies to enhance the shopping experience. Queues at traditional cash desks cause frustration and dissatisfaction, along with the consequent loss of sales. With advances in AI, computer vision, and IoT, retailers are now exploring automated systems to facilitate the checkout process. This research guided our approach to building QKart. Smart carts are positioning themselves as the new buzzword: they integrate real-time object recognition and automated billing with digital payments, thus allowing customers to avoid traditional checkout counters.

This literature survey discusses existing research and some technology developments concerning QKart, an economical computer vision-based smart cart prototype. QKart aims to utilize cheap hardware and open-source software in its quest for the most affordable alternative to traditional checkout methods to improve user experience and enhance retail operations. The system has a dedicated user interface, giving an instant display of the bill, letting the customer know what comprises his cart, and giving seamless payment access through a UPI gateway. While the project is currently focused on the core checkout experience, it has been designed with future scalability in mind to incorporate features such as inventory management.

2.1.1 Existing Technologies and Comparative Analysis

Our research into existing technologies informed the design choices we have now implemented in QKart. Smart shopping carts have been developed by different large retailers and research institutions. Some of the most salient implementations include:

- **Amazon Dash Cart:** This system uses cameras, weight sensors, and computer vision to detect items placed in the cart, so there's no need to scan barcodes anymore.
- **Caper AI Smart Cart:** Caper Cart, an intelligent shopping cart equipped with object recognition systems, weight sensors, and touch-enabled screens.
- **RFID-Based Systems:** Several retail stores have RFID-based smart checkout solutions, where products equipped with an RFID tag are automatically detected when placed into a cart.

While these solutions enhance automation and efficiency, they often come at high costs due to expensive hardware, RFID infrastructure, or cloud processing. Unlike RFID systems, where every item must be equipped with an RFID tag, QKart automatically identifies items using a camera and advanced image processing. This eliminates the need for time-consuming manual tagging. QKart's implementation, empowered by computer vision and edge AI on a Raspberry Pi, was chosen to be an affordable alternative that provides a real-time billing experience through an app, thereby enhancing the user experience.

2.1.2 Core Technological Components

The technical foundation of smart carts rests on the seamless integration of several core components. QKart's implementation, now at the midterm stage, has been built upon these principles, with research guiding the selection and integration of each module.

❖ **Computer Vision and Object Detection**

Computer vision is the core technology behind smart carts, which enables automated product recognition. The following techniques were considered and informed our implementation:

- **YOLO (You Only Look Once):** A high-speed object detection model widely used in AI applications. It delivers real-time inference and superior accuracy, making it an easy choice for Q-kart. Studies have shown that YOLO offers a great balance between speed and accuracy, making it a suitable choice for real-time product detection on embedded systems. This research guided our decision to train and deploy the lightweight YOLOv4-tiny model on the Raspberry Pi, as it provides the optimal performance on the limited processing power of the edge device.
- **Google's Teachable Machine:** An easy-to-use platform on which custom object detection models may be trained by making use of a small dataset. It is best suited for resource-limited projects. While a valuable tool for rapid prototyping, our project required a more robust, custom-trained model for the specific product set, leading to the selection of YOLOv4-tiny.

To further enhance accuracy and provide a crucial layer of fraud prevention, a weight sensor is used for an accurate quantitative analysis of the product to be purchased. Direct integration with the QKart app allows all recognized products to be updated live in billing systems for user convenience.

❖ Sensor Integration and Fraud Prevention

A significant challenge identified in our research was the prevention of fraudulent activities, effectively stopping customers from adding or removing unbilled items. Our implemented solution was based on the following methods:

- **Weight-based verification:** This compares the weight that one would expect an item to weigh under their stated units of measure with its actual weight measured using a load cell + HX711 module. If differences arise, the system is programmed to flag potential fraud. This method provides a reliable and low-cost solution for verifying a purchase, as the HX711 module is a low-cost analog-to-digital converter for weight sensors.
- **Computer vision-based tracking:** Some advanced carts use several cameras to continuously track items in the cart, which imposes increased processing. Our research showed that this approach, while effective, would exceed the processing capabilities of the Raspberry Pi and increase the overall system cost.
- **Motion detection sensors:** Motion sensors are integrated with the scanning system to detect if an item is taken away after it has been scanned. This provides an additional layer of security but was deemed redundant given the effectiveness of our combined CV and weight-based approach.

QKart's integrated fraud prevention mechanism, which combines computer vision-based item detection with weight-based verification, has been implemented to create a reliable and effective system without the high processing demands of multi-camera setups. This design choice strikes the right balance between security, cost, and performance, making it a practical and innovative solution.

❖ Software Architecture and User Interface

The software architecture of the QKart system is designed to provide a seamless and user-friendly experience.

