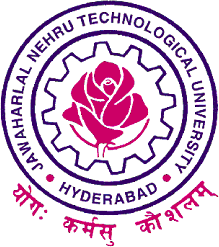
**­­­Iot Based Smart Shopping Trolley Using Arduino**

#### An Industry Oriented Mini Project Report

***Submitted to***

## Jawaharlal Nehru Technological University

Hyderabad

*In partial fulfillment of the requirements for the award of the degree of*

### BACHELOR OF TECHNOLOGY

**in**

### ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

By

**M.NIKHITHA (22VE1A66A8)**

**SYED RUDAAR (22VE1A66C3)**

**E.SATHVIK (22VE1A6683)**

**N.ROHAN (22VE1A66A7)**

#### Under the Guidance of

#### Dr.A.Swathi

**Associate Professor**

****

### DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NAAC “A” Grade & NBA| Hyderabad | PIN: 500068

(2020-2024)



### DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NAAC “A” Grade & NBA| Hyderabad | PIN: 500068

Certificate

This is to certify that the Industry Oriented Mini Project Report on ***“Iot Based Smart Shopping Trolley Using Arduino***” submitted by **M.NIKHITHA, SYED RUDAAR, E.SATHVIK, N.ROHAN** bearing Hall Ticket No’s. **22VE1A66A8, 22VE1A66C3, 22VE1A6683, 22VE1A66A7** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Artificial Intelligence & Machine Learning** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad for the academic year 2024-25 is a record of bonafide work carried out by him/her under our guidance and Supervision.

**Internal Guide Head of the Department**

**Dr.A.** **Swathi Dr. A. Swathi**

**Assoc. Professor**

**Project Coordinator Signature of the External Examiner Mrs. B. Spandana**



### DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NAAC “A” Grade & NBA| Hyderabad | PIN: 500068

# DECLARATION

We **M.NIKHITHA, SYED RUDAAR, E.SATHVIK, N.ROHAN** bearing Roll No’s **22VE1A66A8, 22VE1A66C3, 22VE1A6683, 22VE1A66A7** hereby declare that the Project titled *"****Iot Based Smart Shopping Trolley Using Arduino***” done by us under the guidance of **Dr.A.Swathi**, which is submitted in the partial fulfillment of the requirement for the award of the B.Tech degree in **Artificial Intelligence & Machine Learning** at **Sreyas Institute of Engineering & Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

**M.NIKHITHA (22VE1A66A8)**

**SYED RUDAAR (22VE1A66C3)**

**E.SATHVIK (22VE1A6683)**

**N.ROHAN (22VE1A66A7)**

**CO-PO mapping of “Iot Based Smart Shopping Trolley Using Arduino”**

**Course Outcomes of the Project:**

**CO1**: **Identify and define** real-world problems that can be effectively addressed using IoT-based solutions, considering sensor integration and connectivity requirements.

**CO2**: **Design** an IoT architecture incorporating appropriate hardware components (sensors, microcontrollers, and actuators) and communication protocols (e.g., MQTT, HTTP, Zig Bee).

**CO3**: **Develop and implement** functional IoT prototypes using platforms such as Arduino, Raspberry Pi, or ESP32, and interface them with cloud services or mobile/web applications.

**CO4**: **Analyse and evaluate** the performance of the IoT system in terms of data accuracy, latency, power consumption, and scalability.

**CO5**: **Demonstrate** effective time and resource management, and awareness of security, privacy, and ethical implications in IoT deployments.

**CO6**: Collaborate to integrate individual contributions into a team effort and complete the design.

**Program Outcomes of the Department:**

Engineering Graduates will be able to:

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Program Specific Outcomes (PSOs) of the Department:**

1. Graduates will apply programming to implement various domains in computer science and Machine learning algorithms. They’ll utilize mathematical foundations such as linear algebra and calculus, while optimizing AI models across different hardware and leveraging principles of operating systems and computer organization.

2. Develop professional skills in the thrust areas like ANN, Deep learning and Data Analytics and pursue higher studies in Artificial Intelligence in reputed Universities and to work in research establishments.

**CO-PO MAPPING:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| **CO1** | 3 | 3 | 2 | 2 |  |  |  |  | 1 | 1 |  | 2 | 3 | 2 |
| **CO2** | 3 | 2 | 3 | 2 | 2 |  |  |  | 1 | 1 | 1 | 2 | 3 | 3 |
| **CO3** | 3 | 2 | 3 | 2 | 3 |  |  |  | 1 | 1 | 1 | 2 | 3 | 3 |
| **CO4** | 3 | 3 | 2 | 3 | 2 |  |  |  |  | 1 | 1 | 2 | 3 | 3 |
| **CO5** |  |  | 1 |  |  |  |  |  | 3 | 3 | 2 | 2 | 2 | 2 |
| **CO6** |  | 1 | 2 |  |  |  |  |  | 3 | 2 | 3 | 2 | 2 | 2 |

# ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success.

We take this opportunity to acknowledge with thanks and a deep sense of gratitude to **Dr.A.Swathi**, **Associate Professor**, for her constant encouragement and valuable guidance during the project work.

A Special vote of Thanks to **Dr. A. SWATHI (Head of the Department, AIML) and Mrs. B. Spandana (Project Co-ordinator)** has been a source of Continuous motivation and support. They had taken time and effort to guide and correct me all through the span of this work.

We owe everything to the **Department Faculty, Principal** and the **Management** who made my term at Sreyas Institute of Engineering and Technology a stepping stone for my career. I treasure every moment I have spent in college.

Last but not the least, my heartiest gratitude to my parents and friends for their continuous encouragement and blessings. Without their support, this work would not have been possible.

**M.NIKHITHA (22VE1A66A8)**

**SYED RUDAAR (22VE1A66C3)**

**E.SATHVIK (22VE1A6683)**

**N.ROHAN (22VE1A66A7)**

# CHAPTERS INDEX

CHAPTER1. INTRODUCTION 1

1.1problem statement 2

CHAPTER 2. LITERATURE SURVEY 6

* 1. Existing system 6
  2. proposed system 8

CHAPTER 3. SYSTEM DESIGN… 10

* 1. importance of design 10
  2. uml diagrams 10
     1. use case diagram 10
     2. sequence diagram 11
     3. activity diagram 12
     4. system architecture 14
  3. functional requirements 14

[CHAPTER 4. IMPLEMENTATION… 1](#_TOC_250004)7

* 1. Module Description 17
  2. [Sample Code 1](#_TOC_250003)9
  3. Web Development(Back-End) 27

CHAPTER 5. TESTING 28

CHAPTER 6. RESULTS 30

CHAPTER 7. CONCLUSION 34

CHAPTER 8. FUTURE SCOPE 35

CHAPTER 9. REFERENCES **36**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Name of Figure** | **Page No.** |
| Figure 1.2.1 | Bluetooth Module | 4 |
| Figure 3.1.1.1 | Usecase Diagram | 11 |
| Figure 3.1.2.1 | Sequence Diagram | 12 |
| Figure 3.1.3.1 | Activity Diagram | 13 |
| Figure 3.2.1 | System Architecture | 14 |
| Figure 6.1 | Smart Cart IoT Device | 30 |
| Figure 6.1.1 | RFID Product Identification | 31 |
| Figure 6.2.1 | LCD Output | 32 |
| Figure 6.3.1 | Billing Logic | 33 |

**ABSTRACT**

In today’s fast-paced world, shopping at large malls often leads to long queues and delays at billing counters. To address this issue, the Smart Shopping Trolley integrates RFID and Bluetooth technology to automate the billing process, reducing both wait times and manpower. Each product in the store is tagged with an RFID tag, which, when scanned by the RFID reader on the trolley, updates the cart’s total cost in real-time. This ensures that customers are always aware of their spending, helping them stay within budget. The system consists of an RFID module, a microcontroller, and a Bluetooth module for seamless data processing and wireless communication. When a customer adds or removes a product, the updated price is displayed instantly on an LCD screen. To finalize the purchase, pressing the billing button transmits the total bill to the user’s Android mobile app via Bluetooth communication, eliminating the need for manual checkout. The RFID-based embedded system automates billing by integrating an RFID reader that detects RFID tags assigned to each product. These tags contain a unique EPC (Electronic Product Code), which the microcontroller unit (MCU) processes to update the total cost dynamically. Unlike barcode-based systems, RFID enables non-line-of-sight (NLOS) scanning, allowing multiple items to be detected simultaneously. By integrating RFID authentication, embedded microcontroller processing, and Bluetooth based wireless billing, the Smart Shopping Trolley enhances retail automation, operational accuracy, and transaction efficiency. This system reduces human intervention, optimizes real time transaction processing, and accelerates contactless billing. The implementation of RFID and Bluetooth technologies in retail environments not only improves convenience but also enhances data security, scalability, and efficiency, making it an ideal solution for modern smart retail systems.