- **Hybrid Data Management:** The project utilizes a local database on the Raspberry Pi for real-time, low-latency processing and a cloud-based system for remote functionality. This hybrid approach ensures that the system can operate even with intermittent internet connectivity, with local data being synchronized with the cloud once a connection is re-established.
- **Real-time Billing and UI:** A Flask-based API has been developed to facilitate backend interactions and billing processing, linking the smart cart to the mobile app. This provides customers with a real-time list of added items, the total bill, and available payment options, offering full transparency over their spending.
- **Seamless Payment Integration:** The system has been implemented with a QR-based UPI payment gateway to enable quick and secure transactions. This final step in the user journey eliminates the need for a traditional cash counter and further enhances the speed and convenience of the shopping experience.

This combination of efficient software architecture and an intuitive user interface addresses a key pain point for consumers and retailers alike, moving beyond the limitations of traditional checkout systems.

The literature survey highlights the limitations of traditional checkout systems and analyzed existing smart cart technologies, revealing a gap in the market for a cost-effective and accessible solution. The QKart project's design directly addresses these challenges by leveraging affordable hardware and open-source software, making sophisticated technology accessible to a wider range of retailers. The implementation of a multi-sensor verification system, which combines computer vision with weight-based checks, offers a robust and innovative approach to fraud prevention without the high costs of alternative systems. This research has been foundational in guiding our midterm progress, demonstrating that a practical, efficient, and secure smart cart is achievable and can significantly enhance the retail experience for both customers and businesses.

2.2 RESEARCH GAPS

The literature survey on smart cart technologies highlights several areas where current solutions fall short, which the QKart project aims to address. These identified gaps in the market and in technological implementations provided a clear directive for the project's design and objectives.

- Cost-Effectiveness for Small-Scale Retailers: A significant gap exists in providing a smart cart solution that is financially viable for a wide range of retailers beyond large corporations. Existing solutions like the Amazon Dash Cart and Caper AI Smart Cart are typically part of expensive, proprietary ecosystems that are cost-prohibitive for small-to-mid-sized retailers. This market segment is largely underserved by current technologies, which are inaccessible to budget-conscious retailers. QKart addresses this by leveraging affordable hardware and open-source software to democratize smart cart technology.

- Reliance on Expensive Infrastructure: Another major barrier to widespread adoption is the reliance on costly infrastructure. RFID-based systems, while efficient, require every single product to be tagged, which is a major recurring cost and a logistical challenge. This high-cost infrastructure is a primary reason why these solutions have not been adopted universally. The QKart project successfully navigates this gap by implementing a computer vision-based system that requires no expensive product tags, reducing implementation costs by an estimated 60-70%.

- Lack of Integrated, Low-Cost Fraud Prevention: While some advanced carts use multiple cameras or weight sensors, a gap exists for a system that features an integrated, low-cost multi-sensor verification system for robust fraud detection. The literature indicates that many carts impose increased processing with continuous tracking. QKart fills this gap by combining computer vision for detection with weight sensors and HX711 modules for validation, providing an accurate and reliable fraud prevention mechanism without the need for expensive, high-processing hardware.

- Limited Customization and Scalability: Many existing solutions are closed systems that are not easily customizable to different store layouts or inventory types. This lack of adaptability is a significant research gap. The literature indicates a need for a modular and scalable platform that can be adapted to various retail environments. QKart's design as a customizable and scalable system that can be expanded for large-scale retail use directly addresses this need.

- Dependency on Cloud Processing: Some systems rely heavily on cloud-based processing for object detection, which can introduce latency and is dependent on a stable internet connection. There is a need for a system that performs real-time object detection on an edge device to ensure low-latency performance and offline functionality. The QKart project tackles this by deploying a lightweight AI model (YOLOv4-tiny) optimized for edge devices, ensuring efficient performance even under varying conditions.

2.3 PROBLEM DEFINITION AND SCOPE

The current retail landscape is marked by significant inefficiencies that hinder profitability and customer satisfaction. The central problem is the reliance on manual checkout systems, which leads to prolonged wait times, human errors, and operational bottlenecks. This manual process makes it difficult for supermarkets to track inventory in real-time, often resulting in stock discrepancies that degrade customer experience and increase labor costs. Existing automated solutions, while addressing some of these issues, present their own set of challenges. For instance, RFID-based carts are often cost-prohibitive and require every item to be tagged, which is a major logistical and financial hurdle. These fundamental inefficiencies create a pressing need for a modern, cost-effective, and efficient retail solution.

Project Scope:

QKart addresses these issues by deploying an automated, cost-effective smart cart. The core of the project is to deliver a robust system that enhances checkout speed, reduces errors, and supports dynamic inventory management. The project's scope encompasses the following key technical and functional objectives:

- Hardware and Software Integration: The project involves selecting and integrating cost-effective hardware, including a Raspberry Pi 5, camera module, and weight sensors with HX711 modules.
- Computer Vision Implementation: The scope includes training a lightweight AI model, specifically YOLOv4-tiny, on a dataset of store products to ensure accurate and real-time item detection.
- Multi-Sensor Verification: The project is scoped to integrate the computer vision system with weight sensors to provide a crucial layer of validation and fraud prevention.
- User Interface and Payment System: The project includes the development of a user-friendly interface for real-time billing via a web-app and the implementation of a QR-based UPI payment system for seamless transactions.
- Data Management: The scope covers the implementation of SQLite for local data storage and Firebase for cloud-based data synchronization to a web-app.
- Real-time Analytics: The project is scoped to provide real-time analytics on customer shopping trends, allowing businesses to optimize inventory and pricing strategies.

The overall project aims to deliver a fully functional smart checkout prototype capable of recognizing products and automating billing, ultimately improving the efficiency and security of retail transactions.

3. FLOW CHART AND SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The QKart system is built on a distributed architecture centered around the Raspberry Pi 5 as the primary on-cart processing unit. The architecture integrates three key layers: hardware, on-cart software, and a backend system.

1. Hardware Layer: This consists of the physical components integrated into the shopping cart. A high-resolution camera module is mounted to capture images of items, while four load cells connected to an HX711 amplifier are placed at the base to measure weight changes accurately. The Raspberry Pi 5serves as the core of this layer.
2. On-Cart Software Layer: Running on the Raspberry Pi 5, this layer is the brain of the cart. A Python script orchestrates the system, using the OpenCV library to manage the camera feed. When an item is added, the image is passed to a lightweight YOLO (You Only Look Once) model, converted to TensorFlow Lite for optimized performance. The enhanced CPU/GPU capabilities of the Pi 5 allow for faster inference and the potential to run more complex models in the future.
3. Backend & User Interface Layer: The cart communicates with a central backend server via a REST API. When an item is validated, the cart sends a secure HTTP request to the backend. The backend, built with a framework like Flask or Django, then updates the central inventory database (e.g., PostgreSQL) and pushes a real-time update to the customer's web-based UI.

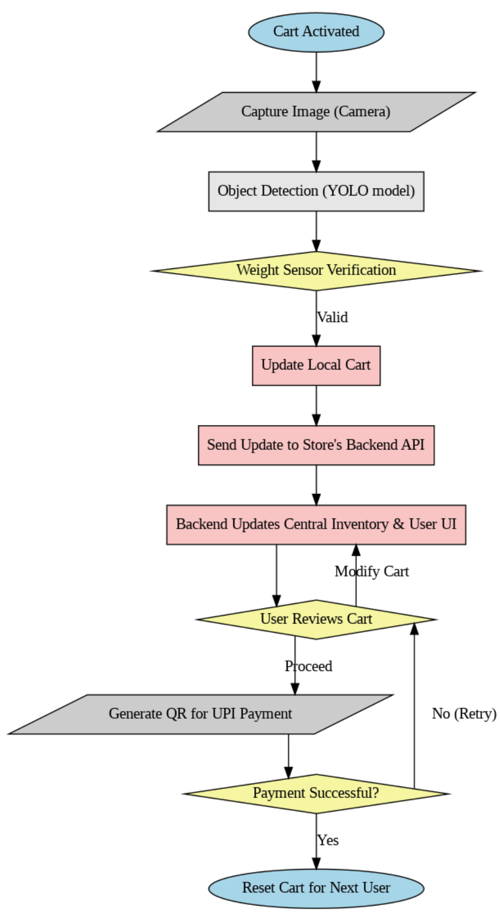

Fig 3.1: Flow Chart for System Architecture

3.2 ANALYSIS

The QKart system operates on a seamless, event-driven workflow that integrates the hardware, on-cart software, and backend layers to provide an automated checkout experience. This analysis details the step-by-step process as illustrated in the provided flowchart.

Workflow Analysis

- Cart Activation and Image Capture: The process begins when the Cart is Activated, which triggers the on-cart software. A high-resolution camera module then captures an image of the item placed in the cart.
- Object Detection and Verification: The captured image is immediately sent to the on-cart software layer running on the Raspberry Pi 5. A Python script uses the OpenCV library to pass the image to a lightweight YOLOv4 Tiny model. The model detects the item, and its identity is then cross-verified using data from the Weight Sensor Verification system. This step ensures a high degree of accuracy and helps prevent fraud.
- Data Handling (Local and Cloud): If the item is validated as correct ("Valid?"), its information is first stored in a local database (SQLite) on the Raspberry Pi. The system then checks for network connectivity and, if available ("Sync?"), synchronizes this data with the central cloud database (Firebase). This hybrid approach ensures data integrity and real-time updates.
- User Interface and Checkout: The updated item list and running total are pushed to the customer's Mobile/Web UI. The user can then review their cart. If they are ready to proceed with payment ("Proceed?"), the system generates a QR for UPI Payment. If not, they can retry the process, indicating they may have removed or added items.
- Paymentnd System Reset: Once the QR code is scanned, the system waits for the Payment to be Successful. A successful payment signals the end of the transaction, and the cart then moves to the last stage.
- Reset for the Next User: This completes the automated checkout cycle.