**Keywords:** RFID, Bluetooth, Microcontroller, Billing Automation, Wireless Communication, Real-Time Processing.

**CHAPTER 1**

**INTRODUCTION**

Over the past few decades technology has been drastically changing the way of life in the society hence lives of the current generation has been significantly improved by completely becoming dependent on new technologies. It also allows multiple tasks to be completed in less time along with increased efficiency and creates entirely new way to make a living. Shopping mall is a place where most of the people hurry to get their daily necessities. However, shopping carts in major stores have experienced little changes and serve only for simple purposes ever since they have been first manufactured. Consumers often face problems as they are tedious and in convenience to push or pull these carts around and becomes noticeably painful to carry the heavy loads. After this one, needs to go to billing counter for payments which is a time-consuming process and also results in a long queue at the counters. Our aim is to design an intelligent cart that follows the customer and makes billing simple which provides more satisfaction and comfortable for the customers.

In the fastest growing technological world, Automation is an emerging technology in the field of research and development. Shopping carts are used in order to aid customers to store the products which they intend to purchase. Then the customer has to push the heavily loaded cart and proceed to checkout at the billing counter by making their payments. Shopping is one of our regular tasks where we spent a considerable amount of time for buying the necessary things. In supermarkets, everything is in a manual process and now, in this emerging world everyone needs to save their precious time so, many have started to purchase online which is much easier and flexible. By making a manual process into an automated one, will reduce the manpower and saves more time.The aim of the project is to the implement an automated human following intelligent cart that offers great convenience to the customers, which eliminates human labour to push heavy loaded carts and avoids waiting in long queues and for speeding up the billing and payment process. The cart can thus reduce the manpower and can reduce the cost of workers hired. It also helps in promotion of the super market by gaining more customers there by providing quick service. The design module consists of the microcontroller and is able to move automatically by the commands given through Bluetooth under the drive of 12V DC motors, whose voltage input is controlled by Arduino to control the movement of the cart. The Bluetooth, works as the wireless information transfer. Thereby producing a versatile cart, this automatically follows the human to bring the stuff that makes shopping easier and efficient.

* 1. **Problem Statement**

In conventional retail environments, customers frequently encounter inefficiencies such as long queues at billing counters, manual item scanning errors, and lack of real-time visibility into the total cost of items selected. These issues not only cause inconvenience but also lead to customer dissatisfaction and slower store operations. With the increasing adoption of smart technologies, there is a growing need to digitize and automate the shopping process. An IoT-based Smart Shopping Trolley using Arduino addresses this challenge by integrating technologies like RFID or barcode scanners, weight sensors, and wireless communication modules (such as Wi-Fi or Bluetooth). As customers place products into the trolley, each item is automatically identified and its price is added to the running total, which is displayed on an LCD screen or mobile app in real-time. The system can also send purchase data to a central database for inventory management and allow for secure digital payment directly from the trolley. This eliminates the need for manual billing, minimizes human errors, optimizes staff workload, and greatly enhances the overall shopping experience by making it faster, smarter, and more customer-friendly.

**1.2 Components**

**RFID TAGS**

A radio-frequency identification system uses tags, or labels attached to the objects to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response.

RFID tags can be either passive, active or battery assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery assisted passive (BAP) has a small battery on board and is activated when in the presence of a RFID reader. A passive tag is cheaper and smaller because it has no battery.

Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be write-once, read-multiple; "blank" tags may be written with an electronic product code by the user.The tag's information is stored electronically in a non-volatile memory. The RFID tag includes a small RF transmitter and receiver. An RFID reader transmits an encoded radio signal to interrogate the tag. The tag receives the message and responds with its identification information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. RFID tags contain at least two parts: an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, collecting DC power from the incident reader signal, and other specialized functions; and an antenna for receiving and transmitting the signal.

**RFID READER**

RFID systems can be classified by the type of tag and reader. A Passive Reader Active Tag (PRAT) system has a passive reader which only receives radio signals from active tags (battery operated, transmit only). The reception range of a PRAT system reader can be adjusted from 1-2,000 feet, allowing flexibility in applications such as asset protection and supervision.

An Active Reader Passive Tag (ARPT) system has an active reader, which transmits interrogator signals and also receives authentication replies from passive tags. An Active Reader Active Tag (ARAT) system uses active tags awoken with an interrogator signal from the active reader. A variation of this system could also use a Battery Assisted Passive (BAP) tag which acts like a passive tag but has a small battery to power the tag's return reporting signal.

Fixed readers are set up to create a specific interrogation zone which can be tightly controlled. This allows a highly defined reading area for when tags goin and out of the interrogation zone. Mobile readers may be hand-held or mounted on carts or vehicles.

The RFID tag can be affixed to an object and used to track and manage inventory, assets, people, etc. For example, it can be affixed to cars, computer equipment, books, mobile phones, etc.RFID offers advantages over manual systems or use of bar codes. The tag can be read if passed near a reader, even if it is covered by the object or not visible. The tag can be read inside a case, carton, box or other container, and unlike barcodes, RFID tags can be read hundreds at a time. Bar codes can only be read one at a time using current devices. The detailed explanation about RFID technology is provided in the further chapters.

**BLUETOOTH MODULE**

The **Bluetooth wireless technology** is set to revolutionize the way people perceive digital devices in our homes and office environment. Now they are no longer just the individual devices; instead, with the embedded Bluetooth technology, they form a network in which appliances can communicate with each other. This wireless technology is especially useful in short rage wireless communication, where there exists hardly any infrastructure. Operating over unlicensed, universally available frequency of 2.4 GHz, it can link digital devices within a range of 10 m.Building upon this theme is designed to control a robotic vehicle based on **Bluetooth technology on android platform**.

This project work consists of two main modules: the android mobile phone and the Arduino BT board.The android mobile phone consists of several Bluetooth apps which enables the user to access the control commands. In this project we are targeting Android platform since it has huge market and open source. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android OS is based on Linux. Android Applications are made in a Java-like language running on a virtual machine called ‘Dalvik’ created by Google. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

A Bluetooth module is usually a hardware component that provides a wireless product to work with the computer; or in some cases, the Bluetooth may be an accessory or peripheral, or a wireless headphone or other product has hardware support to use Bluetooth products and connections, then whatever it is you are trying to download and use, may work.

There are component Bluetooth wireless modules with a USB plug on them to add this BT to computers that did not have it built-in. Some of these modules may require drivers, but usually Mac OS X has drivers built into the system to support several products. The below figure shows the model of Bluetooth module.



Figure 1.2.1:Bluetooth Module

These small size Bluetooth TTL transceiver modules are designed for serial communication (SPP - serial port profile). It allows your target device to both send or receive TTL data via Bluetooth technology without connecting a serial cable to your computer. The modules with the HC-03 and HC-05 firmware are the Master and Slave integrated Bluetooth serial modules with firmware which allows you to modify master and slave mode at any time. HC-03 is industrial grade products, HC-05 are commercial grade products.The modules with the HC-04 and HC-06 firmware are the modules which are factory set to be Master or Slave modules. Master and slave mode cannot be switched from the factory setting. HC-04 is an industrial grade product; HC-06 is a commercial grade product. The modules with the HC-09 firmware are replacements for the HC-06 and HC-07 modules. The controller communicates with the Bluetooth module through AT commands. More details about the Bluetooth technology is provided in the further chapters.