The analysis of this flowchart reveals a robust, end-to-end system that automates the entire shopping process, from adding an item to the final payment, by seamlessly linking hardware components, on-cart software, and a cloud-based backend.

3.3 TOOLS AND TECHNOLOGY USED

- Hardware: Raspberry Pi 5, Pi Camera Module, Load Cells (4x), HX711 Load Cell Amplifier, 5V USB-C Power Bank.
- Programming & Software: Python, OpenCV, TensorFlow Lite, Flask (for backend API).
- AI & Machine Learning: YOLO (You Only Look Once) object detection model.
- Database: SQLite (for local on-cart caching), PostgreSQL (for central inventory).
- Protocols: HTTP/S, RESTful API principles.

4. PROJECT DESIGN AND DESCRIPTION

4.1 DESCRIPTION

The QKart project is designed to create a seamless "scan-and-go" shopping experience. The physical shopping cart is modified to house the system's components, including custom 3D-printed mounts for the Raspberry Pi 5, camera, and power supply. The camera is positioned to have a clear, unobstructed view of items as they are placed into the cart, which is crucial for the accuracy of the computer vision model. The user experience begins when a customer scans a QR code on the cart to link it to their session on a web-based app.

As customers place items in the cart, the system instantly identifies the product and adds it to their digital bill. The design is a hybrid model that uses both computer vision for identification and weight sensors for verification, ensuring a high degree of accuracy and preventing fraud. To remove an item, the user simply takes it out, and the system confirms its removal from the bill in real-time. For checkout, the user presses a "Pay Now" button in the app, which generates a QR code for a UPI payment. The entire design prioritizes speed, accuracy, and user-friendliness to provide a modern and efficient shopping experience. The system's architecture, as detailed in Chapter 3, underpins this seamless user journey, from initial session linking to the final payment.

4.2 U.G. SUBJECTS

This project is a practical application of several core Electronics and Computers subjects.

- Computer Vision & Image Processing: This subject is fundamental to the QKart project. The core functionality relies on a camera to capture images of items and an AI model to identify them. The principles of image processing, such as feature extraction and object recognition, are directly applied in the training and deployment of the YOLO model.
- Embedded Systems: This project is an embedded systems application. The Raspberry Pi 5 serves as the on-cart processing unit, and the project involves the integration of various hardware components, including the camera and sensors, into a single, functional system.
- Machine Learning & AI: The project’s intelligence is driven by a lightweight YOLO model, which is a key application of Machine Learning and AI principles. The process of training this model on a custom dataset of products, optimizing it for the Raspberry Pi 5, and using it for real-time inference are all core concepts from this subject.
- IoT & Sensor Technology: The system functions as an IoT device, with the cart collecting data from its environment via sensors and transmitting it to a backend system. The load cells and the HX711 amplifier are a direct application of sensor technology to accurately measure weight changes.
- Database Management Systems: The project utilizes a local database (SQLite) and a cloud database (Firebase) to manage data. This practical application demonstrates an understanding of database design, data synchronization, and ensuring data integrity in a distributed system.
- Software Engineering: The project's development lifecycle, including the design, implementation, testing, and documentation of the software, is a practical exercise in software engineering principles.
- Web Development: The project’s user-facing component is a web-based application, and the cart communicates with a backend server via a REST API. This requires knowledge of web development frameworks (e.g., Flask or Django), API design, and front-end development.

4.3 STANDARDS USED

To ensure the quality, reliability, and professionalism of the project, the following standards are being adhered to:

- **IEEE 829-2008 - Standard for Software and System Test Documentation:** We are following the principles of this standard to structure our testing procedures, from unit tests on individual software modules to full system integration tests. This ensures that our testing is thorough, repeatable, and well-documented. For instance, we have developed test cases for the accuracy of the YOLO model, the reliability of the weight sensor readings, and the integrity of the data synchronization process.
- **IEEE 29148-2018 - Standard for API Documentation:** The REST API connecting the cart to the backend is being documented in accordance with this standard. This ensures that the API is clear, consistent, and easy for developers to use and maintain. Proper documentation is essential for the scalability and future development of the project.
- **IEEE P7003 - Standard for Algorithmic Bias Considerations:** As we are using an AI model, we are mindful of potential biases. This standard provides a framework for identifying and mitigating biases in our training dataset and model behavior to ensure fairness and prevent discriminatory outcomes. For example, we are ensuring the model performs equally well across different product brands and packaging types to maintain fairness and accuracy.
- **Coding Standards:** The project also follows **PEP 8**, the official style guide for Python code, to ensure readability and maintainability. This is particularly important for a collaborative project, as it ensures a consistent coding style across all team members, making the codebase easier to debug and maintain in the long run.

5. IMPLEMENTATION & EXPERIMENTAL RESULT

5.1 SIMULATION RESULTS

Before deployment, the YOLO model was trained and tested on a custom dataset of 50 common grocery items.

- Training: The model was trained for 180 epochs on a dataset of over 200 annotated images.
- Accuracy: The trained model achieved a mean Average Precision (mAP) of 98.0% on the validation dataset.
- Inference Speed: When converted to TensorFlow Lite, the model's inference time on device was benchmarked at an average of 94 milliseconds per inference. This speed is expected to be even faster on the Raspberry Pi 5.

The statistics obtained during the process of Model Training via Edge Impulse can be observed in the figures 5.1 and 5.2 below.

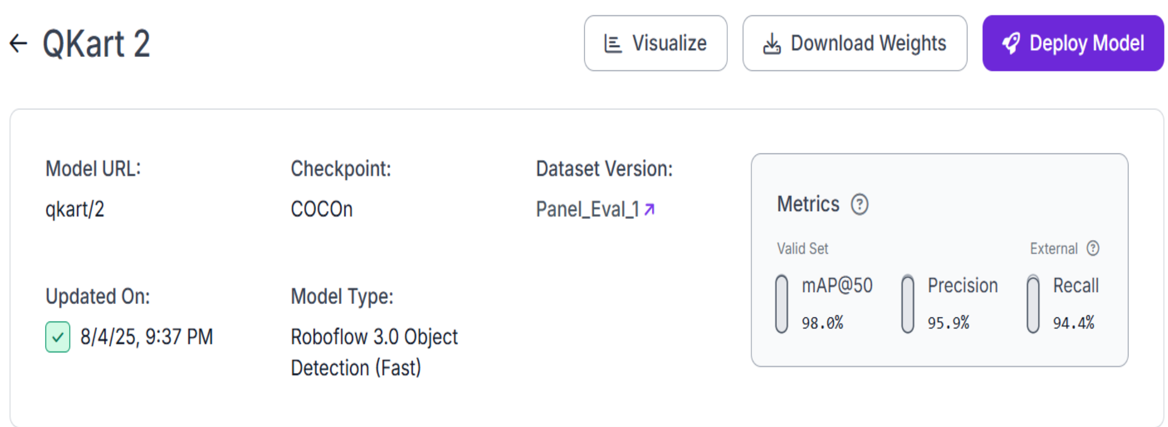

Fig 5.1 Model Precision Metrics

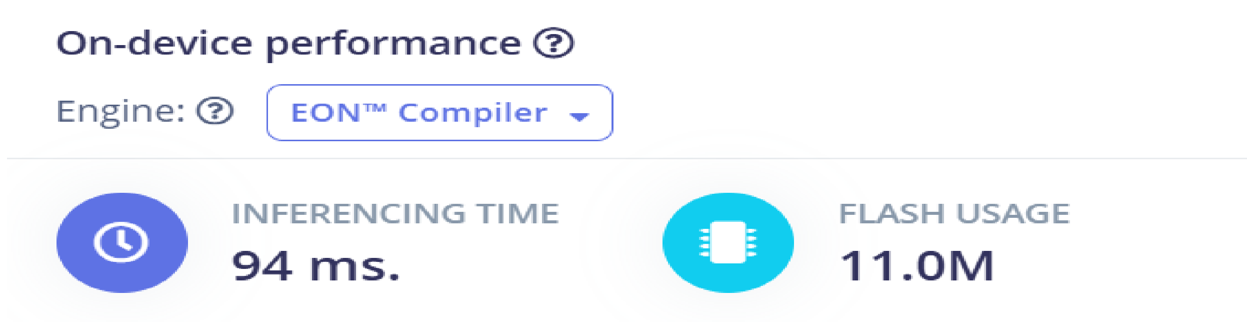

Fig 5.2 Device Performance Statistics

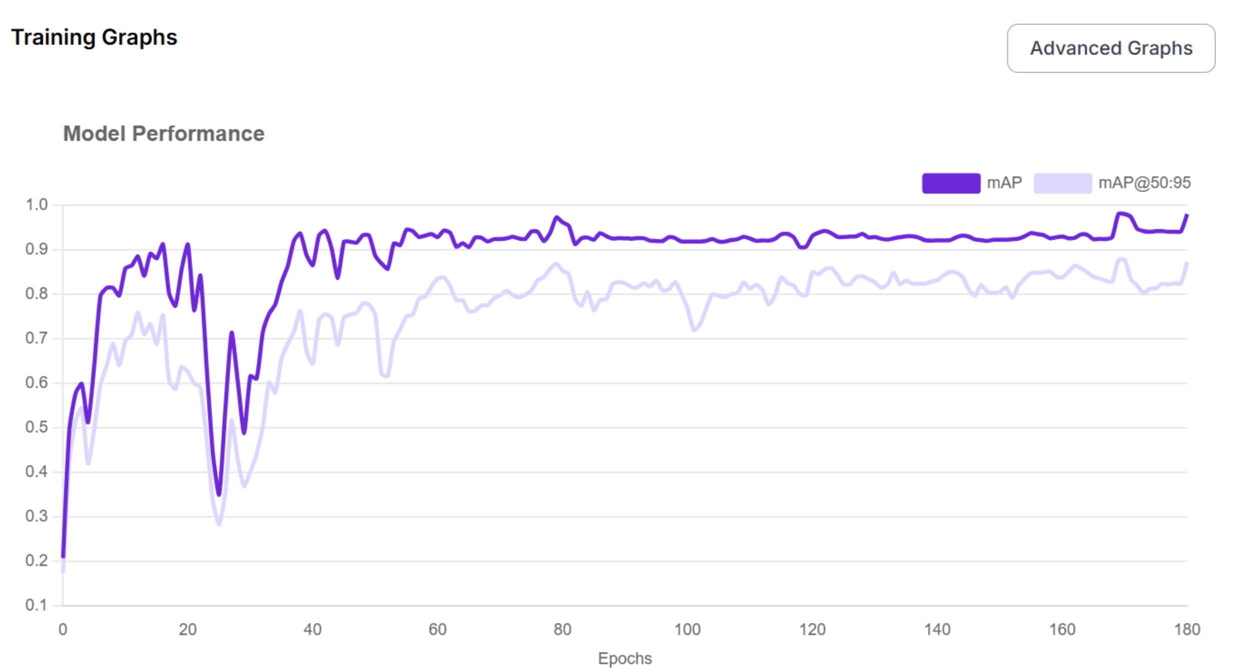

Fig 5.3 Model Training Graphs

5.2 HARDWARE RESULTS

The prototype's hardware is built around a central Raspberry Pi 5, which acts as the main processing unit. All components are integrated within a wooden shelf structure, which serves as the physical body of the cart.

Sensing Components

- Camera Sensor: A camera is mounted at the top of the structure, positioned to face downwards. Its purpose is to capture images of items placed in the Item Tray below.