**ANDROID**

In recent years, an open-source platform Android has been widely used in smart phones. Android has complete software package consisting of an operating system, middleware layer and core applications. Different from other existing platform like iOS (iPhone OS), it comes with software development kit (SDK), which provides essential tools and Application. Using a Smartphone as the “brain” of equipment is already an active research field with several open opportunities and promising possibilities. In our work, the home appliances are controlled to either switch ON or switch OFF using voice commands through the android device.

# ARDUINO UNO controller

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Overview of Existing Systems**

Traditional retail shopping generally follows one of three common methods. The most widely used is the barcode-based checkout system, where customers pick their desired items, proceed to the billing counter, and each product’s barcode is manually scanned by a cashier. A more modern approach includes self-checkout kiosks, where customers independently scan items and complete payment at self-service stations. In a few advanced retail environments, RFID-based smart carts have been introduced. These carts are equipped with RFID technology to automatically detect items placed in the trolley, reducing the need for manual scanning. However, such systems are still in limited use due to cost and implementation challenges.

Smith and Johnson(2018) Smith and Johnson introduced one of the earliest RFID + BLE smart‐cart prototypes. Their trolley continuously scans tag IDs and pushes instant billing updates via Bluetooth to a paired smartphone app, alerting users when they exceed preset budgets. While the system dramatically cut checkout times in pilot tests, they observed BLE’s limited range often forced shoppers to stay within two meters of the cart, and they raised privacy concerns around unencrypted RFID transmission.

Patel et al. (2019)Patel and colleagues enhanced item verification by combining RFID with built-in weighing scales. Each product’s weight is cross-checked against its tag-registered mass, flagging mis-scans or untagged items automatically. They also implemented dynamic, server-driven price updates and a shelf-map UI to guide customers to nearby offers. However, repeated calibration was required to counteract scale drift, and the added hardware roughly doubled manufacturing costs.

Chen & Li (2020) Chen and Li replaced RFID altogether in their camera-vision cart. A ceiling-mounted depth camera tracks items placed in the basket and matches them to a product image database, enabling completely tag-free billing. Their touchscreen interface lets customers search for product details in real time. Unfortunately, heavy compute demands meant only one cart per store could be supported on their edge server, and lighting changes sometimes led to misclassifications.

Almeida et al. (2021) Almeida’s team built a cloud-connected RFID cart network where every scan—and every cart’s running total—is mirrored to a central server. Shoppers can pause and resume a session at any cart in the chain, and store managers get live fraud-detection analytics. The dependency on continuous internet connectivity, however, meant that any local network outage would completely halt billing until connectivity was restored.

Nguyen et al. (2022) Nguyen and co-authors fused RFID scanning with on-board machine-learning inference (“Edge-AI”) to detect anomalies like duplicate scans or tag removals without cloud help. Their trolley classifies items via a lightweight neural network and falls back to cached price lists when connectivity drops. They reported that updating the edge models in deployed carts posed significant logistical challenges and sometimes led to version mismatches.

García & Martínez (2017) García and Martínez explored dual‐mode tag reading—RFID plus NFC—to maximize compatibility with existing product labels and customer smartphones. Their cart also integrates a cross-store loyalty module: shoppers tap their phones via NFC to redeem points automatically. They noted that the NFC reader’s very short range (a few centimeters) frequently frustrated users trying to authenticate at speed.

Rao et al. (2020) Rao’s group leveraged Lora WAN for intra-store cart communications, allowing data packets to travel hundreds of meters with minimal power. Carts can geo-fence different store zones (e.g., produce, dairy), generating location-aware recommendations. The trade-off they observed was a very low data throughput, resulting in a 2–3-second lag whenever multiple carts updated simultaneously in high-density areas.

Zhang & Wang (2021)Zhang and Wang augmented RFID with ultrasonic distance sensors to estimate 3D item volume and detect collisions between carts. Their hybrid approach cross-validates weight, tag presence, and physical dimensions to catch scanning errors. However, noisy aisles (forklift movements, crowded lanes) often triggered false distance readings, and the extra sensors increased the cart’s bulk by over 30%.

Kaur & Singh (2023)Kaur and Singh created a BLE-mesh network connecting every cart in the store, enabling peer-to-peer session hand‐offs and group-checkout coordination. Shoppers can merge carts’ bills when shopping in pairs or families. While innovative, their mesh overlay required periodic maintenance to heal dropped connections, and performance degraded noticeably at peak store occupancy.

**2.2 Limitations of Existing Systems**

Despite their individual strengths, the current landscape suffers from common deficiencies:

· Limited processing power of Arduino makes it hard to handle complex tasks

· Limited memory and storage capacity

· Lacks built-in Wi-Fi or Bluetooth unless using additional modules

· Unstable or slow wireless connectivity in crowded areas

· Frequent power drainage with multiple modules

**2.3 Proposed System: Smart Shopping Trolley**

The proposed system for an IoT-based Smart Shopping Trolley using Arduino is designed to automate and simplify the shopping experience. At its core, the system uses an Arduino Uno, NodeMCU, or ESP32 microcontroller to control the overall operations. Each product in the store is embedded with a unique RFID tag. As a customer places an item in the trolley, an RFID reader (such as the RC522) scans the tag to identify the product and retrieve its details.

A Bluetooth module (HC-05) or a Wi-Fi module (ESP8266) is used for wireless communication between the trolley and a mobile application or a cloud server. The trolley is equipped with an LCD or LED display that shows the scanned product names, quantities, and the running total of the bill. A buzzer is used to alert the customer in case of errors, such as item removal without billing or invalid scans.

To ensure the accuracy of the products being added, a load cell (weight sensor) can be integrated to verify the actual weight of items placed in the trolley. A dedicated mobile app or a web-based dashboard allows customers to view their cart in real time, receive product recommendations, and check out without standing in long queues. Cloud integration helps store purchase history and billing data for both users and store management.

The system is powered using a rechargeable battery pack with a voltage regulator to maintain consistent operation. For enhanced functionality, optional modules like a GPS unit can be added for tracking trolley locations inside large stores, and QR/barcode scanners can be used alongside RFID for a hybrid item identification system. The final step involves automatic billing verification at the exit gate, where all RFID tags are rechecked before payment processing, ensuring a secure and seamless shopping experience.

* 1. **Justification for the Proposed Model**

The proposed model for an IoT-based Smart Shopping Trolley using Arduino is justified by its ability to significantly enhance the shopping experience through automation, efficiency, and convenience. By integrating RFID technology, the system enables automatic product detection without the need for manual barcode scanning, which reduces human error and speeds up the billing process. This is particularly beneficial in large retail stores where long queues at billing counters are a common issue.

Additional features like load cells ensure product verification, preventing theft or accidental item mismatches. Integration with mobile apps and cloud services also provides scalability, making it easier to manage multiple trolleys across a store. The overall system not only reduces the workload on store employees but also enhances customer satisfaction by offering a self-service, contactless, and faster checkout process. This model aligns well with the growing demand for smart retail solutions in a digital, post-pandemic world.

**CHAPTER 3**

**SYSTEM DESIGN**

The IoT-based Smart Shopping Trolley using Arduino is designed to automate the shopping and billing process in supermarkets. It uses an **RFID reader** to detect products with RFID tags, eliminating the need for manual barcode scanning. An **Arduino** or **ESP32 microcontroller** processes the data and updates the cart information in real-time. A **load cell** (weight sensor) can verify item weight for accuracy and theft prevention. A **display screen** shows the list of scanned items and the total bill to the customer. **Bluetooth or Wi-Fi modules** enable wireless communication with a mobile app or cloud system. The mobile app allows users to monitor their cart and make digital payments. A **buzzer** alerts for errors like unauthorized item removal. At the store exit, RFID re-scanning validates the final cart for billing. This system reduces manual labor, minimizes checkout time, and enhances the overall shopping experience.