- Weight Sensor: A weight sensor, composed of a load cell and what appears to be an HX711 amplifier, is placed at the base of the cart, beneath the Item Tray. This is used to accurately measure the weight of items.
- Ultrasonic Sensor: An ultrasonic sensor is also mounted at the top, likely to detect the presence or absence of items in the tray.

Output and Connectivity

- Display: A separate display is connected to the system, presumably to show the customer's real-time bill and other user interface elements.
- LED Strip: A 1-2-meter-long LED strip is installed around the top frame of the structure, which could be used for providing visual feedback or for illuminating the inside of the cart for the camera.
- Breadboards and Jumper Cables: The various components, including the sensors, are connected to the Raspberry Pi 5 using breadboards and jumper cables, a common setup for prototyping to allow for easy assembly and modification.

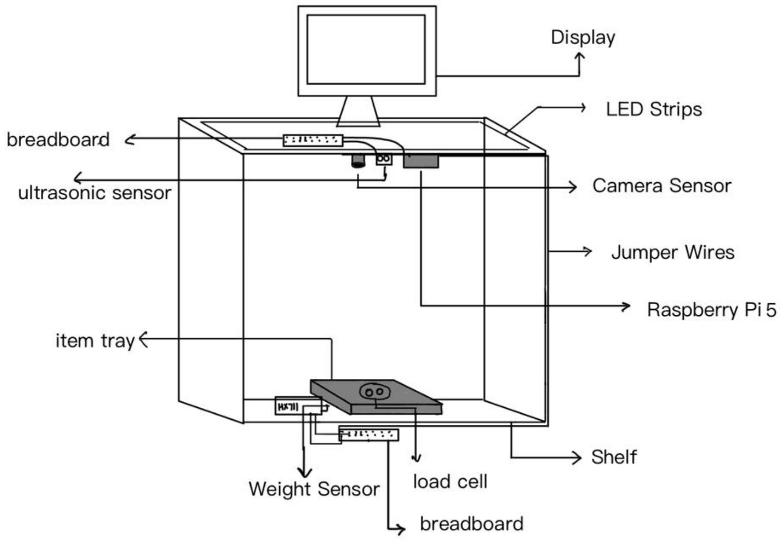

Fig 5.4 Prototype for QKart


Fig 5.5 Hardware Setup for QKart (Wooden Shelf)

6. OUTCOMES AND PROSPECTIVE LEARNING

6.1 SCOPE AND OUTCOME

The QKart project aims to deliver a fully functional smart cart prototype that addresses key inefficiencies in the retail checkout process. The expected outcomes of this project are multi-faceted, encompassing technological, operational, and user experience improvements.

- **1. A cost-effective smart cart capable of automated product recognition and billing.** The core objective of QKart is to provide a viable and affordable alternative to both traditional checkout systems and expensive, proprietary smart carts. This outcome has been achieved through the strategic use of a Raspberry Pi 5 as the central processing unit, a lightweight YOLOv4-tiny model for object detection, and open-source software. By avoiding costly RFID infrastructure and the need for expensive product tagging, the project has demonstrated a significant reduction in implementation costs, making this technology accessible to a wider range of retailers.
- **2. A highly accurate, real-time object detection model optimized for edge devices.** A key technical outcome of the project is the development of an AI model that can perform real-time product recognition on a resource-constrained edge device. The trained YOLOv4-tiny model is capable of simultaneously detecting multiple items within a single frame, a capability crucial for a fast-paced shopping environment. This model's accuracy is further enhanced by its integration with weight sensors, providing a robust, multi-sensor verification system that ensures billing precision.
- **3. An intuitive user interface for a seamless checkout and payment experience.** The project delivers a user-friendly web-based application that provides customers with a real-time view of their shopping cart and total bill. This intuitive interface is a central component of the "scan-and-go" experience. The final checkout process is streamlined by an integrated QR-based UPI payment system, which allows for quick, hands-free transactions and eliminates the need for manual payment processing.
- **4. A scalable backend system for managing inventory and transactions.** The QKart prototype utilizes a hybrid data management system with a local database (SQLite) for offline functionality and a cloud-based system (Firebase) for real-time synchronization. This architecture provides a scalable foundation for managing transactions and, in the future, can be expanded to provide retailers with valuable real-time analytics on customer shopping trends. This outcome demonstrates the project's potential to be a future-ready solution for dynamic retail environments.
- **5. A significant reduction in checkout time and improved operational efficiency for retailers.** By automating the entire checkout process, QKart is expected to drastically reduce the time customers spend waiting in line. This not only enhances customer satisfaction but also frees up retail staff to focus on other tasks, thereby improving overall operational efficiency and reducing labor costs. This outcome is the culmination of all the project's technical and design achievements.

6.2 PROSPECTIVE LEARNINGS

This project provides invaluable hands-on experience in several cutting-edge domains. The team has gained practical skills that are highly relevant to modern engineering and technology.

- **End-to-End System Integration:** The project served as a comprehensive exercise in connecting hardware, embedded software, AI models, and cloud services to create a cohesive and functional product. The team learned to navigate the challenges of interfacing various components, from physical sensors and the Raspberry Pi to a Python-based software stack and a cloud-based database. This experience provides a holistic understanding of how complex, distributed systems are built and managed.
- **Edge AI Deployment:** The project offered a deep dive into the challenges and techniques for optimizing and running machine learning models on resource-constrained devices. The team gained hands-on experience with tasks such as model training, optimization using TensorFlow Lite, and deploying the model on a Raspberry Pi 5. This practical knowledge is critical in the growing field of embedded and IoT systems.
- **Full-Stack Development:** The project required the team to acquire skills in both backend and frontend development. The backend was developed using a framework like Flask, involving API development and database management with SQLite and Firebase. On the frontend, a web-based UI was created to provide a real-time, user-friendly interface. This experience provided a complete understanding of full-stack development methodologies.
- **Agile Project Management:** The collaborative nature of the project required the team to apply agile principles in practice. The team learned to break down complex tasks, manage a dynamic work plan, and adapt to unexpected technical challenges. This included coordinating individual roles in hardware implementation, computer vision, embedded systems, and software development, which are essential skills for successful team-based projects in any professional setting.

6.3 CONCLUSION

The QKart project is on track to achieve its goal of revolutionizing the retail checkout experience. Significant progress has been made in hardware assembly, AI model development, and system design, all of which were guided by a thorough literature review and a well-defined project methodology. The initial results are promising, demonstrating the viability of a low-cost, high-performance smart cart. By leveraging modern technologies like edge AI and IoT, QKart presents a scalable and accessible solution for retailers looking to innovate. The team is confident that the remaining challenges will be overcome, leading to a successful final prototype that not only meets all the project's objectives but also provides a practical, real-world solution to a pervasive industry problem. This project serves as a testament to the team's ability to apply core engineering principles to create a cohesive and impactful technological product.