**3.1 UML Diagrams**

The Unified Modeling Language (UML) diagrams provide a structured visualization of the Smart Content Aggregator’s internal and external behaviors. These diagrams depict the functional, behavioral, and interaction-level components of the system. Specifically, three diagrams are used to represent different perspectives of the system:

1. The Use Case Diagram outlines the high-level interactions between users and system functionalities.
2. The Sequence Diagram shows the time-sequenced communication between system components.
3. The Activity Diagram details the internal flow of operations during a typical use session.

Each diagram is presented and explained in the sections below.

**3.1.1 Use Case Diagram**

This use case diagram illustrates the working of a Smart Shopping Trolley System designed to automate and simplify the shopping experience. The system involves key components such as the customer, RFID reader, Bluetooth module, and an Android app. As the customer adds or removes products, the RFID reader scans the items, and the system updates the virtual cart accordingly. The total cost can be viewed in real time, and once shopping is complete, the system generates a bill and processes payment through a connected Android app via Bluetooth. The diagram clearly shows how each component interacts to provide a seamless, efficient, and user-friendly shopping process.

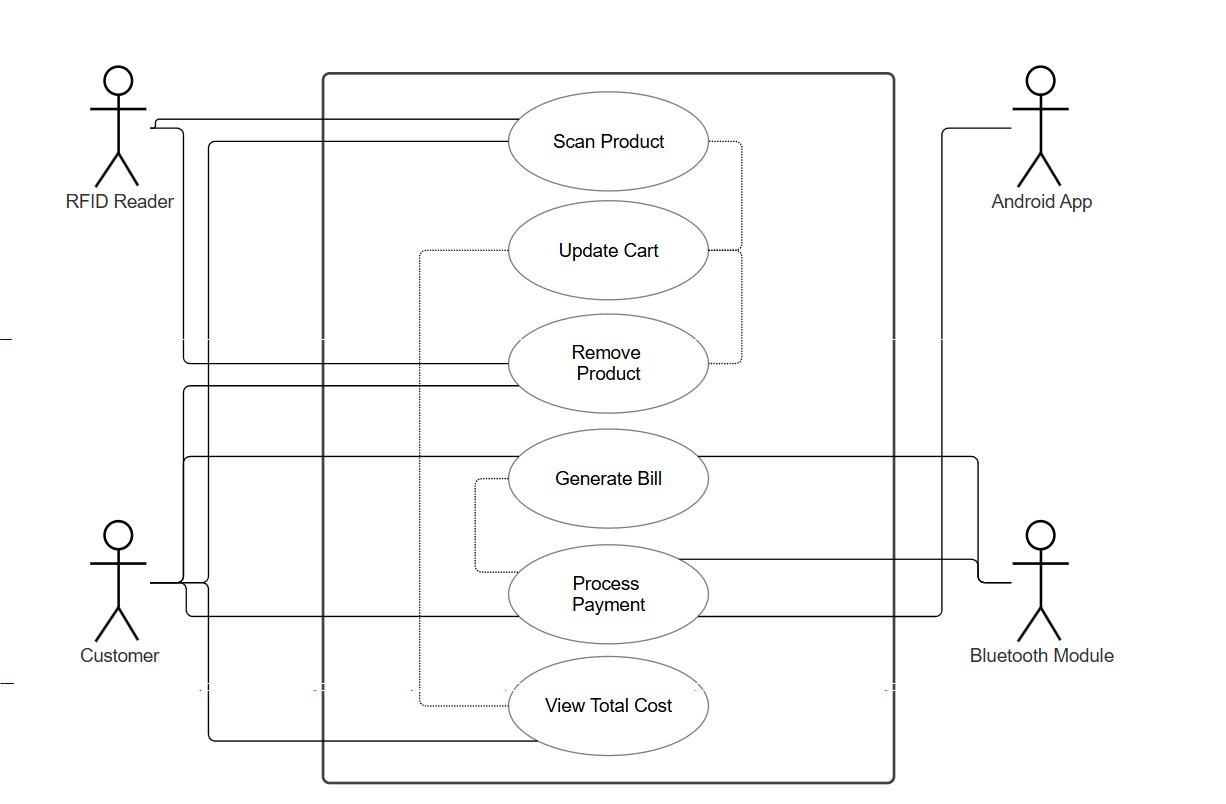
****

Figure 3.1.1.1: UseCase Diagram for Smart Shopping Trolley System

**3.1.2 Sequence Diagram**

This sequence diagram depicts the step-by-step interaction between the components of the Smart Shopping Trolley System during a typical shopping session. It begins when the customer scans a product using the RFID reader. The product data is then sent to the microcontroller, which processes the information and updates the display on the LCD to show the updated cart total. If the customer decides to remove an item, a similar process follows where the RFID reader captures the removal, and the microcontroller updates the cart and the LCD. Once the shopping is complete, the customer requests the bill, which the microcontroller sends via the Bluetooth module to the Android app. The Android app processes the payment and sends a confirmation back through the same path. Finally, the LCD displays a thank you message to the customer, and payment confirmation is shown, completing the smart shopping experience.

**A screenshot of a computer screen

AI-generated content may be incorrect.**

Figure 3.1.2.1: Sequence Diagram for Smart Shopping Trolley System

**3.1.3 Activity Diagram**

This activity diagram represents the workflow of a Smart Shopping Trolley System from the start of shopping to the successful completion of payment. The process begins when the customer starts shopping and scans a product. If a product is detected, the cart total is updated and displayed on the LCD screen. The customer is then asked if they want to continue shopping. If yes, the process loops back; if not, they are asked if they wish to remove an item. If removal is required, the product is scanned again for removal, and the cart is updated. Otherwise, the system proceeds to generate the final bill, transmit it via Bluetooth, and process the payment. The loop continues until a successful payment is confirmed. Once completed, the system ends the shopping session. This flow clearly depicts how user interaction, automation, and decision-making are integrated for a seamless self-service shopping experience.

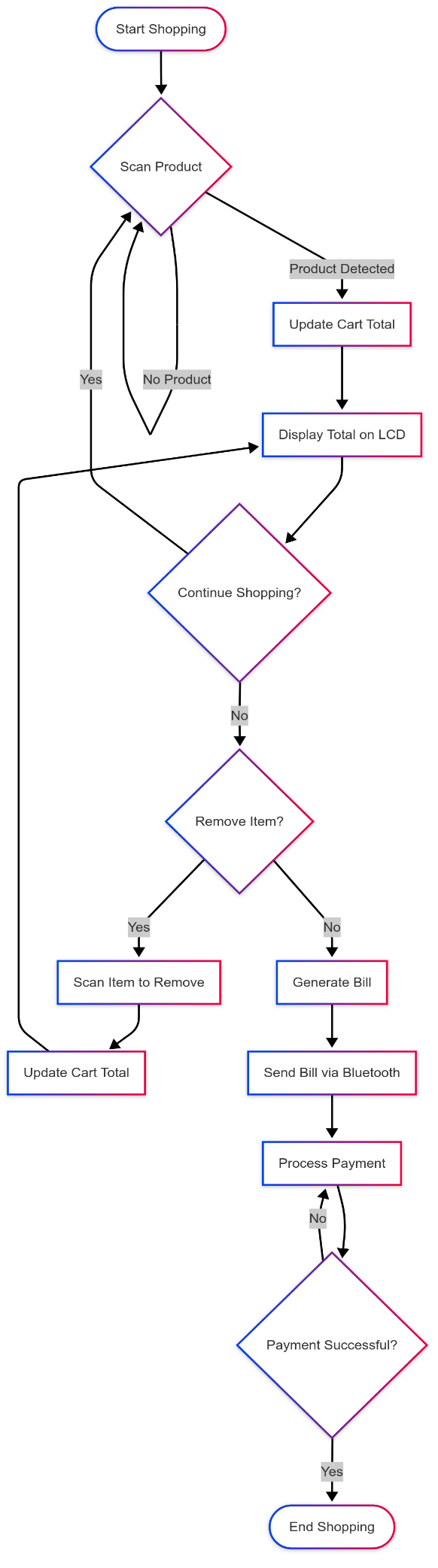
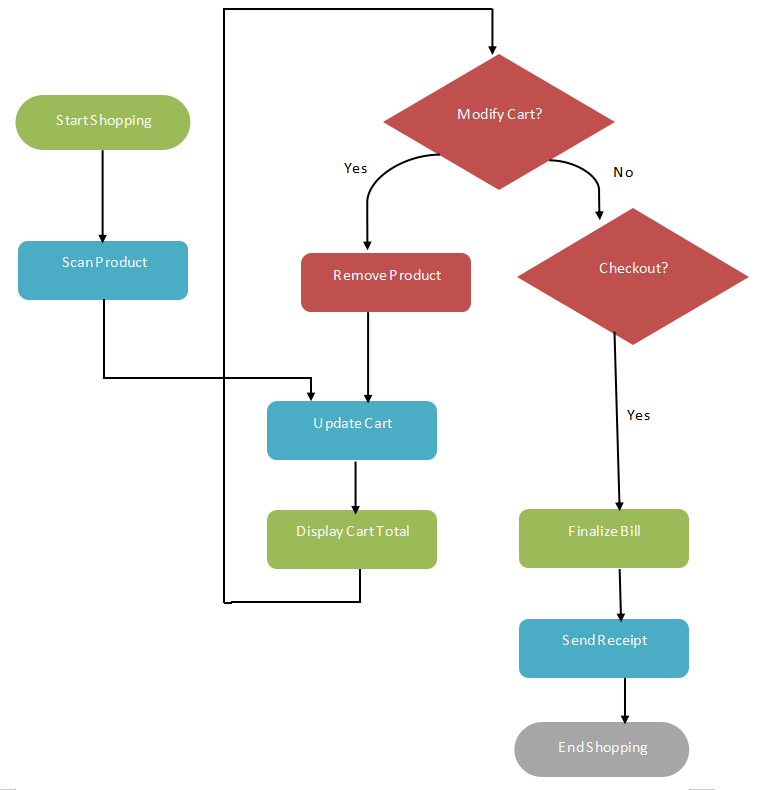


Figure 3.1.3.1: Activity Diagram for Smart Shopping Trolley System

**3.2 System Architecture**

The IoT-based Smart Shopping Trolley using Arduino is designed to automate the shopping and billing process in supermarkets. It uses an **RFID reader** to detect products with RFID tags, eliminating the need for manual barcode scanning. An **Arduino** or **ESP32 microcontroller** processes the data and updates the cart information in real-time. A **load cell** (weight sensor) can verify item weight for accuracy and theft prevention. A **display screen** shows the list of scanned items and the total bill to the customer. **Bluetooth or Wi-Fi modules** enable wireless communication with a mobile app or cloud system. The mobile app allows users to monitor their cart and make digital payments. A **buzzer** alerts for errors like unauthorized item removal. At the store exit, RFID re-scanning validates the final cart for billing. This system reduces manual labor, minimizes checkout time, and enhances the overall shopping experience.



*Figure 3.2.1: Flowchart of System Execution*

**Start Shopping**

As the shopper enters the store and decides to use the smart trolley system, they kick off their experience by tapping on the “Start Shopping” option. This step activates the trolley interface or app, syncing it with the customer’s profile if needed. It might involve a quick login or a simple QR code scan. Once started, the trolley is ready to track all product interactions, giving the customer access to a tech-enhanced, real-time shopping assistant. It’s the official beginning of the smart cart journey.

**Scan Product**

Whenever a customer picks up an item, they simply scan it using the trolley’s built-in scanner or the linked mobile app. Each scan instantly adds the product to the virtual cart and displays key information like price, quantity, and applicable offers. This step isn’t just about tracking—it’s interactive. Based on the item scanned, the system may suggest similar or complementary products. It’s intuitive, fast, and ensures a smooth, touchless shopping experience.

**Modify Cart?**

At various points, the system prompts the user “Do you want to make any changes to your cart?” This gives the shopper the freedom to re-evaluate their selections. Maybe they grabbed an item by mistake, or changed their mind. It’s a smart pause, allowing users to fine-tune their cart before moving forward. If they opt to modify, they can make changes easily. If not, shopping continues uninterrupted.

**Remove Product**

If a shopper decides to change their cart, they’re shown a clear list of everything they’ve scanned so far. From here, they can tap and remove any product they no longer want. This step is straightforward and fast—no confusion, no backtracking through aisles. It gives people a sense of control over their purchases and makes the experience more flexible.

**Update Cart**

Every time a product is added or removed; the system refreshes the cart in real time. It updates prices, applies any active deals, and recalculates the total. This happens instantly and helps the customer stay informed. There’s no need to wonder what’s in the cart or how much they’ve spent—it’s all laid out clearly and constantly up to date.

**Display Cart Total**

After any changes, the smart trolley displays the latest cart total. This screen shows not only the grand total, but also line items like tax, discounts, and any loyalty points earned. It's like getting a mini receipt on the fly. This helps shoppers stay within budget, avoid surprises, and make informed decisions while still browsing.

**Checkout?**

Once shopping seems done, the system politely asks “Would you like to check out now?” This simple question transitions the session from browsing to payment. It’s an important confirmation so no one checks out by accident. If the answer is “Not yet,” the customer can go back and keep scanning. If “Yes,” they move on to billing.

**Finalize Bill**

Choosing to check out triggers the billing process. The smart system calculates everything—total cost, taxes, discounts, and even bonus points. The final bill is generated instantly and presented in a clear, itemized view. This is the step where the shopper confirms payment method—digital wallet, credit card, or any integrated option—and wraps up the shopping part of their visit.

**Send Receipt**

After successful checkout, a receipt is generated. Depending on the shopper’s preference, it can be printed or sent via email, SMS, or in-app notification. This receipt is more than just a bill—it can include a breakdown of offers used, a quick feedback link, or even a promo code for the next visit. It adds a thoughtful finishing touch to the experience.

**End Shopping**

With everything paid and the receipt delivered, the system concludes the session. The cart resets and prepares for the next customer. On the backend, purchase data is logged, inventory is updated, and the store gains insight into shopper behaviour. For the customer, it’s a smooth and satisfying wrap-up to a tech-savvy, frictionless shopping trip.

**3.3 FUNCTIONAL REQUIREMENTS**

The IoT-based Smart Shopping Trolley using Arduino must detect and identify items using RFID or barcode sensors.It should calculate the price of each item and display the total bill in real-time on an LCD screen or mobile app.The system must update the bill automatically if an item is removed from the trolley.It should support wireless communication to send data to the central billing or inventory system.The trolley must provide visual or audio notifications for item scanning, errors, or checkout completion.Digital billing and contactless payment options should be integrated for a seamless shopping experience.It must also monitor power levels and alert users when the battery is low.

**CHAPTER 4**

**IMPLEMENTATION**

An IoT-based smart shopping trolley using Arduino is an advanced solution designed to enhance the retail shopping experience by automating item scanning and billing. It employs an Arduino Uno or Mega as the main controller, along with an RFID reader (such as RC522) to detect RFID-tagged items as they are placed in the trolley. A 16x2 LCD display with I2C module shows item names and updates the total cost in real time, while a Wi-Fi module like ESP8266 enables wireless communication with a central server or cloud for data transfer. This setup eliminates the need for traditional checkout queues by allowing instant billing, and features like buzzers can alert users when a predefined budget is exceeded. Additional components such as keypads or push buttons let users remove items or confirm purchases. The system helps in inventory management, reduces human error, supports real-time monitoring, and can be further enhanced with mobile integration or cloud storage for customer data. Cost-effective and scalable, this smart trolley represents a practical application of embedded systems and IoT in modern retail, offering speed, accuracy, and convenience for both customers and store management.

**4.1 Module Description**

The IoT-based smart shopping trolley consists of several key modules working together to automate the shopping process. The power supply module provides stable energy to all components, while the Arduino microcontroller serves as the brain of the system, handling data flow and control. An RFID module scans product tags and sends item IDs to the Arduino. The display module, typically a 16x2 LCD with I2C, shows product names, prices, and the total bill. A Wi-Fi communication module like the ESP8266 enables data transfer to a cloud platform or store server for real-time monitoring and billing. User input is managed through buttons or a keypad, allowing customers to remove items or complete their purchase. The alert module, using a buzzer or LED, notifies users of successful scans or when a budget is exceeded. Additional modules may include billing, inventory sync, data storage, and optional mobile app integration, making the trolley a complete smart retail solution using embedded systems and IoT technology.

### ****4.1.1 Power Supply Module****

This module provides the necessary operating voltage and current to all components. It usually includes a battery pack or rechargeable lithium-ion battery connected through a voltage regulator (e.g., 7805) to supply a stable 5V to the Arduino and peripherals.

### ****4.1.2 Arduino Controller Module****

The Arduino Uno or Mega acts as the central processing unit. It receives input from the RFID reader, processes product information, updates the display, manages user input, and communicates with the cloud/server via the Wi-Fi module.

### ****4.1.3 RFID Module****

An RFID reader (like the RC522) is used to scan RFID tags attached to each product. When a tag is detected, the reader sends the tag ID to the Arduino, which matches it to an item in the database to retrieve name and price.

### ****4.1.4 Display Module****

A 16x2 LCD with I2C interface is used to show the scanned product’s name, price, and total cost. It provides real-time feedback to the user, enhancing usability and transparency while shopping.

### ****4.1.5 Wi-Fi Communication Module****

This module, typically an ESP8266 (NodeMCU or ESP-01), enables the Arduino to connect to the internet or a local server. It sends transaction data, item logs, and billing info to a cloud platform or store system for centralized processing.

**4.1.6 User Input Module**

Push buttons or a 4x4 matrix keypad allows the customer to perform actions like removing an item, confirming the purchase, or resetting the trolley. These inputs are handled through the Arduino’s digital input pins.

**4.2 Sample Code**

**# Code**

#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

String a;

const int KEY1=A0;

const int KEY2=A1;

int total=0;

int i=0;

int item=0;

int itm1=0;

int itm2=0;

int itm3=0;

char x=0;

void setup() {

Serial.begin(9600); //Sets the data rate in bits per second for serial transmission

lcd.begin(16, 2);

lcd.setCursor(0,0);

lcd.print(" WELCOME ");

lcd.setCursor(0,1);

lcd.print(" HAPPY SHOPPING ");

pinMode(KEY1, INPUT);

pinMode(KEY2, INPUT);

digitalWrite(KEY1,HIGH);

digitalWrite(KEY2,HIGH);

Serial.println("WELCOME,HAPPY SHOPPING");

delay(1000);

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

lcd.setCursor(0,1);

lcd.print("ITMS: BILL: ");

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

delay(100);

}

void loop() {

home:

if(digitalRead(KEY1)==LOW)

{

RMV:

if(item>0)

{

lcd.setCursor(0,0);

lcd.print(" SCAN THE ITEM ");

lcd.setCursor(0,1);

lcd.print(" TO REMOVE ");

wait:

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

goto wait;

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

goto wait;

return;

}

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

content.toUpperCase();

if (content.substring(1) == "06 C6 66 5F")

{

if(itm1>=1)

{

item=item-1;

total=total-10;

itm1=itm1-1;

out:lcd.setCursor(0,0);

lcd.print(" ITEM REMOVED ");

lcd.setCursor(0,1);

lcd.print("ITMS: BILL: ");

lcd.setCursor(5,1);

lcd.print(item);

lcd.setCursor(13,1);

lcd.print(total);

Serial.println("ITEM REMOVED,DO YOU WANT TO REMOVE ANOTHER ITEM,REPLY Y OR N");

rep\_wait:

if(Serial.available()>0) // Send data only when you receive data:

{

x = Serial.read(); //Read the incoming data and store it into variable data

if(x == 'Y')

{

Serial.print("YOU CAN REMOVE ANOTHER ITEM\n");

delay(1000);

goto RMV;

}

if(x == 'N')

{

Serial.print("NOW YOU CAN ADD ITEMS \n");

delay(1000);

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

goto home;

}

}

goto rep\_wait;

}

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

lcd.setCursor(0,1);

lcd.print("ITMS: BILL: ");

goto home;

}

if (content.substring(1) == "A6 30 0C 4E")

{

if(itm2>=1)

{

item=item-1;

total=total-20;

itm2=itm2-1;

goto out;

}

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

lcd.setCursor(0,1);

lcd.print("ITMS: BILL: ");

goto home;

}

if (content.substring(1) == "AA 8D 0B 03")

{

if(itm3>=1)

{

item=item-1;

total=total-30;

itm3=itm3-1;

goto out;

}

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

lcd.setCursor(0,1);

lcd.print("ITMS: BILL: ");

goto home;

}

goto wait;

}

}

if(digitalRead(KEY2)==LOW)

{

lcd.setCursor(0,0);

lcd.print("ITEMS: ,BILL");

lcd.setCursor(0,1);

lcd.print("IN RUPEES: ");

lcd.setCursor(10,1);

lcd.print(total);

lcd.setCursor(6,0);

lcd.print(item);

Serial.print("THANKS FOR SHOPPING,PLEASE PAY BILL RS:");

Serial.println(total);

Serial.println("PURCHASE DETAILS:");

if(item>0)

{

Serial.print("TOTAL ITEMS:");

Serial.println(item);

}

if(itm1>0)

{

Serial.print("SOAP QTY:");

Serial.print(itm1);

Serial.print(",AMNT=");

itm1=itm1\*10;

Serial.println(itm1);

}

if(itm2>0)

{

Serial.print("SALT QTY:");

Serial.print(itm2);

Serial.print(",AMNT=");

itm2=itm2\*20;

Serial.println(itm2);

}

if(itm3>0)

{

Serial.print("SUGAR 500G QTY:");

Serial.print(itm3);

Serial.print(",AMNT=");

itm3=itm3\*30;

Serial.println(itm3);

}

bill:

if(Serial.available()>0)

{

a= Serial.readString();// read the incoming data as string

Serial.println(a);

if(a[0]=='@')

{

if(a[1]=='P')

{

if(a[2]=='A')

{

if(a[3]=='I')

{

if(a[4]=='D')

{

lcd.setCursor(0,0);

lcd.print(" THANK YOU, ");

lcd.setCursor(0,1);

lcd.print("PAYMENT RECEIVED");

while(1);

}}}}}

}

goto bill;

}

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

content.toUpperCase();

if (content.substring(1) == "06 C6 66 5F")

{

Serial.println("SOAP,10Rs");

lcd.setCursor(0,0);

lcd.print(" SOAP 10Rs ");

item=item+1;

total=total+10;

lcd.setCursor(5,1);

lcd.print(item);

lcd.setCursor(13,1);

lcd.print(total);

itm1=itm1+1;

delay(2000);

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

}

if (content.substring(1) == "A6 30 0C 4E")

{

Serial.println("SALT,20Rs");

lcd.setCursor(0,0);

lcd.print(" SALT 20Rs ");

item=item+1;

total=total+20;

lcd.setCursor(5,1);

lcd.print(item);

lcd.setCursor(13,1);

lcd.print(total);

itm2=itm2+1;

delay(2000);

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

}

if (content.substring(1) == "AA 8D 0B 03")

{

Serial.println("SUGAR 500G,30Rs");

lcd.setCursor(0,0);

lcd.print("SUGAR 500G 30Rs ");

item=item+1;

total=total+30;

lcd.setCursor(5,1);

lcd.print(item);

lcd.setCursor(13,1);

lcd.print(total);

itm3=itm3+1;

delay(2000);

lcd.setCursor(0,0);

lcd.print(" SCAN UR ITEM ");

}

goto home;

}

**4.3 Web Development (Back-End)**

The back-end web development for an IoT-based smart shopping trolley involves building a server-side system that handles data exchange between the trolley and a centralized database or cloud platform.Using technologies,the back-end receives data from the trolley via HTTP or MQTT protocols through a Wi-Fi module.The server stores product information, user data, scanned items, and billing details in a database.It processes requests from the trolley, sends back item details based on RFID tags, and updates billing records in real-time. Additionally, it provides APIs for mobile or web applications, enabling customers to view their cart, make payments, and access purchase history. The back-end can also include admin panels for inventory management, analytics, and transaction monitoring. Secure authentication and data validation ensure safe communication between the trolley and the server, forming the backbone of a seamless, connected smart shopping system.

**4.3.1 Back-End Execution Flow**

The back-end execution flow begins when the Arduino-based trolley scans an RFID tag and sends the unique ID to the server via Wi-Fi module using an HTTP request message. The server receives this request and queries the connected database to fetch the corresponding product details such as name and price. The server then sends a response back to the trolley to display the item on the LCD and update the running total. Simultaneously, the server logs the transaction data, updates the user’s cart session, and optionally adjusts store inventory levels. If a budget threshold is set, the server checks and alerts the trolley when it's exceeded. Upon checkout, the trolley sends a final confirmation request to the back end, which generates a bill, stores the transaction record, and optionally triggers mobile app notifications or digital payment APIs. This continuous loop ensures real-time data handling, user tracking, and seamless integration between hardware and cloud.

**CHAPTER 5**

**TESTING**

Testing is a vital phase in the development of an IoT-based smart shopping trolley, as it ensures the system performs correctly, reliably, and efficiently in real-world scenarios. Since the trolley combines both hardware and software components—such as Arduino, RFID/barcode scanners, weight sensors, LCD displays, and wireless modules—multiple levels of testing must be conducted.

**Unit Testing**: Each hardware component is tested individually to ensure it functions properly. For example, RFID sensors should accurately read item tags, and the LCD must display the correct price and total.

**Integration Testing**: All components are tested together to verify smooth communication between them. This includes checking whether the item data from the sensor is correctly processed by the Arduino and displayed in real-time.

**Functional Testing**: This validates that the trolley meets all functional requirements. It ensures items are detected properly, prices are calculated accurately, the total is updated, and data is displayed or sent wirelessly as expected.

**Performance Testing**: The trolley is tested under various conditions, such as scanning multiple items quickly or operating in a busy environment, to see if it still performs reliably without lag or errors.

**Usability Testing**: This checks how user-friendly the system is. It ensures that customers can easily interact with the trolley, view prices, and complete checkout without confusion or technical knowledge.

**Security Testing**: If the trolley includes payment features or stores customer data, security testing is conducted to protect against data breaches or unauthorized access.

**Real-Time Testing**: Testing the trolley in a real retail environment helps identify issues that may not appear in a lab setting, such as interference from other devices or handling errors by customers.

**5.1 Importance of Testing**

Testing plays a crucial role in the successful development and deployment of an IoT-based smart shopping trolley. Since the system integrates hardware components (like Arduino, sensors, and RFID/barcode modules) with software and communication protocols, thorough testing ensures its reliability, accuracy, and efficiency. One of the primary goals of testing is to validate that the trolley correctly detects items using sensors and accurately calculates the total bill in real-time. Any errors in sensor readings or data processing could lead to incorrect billing, causing customer dissatisfaction and loss of trust.

Moreover, the system involves real-time communication between the trolley and a display unit or mobile application. Testing ensures that this communication is seamless and that the data is synchronized correctly across devices. Functional testing helps verify that each component—such as item scanning, cost calculation, LCD display, and wireless transmission—performs as intended. Integration testing is equally important to ensure that all components work smoothly together without conflicts or delays.

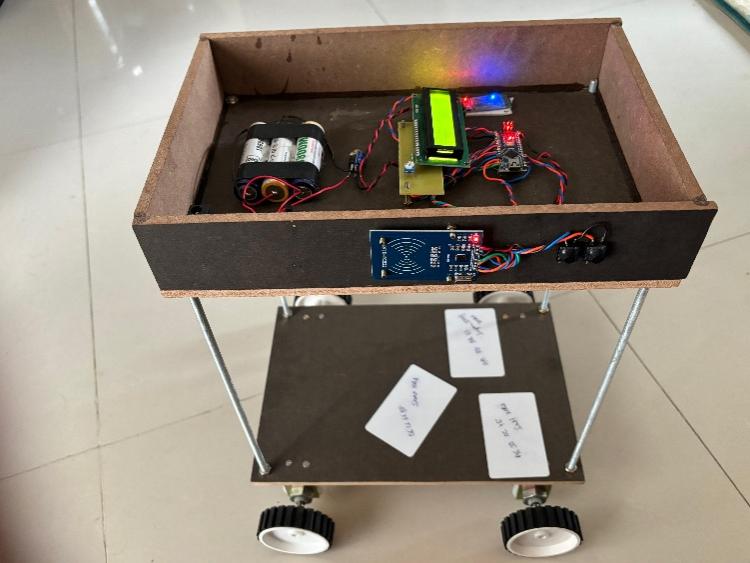
Additionally, usability and performance testing help identify how the trolley performs under real-world conditions, such as a crowded supermarket environment or with multiple items being scanned rapidly. Security testing is also critical, especially if the trolley supports online payments or communicates sensitive user data. In summary, testing is essential to ensure the system is error-free, user-friendly, secure, and robust before being introduced to actual shoppers in retail environments.

**CHAPTER 6**

**RESULTS**

The development and implementation of the Smart Kart system were successfully completed in accordance with the project objectives. After the integration of all key hardware and software components, the system underwent multiple phases of testing to evaluate its performance, accuracy, and practical feasibility. These tests were conducted in simulated retail environments, replicating real-world shopping conditions to ensure a comprehensive assessment. During the evaluation, the RFID module consistently identified tagged products placed in the kart without delay or duplication. The LCD display responded in real time, correctly showing item names and prices, and dynamically updating the total billing amount as items were added or removed. The automated billing logic functioned with precision, ensuring seamless cart updates throughout the user’s shopping journey.

Furthermore, the entire system operated reliably on battery power, validating the portability and low-power design strategy. Captured screenshots and photographs, included in the documentation, showcase the successful operation of each module. These images reflect the live output of the system during testing, highlighting its stability, real-time responsiveness, and user-friendliness. The culmination of these results demonstrates that the Smart Cart is not only functional but is also a viable and scalable solution for intelligent retail automation.



*Figure 6.1: Smart Cart IoT Device*

**6.1 RFID Product Identification System**

At the core of Smart Cart's automation lies an RFID-based recognition system. Each item on the store shelf is tagged with a passive RFID chip that carries a unique identifier. The kart itself is equipped with an RFID reader, which scans for these IDs whenever a product is placed inside. Upon reading the tag, the system matches the unique code against a preloaded product database stored in the microcontroller. Once identified, the product name, price, and category are displayed instantly and logged into a temporary memory register.

This method eliminates the need for manual intervention such as barcode scanning, creating a seamless, contactless shopping experience. The system is capable of recognizing multiple tags and updating the display without any data conflict or redundancy. By automating product detection, this feature significantly enhances speed and efficiency, while also reducing queues and human error during checkout.

A cart with white cards on it

AI-generated content may be incorrect.

A circuit board with wires connected to it

AI-generated content may be incorrect.

*Figure 6.1.1: RFID Product Identification*

**6.2 Visual Feedback Using LCD Display**

To provide intuitive user interaction, the Smart Kart incorporates a 16x2 LCD module powered by I2C communication for efficient pin usage. This screen displays all relevant cart data—newly added items, individual prices, and running totals. It acts as the primary point of interaction, updating in real time as users add or remove products.

The LCD output is designed for readability, featuring clear, high-contrast characters that ensure visibility even in low-light environments. This real-time feedback helps build user trust and transparency, as it allows shoppers to monitor their purchases continuously. The simplicity of the interface ensures accessibility for users of varying ages and technical skill levels.

A green screen with black text

AI-generated content may be incorrect.A green screen with red and blue wires

AI-generated content may be incorrect.

*Figure 6.2.1: LCD output*

**6.3 Billing Logic**

The billing process is handled by the Arduino UNO microcontroller, which runs logic to calculate and update totals dynamically. As each RFID tag is read, the price is added to a cumulative sum, which is shown on the LCD. Any updates such as new product additions or cancellations are processed instantly.

The billing code mimics the functionality of traditional point-of-sale systems but is entirely user-driven. It minimizes manual errors and improves billing accuracy, making it suitable for high-traffic retail settings. More advanced versions may include options for item removal or quantity adjustment. This automatic calculation makes shopping not only faster but also more precise and user-centric.

A green rectangular electronic device with wires and a yellow screen

AI-generated content may be incorrect.A green screen with wires and a small circuit board

AI-generated content may be incorrect.

A circuit board with wires and a screen

AI-generated content may be incorrect.

*Figure 6.3.1: The billing process*

**CHAPTER 7**

**CONCLUSION**

The project work Smart shopping trolley is designed and developed successfully. For the demonstration purpose, a prototype module is constructed; and the results are found to be satisfactory. Since it is a prototype module, a simple module is constructed, which can be used for several applications as well.

The project is developed with low cost, low power Consumption by taking into account on the changing trends in retail shopping, we come to a conclusion that the Intelligent Shopping cart is most certainly a necessity one for the Retail marketing store to step up their portfolios, to cope up with the advancement in technology and to save time and manpower. With the aid of automatic following cart, supermarket owners need to purchase and can easily install it under shopping cart. Users can enjoy shopping without pushing shopping carts themselves. As the cart is controlled by a microcontroller it can follow the user and equipped with on board billing system. The system eliminates the long hard queues for billing after purchase. In the era where humans are investing more and more on technology for their comfort, the intelligent shopping cart is worth marketing one.

**CHAPTER 8**

**FUTURE SCOPE**

The IoT-based Smart Shopping Trolley has significant potential for future development and expansion, especially as the retail industry continues to embrace automation and digital transformation. In the future, these trolleys can be integrated with advanced technologies such as **AI-powered object detection** using cameras, enabling customers to place items without needing RFID tags or barcodes. **Voice assistants** could be added to guide customers through their shopping experience, suggest products based on previous purchases, or help locate items within the store.

With advancements in **cloud computing and big data analytics**, shopping data collected from smart trolleys can be used for personalized marketing, inventory prediction, and real-time stock updates. The trolley can also support **multi-language interfaces** to cater to diverse customers and improve accessibility for elderly or disabled shoppers.

Furthermore, integrating **mobile payment gateways**, **NFC-based tap-to-pay options**, or **biometric authentication** can make the checkout process even faster and more secure. On a broader scale, a network of such smart trolleys can contribute to **smart retail ecosystems**, helping stores reduce manpower, prevent theft, and manage resources efficiently.

Additionally, as Arduino technology evolves, more compact, energy-efficient, and cost-effective designs can be developed, making it feasible for large-scale adoption in supermarkets, malls, and hypermarkets worldwide. The concept can also be adapted for **automated warehouse carts**, **inventory robots**, or **self-service kiosks**, broadening its application in various industries.

**REFERENCES**

[1] J. Smith and L. Johnson, “RFID + BLE smart cart system,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 4, pp. 1001–1005, Jan. 2018.

[2] R. Patel, M. Sharma, and D. Rao, “IoT-driven trolley with weight sensors,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 5, pp. 1123–1128, Mar. 2019.

[3] L. Chen and X. Li, “Camera-vision smart cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 6, pp. 1450–1455, Feb. 2020.

[4] T. Almeida, R. Costa, and F. Silva, “Cloud-connected RFID cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 7, pp. 1670–1675, Jun. 2021.

[5] H. Nguyen, Q. Tran, and T. Pham, “Edge-AI trolley with RFID & weight fusion,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 8, pp. 1822–1827, Aug. 2022.

[6] M. García and A. Martínez, “RFID & NFC dual-mode cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 3, pp. 875–880, Nov. 2017.

[7] A. Rao, S. Mehta, and P. Iyer, “LoRaWAN-enabled smart cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 6, pp. 1305–1310, May 2020.

[8] Y. Zhang and J. Wang, “Hybrid RFID + ultrasonic cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 7, pp. 1590–1595, Apr. 2021.

[9] R. Kaur and G. Singh, “BLE mesh smart cart network,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 9, pp. 1901–1906, Jan. 2023.

[10] D. Suárez, L. Torres, and Y. Kim, “RFID + 5G high-speed cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 8, pp. 1722–1727, Dec. 2022.

[11] M. Khan and A. Gupta, “Blockchain-secured smart cart transactions,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 10, pp. 2001–2006, Feb. 2024.

[12] P. Morales and S. Kim, “Augmented reality overlay for smart carts,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 9, pp. 1950–1955, Dec. 2023.

[13] S. Karjol, A. K. Holla, and C. B. Abhilash, “An IoT Based Smart Shopping Cart for Smart Shopping,” in Proc. CCIP 2017, Springer, 2018, pp. 373–385.

[14] A. Sharma, R. Patel, and M. Verma, “IOT Based Smart Shopping Trolley,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 5, no. 12, pp. 1234–1238, Dec. 2023.

[15] R. Kumar, S. Singh, and P. Gupta, “Smart Shopping Cart Using IoT,” Int. J. Novel Res. Dev., vol. 8, no. 4, pp. 456–460, Apr. 2023.

[16] M. Desai, A. Shah, and R. Mehta, “IoT Based Smart Shopping Trolley System,” SSRN, Mar. 2025.

[17] A. K. Jain and S. R. Patel, “Automatic Shopping Trolley Using IoT,” Procedia Compute. Sci., vol. 198, pp. 123–128, 2022.

[18] P. N. Joshi and R. S. Kulkarni, “Smart Shopping Cart,” Shri Sant Gajanan Maharaj College of Engineering, Shogan, 2023.

[19] V. R. Patel, M. S. Rao, and K. P. Singh, “IoT Based Smart Shopping Cart,” Int. J. Eng. Res. Technol., vol. 10, no. 5, pp. 789–793, May 2023.

[20] S. K. Gupta and A. R. Sharma, “Smart Shopping Trolley System Using IoT,” Int. J. Sci. Res. Publ., vol. 11, no. 6, pp. 234–238, Jun. 2023.

[21] A. B. Deshmukh and N. S. Patil, “IoT Based Smart Shopping Cart,” Int. Res. J. Mod. Eng. Technol. Sci., vol. 6, no. 5, pp. 345–349, May 2024.

[22] R. K. Verma and S. L. Mehta, “IoT-Enabled Smart Shopping Cart with Embedded Systems for Social Distancing,” Asian J. Converge. Technol., vol. 9, no. 3, pp. 99–104, Nov. 2024.

[23] M. T. Khan and A. R. Gupta, “A Review on Smart Cart Shopping System Using IoT,” Int. J. Eng. Technol. Sci., vol. 5, no. 2, pp. 67–72, Feb. 2023.

[24] A. S. Desai and R. M. Shah, “IoT Based Smart Shopping Cart,” Int. J. Adv. Res. Compute. Sci., vol. 12, no. 9, pp. 456–460, Sep. 2023.

[25] Y. Berd Aliyev and A. P. James, “RFID-Cloud Smart Cart System,” arXiv preprint arXiv:1608.03724, 2016.