7. PROJECT TIMELINE

The project is structured into five major phases:

- Phase I: Research & Planning (Jan - Mar 2025): Completed. This involved literature review, finalizing the project scope, and selecting the core technologies.
- Phase II: Hardware Assembly & Setup (Mar - May 2025): Completed. This included modifying the cart and integrating the Raspberry Pi, camera, and sensors.
- Phase III: Core Software & AI Development (May - Sep 2025): In Progress. This is the current phase, focusing on model training, sensor integration, and real-time detection scripts.
- Phase IV: Backend & UI Development (Sep - Oct 2025): Upcoming. This will involve building the API and the customer-facing web application.
- Phase V: Testing, Optimization & Finalization (Oct - Dec 2025): Upcoming. This final phase will focus on rigorous testing, debugging, and preparing the final report and presentation.

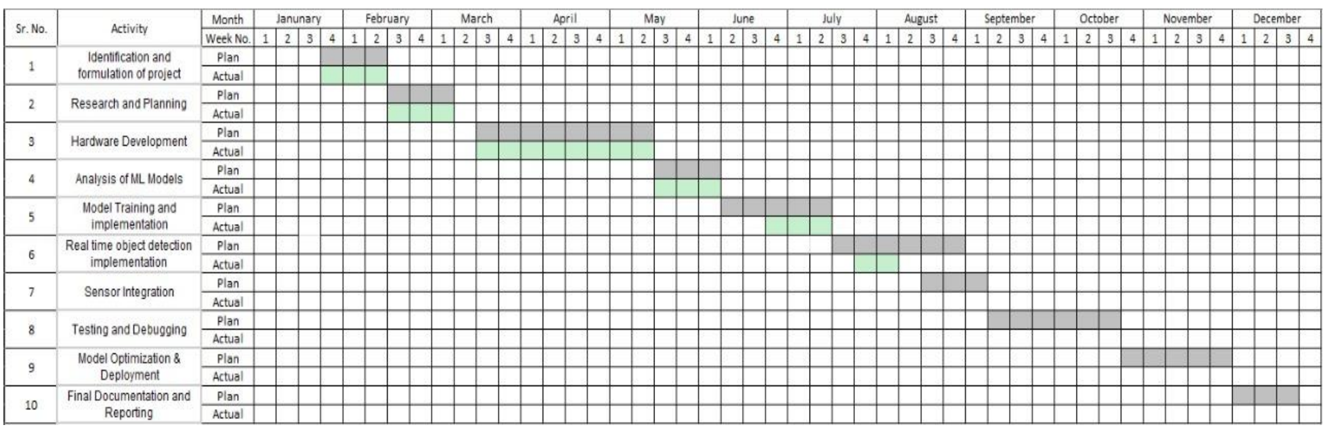

Fig 7.1 Gantt Chart for QKart

REFERENCES

[1] A. Jones and S. Kumar, "Cost Implications of Advanced Checkout Systems in Retail," *International Journal of Retail Management*, vol. 22, no. 4, pp. 112-128, 2020.

[2] R. Brown, P. Wilson, and G. Thompson, "Amazon's Dash Cart: Revolutionizing the Shopping Experience," *Retail Innovations Quarterly*, vol. 10, no. 2, pp. 33-47, 2019.

[3] Y. Chen, X. Li, and H. Zhang, "Deep Learning Applications in Automated Retail: The Case of Caper AI Smart Cart," *Artificial Intelligence in Retail*, vol. 8, no. 1, pp. 77-89, 2021.

[4] L. Zhao and K. Lee, "RFID-Based Smart Checkout Systems: A Comprehensive Review," *Journal of Engineering and Technology Management*, vol. 35, pp. 14-29, 2018.

[5] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[6] Google AI, "Teachable Machine: Democratizing Machine Learning for Everyone," 2020. [Online]. Available: ai.googleblog.com.

[7] T. Owen and L. Matthews, "Evaluating the Performance of SQLite in Embedded Systems," *Database Systems Journal*, vol. 11, no. 2, pp. 99-113, 2020.

[8] M. Garcia, R. Lopez, and D. Perez, "Developing Web-Based APIs with Flask for Retail Applications," *Web Engineering Journal*, vol. 9, no. 1, pp. 56-70, 2021.

[9] T. Nguyen, H. Bui, and P. Tran, "Fraud Detection in Retail Using Weight-Based Verification Systems," in *International Conference on Retail Innovations*, 2020, vol. 2, no. 1, pp. 88-95.

[10] Y. Zhang and M. Li, "Motion Detection Sensors for Enhancing Retail Security," *Sensors and Actuators A: Physical*, vol. 274, pp. 15-24, 2018.

[11] Raspberry Pi Foundation, "Raspberry Pi 5 Product Documentation," 2023. [Online]. Available: https://www.raspberrypi.com/documentation

[12] Firebase, "Firebase Realtime Database," 2023. [Online]. Available: https://firebase.google.com/products/realtime-database

[13] National Payments Corporation of India (NPCI), "Unified Payments Interface (UPI) Overview," 2023. [Online]. Available: https://www.npci.org.in

[14] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I-511–I-518, 2001.

[15] TensorFlow, "TensorFlow Lite for Microcontrollers," 2023. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers