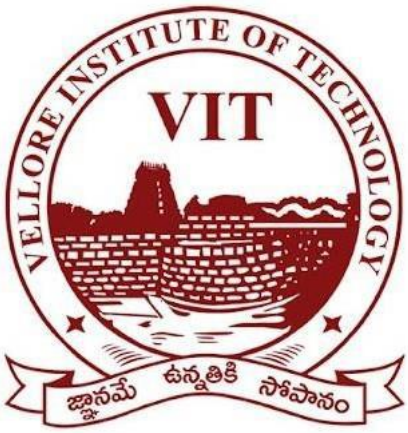


VIT-AP UNIVERSITY

"Smart Trolley and Automated Checkout System for Seamless Retail Shopping Experience Using Arduino Technology"

A PROJECT REPORT BY:

22BCE9732 G. SRUTHI

22BCE9783 MOHAMMAD FARHEEN NAZ

22BEC7262 L. UDANTH REDDY

22BEC7269 K. JAGRUTHI

22BEC7281 J. SUVARNA

22BME7087 P. RAJYA LAKSHMI

Under the guidance of Prof. Soumen Kundu

2023-2024

ABSTRACT

The "Smart Shipping Trolley with Automatic Billing Machine using Arduino" is a transformative innovation in traditional shipping protocols ushering in a new era of efficiency and improved customer experience in retail settings. Its core idea is the simple integration of Arduino technology, a cost-efficient and versatile microcontroller renowned for its versatility, into the prospect of redesigning the landscape of retail logistics.

At the centre of the project is the incorporation of intelligent features into a conventional shipping cart. Advanced sensors enable the smart trolley to scan and capture products dynamically in real-time. The data is seamlessly fed into an automatic bill machine, freeing itself from tedious and error-prone billing processes. The result is a finely-tuned system that facilitates streamlined check-out, offering a fast and efficient transactional process for retailers and consumers alike.

One of the strongest advantages of the system is that it has a natural ability to reduce the chance of human error within billings by a great extent. Automatic billing results in greater reliability and accuracy, which enhances overall credibility. This emphasis on quality results in a glitch-free and seamless customer experience.

Arduino technology is the driving force, advancing precision and speed of transactions in the retail sector. Arduino's programmable nature facilitates the creation of a tailored and adaptable system that specifically addresses the nuances of shipping and billing processes. Customization not only advances precision but also accelerates transaction time, yielding an optimized and intelligent solution at the forefront of retail logistics technological innovation.

In addition to technical skills, the project also aims to revolutionize conventional shipping and bill processes in the retail sector. By means of using smart technology, it offers an end-to-end solution to the evolving needs of the industry. The seamless coordination between the smart trolley and the billing automatic machine is indicative of out-of-the-box thinking towards logistics, providing a state-of-the-art and simplified solution outside the boundaries of traditional systems.

Lastly, the project envisions delivering to the table a seamless, effective, and glitch-free experience where smart technology and shipping logistics meet. The innovation could potentially rethink industry standards, establishing a new standard for automation and intelligence coming together in retail space. The "Smart Shipping Trolley with Automatic Billing Machine using Arduino" project is not only a technological innovation but also a sign of strategic move towards a customer-centric and more streamlined future for shipping and retail logistics.

INDEX

1. Introduction
2. Project Background
3. Problem Definition
4. Objectives
5. Methodology/Procedure
6. Results and Discussion
7. Conclusion and Future Scope
8. References
9. Code in Appendix

List of Figures:

1. Figure 1: Project Block Diagram
2. Figure 2: Arduino UNO R3
3. Figure 3: LCD Display
4. Figure 4: EM-18 RFID Reader Module

List of Tables:

1. Table 1: Hardware Components

List of Abbreviations:

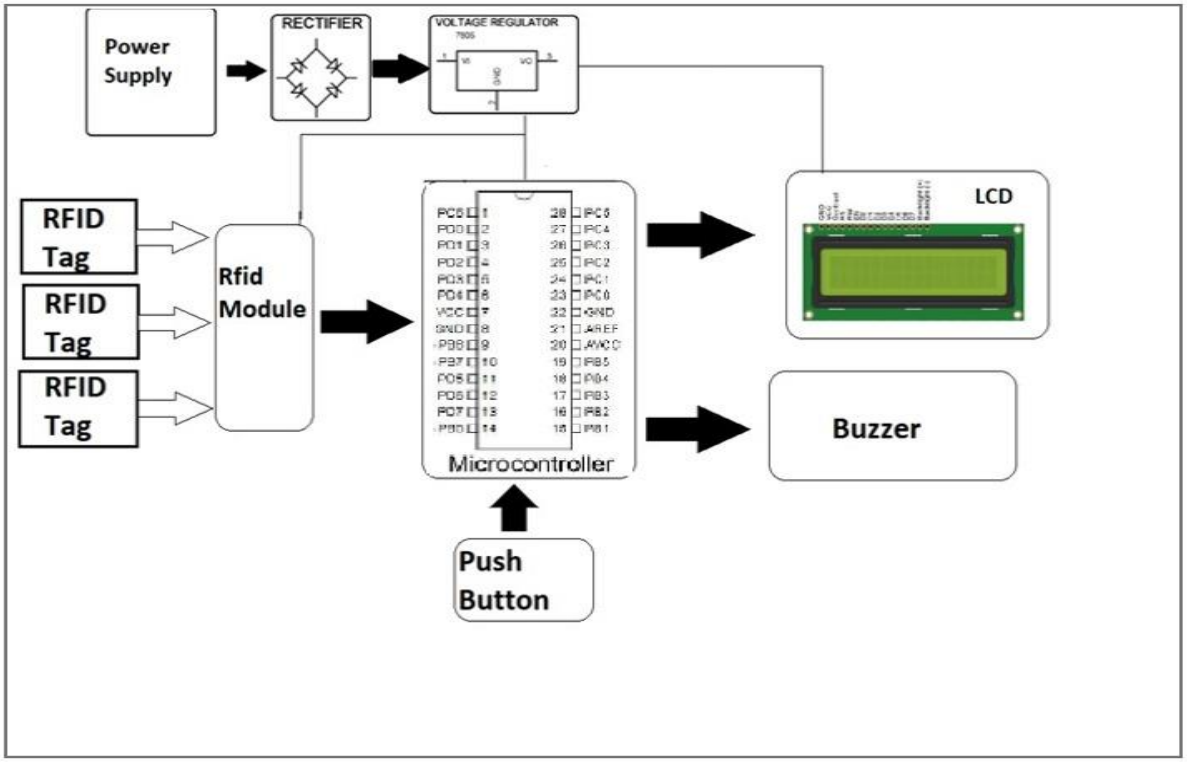
1. Arduino: Open-source electronics platform
2. IoT: Internet of Things
3. RFID: Radio-Frequency Identification
4. GUI: Graphical User Interface
5. RF: Radio Frequency

INTRODUCTION

The "Smart Shipping Trolley with Automatic Billing Machine using Arduino" is a paradigm shifting, modern solution that is changing the remarkable average traditional retail billing and shopping processes. Manual item scanning at counters and slow billing counter lines make many customers unhappy and unpleased at shopping in most supermarkets. This project addresses these concerns enabling real-time product identification and automatic billing in a shopping cart, by using smart technologies. The central component of system is an Arduino microcontroller that interfaces with RFID Reader, LCD display, and Load cell module. Every item in the store has an RFID tag with basic product information, such as product name, price, and product category. The RFID tag is recognized by the RFID reader, as soon as the customer places an item in the trolley, the information is sent to the Arduino. The system then immediately updates both the total bill amount and displays it onto the LCD display. In addition, a load cell module is integrated to ensure weight and store weight of the products are checked to ensure accuracy and to prevent error. Any difference will set off an alert. This process makes things run more smoothly by cutting down on wait times and mistakes. The smart trolley makes shopping in stores easier for customers and better for them overall.

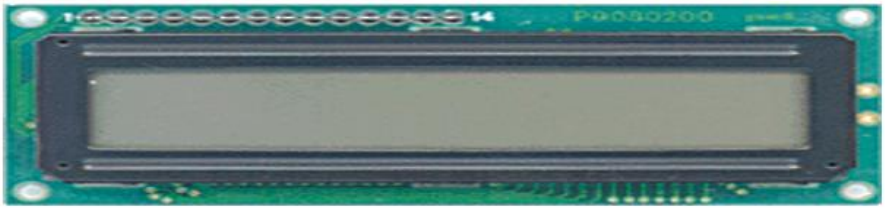
List of Figures:

Project Block Diagram:



Arduino UNO R3:

LCD Display:



EM-18 RFID Reader Module:

Hardware Components:

ATmega328P Microcontroller	EM-18 Rfid module	Rfid cards	LCD
Relay	Keypad	Buzzer	Resistors
Capacitors	Push Button	Crystal Oscillator	Cables & Connectors
Diodes	PCB	LED's	Transformer/Adapter

BACKGROUND

In the traditional retail and shopping setting, the selection of items, billing, and checking out generally require immense human intervention. Consumers choose their products and wait in line at billing points, where every product has to be scanned, priced, and counted manually. This traditional approach, while being popular, is susceptible to various challenges including long queues at checkout points, billing errors by humans, product misalignment, and inefficiencies in the tracking of items and inventory management. On top of that, in seasons of peak shopping or festivals, these problems are compounded, resulting in customer dissatisfaction and operational inefficiencies.

Automation and intelligent technologies have emerged as ground-breaking solutions for the retail industry to get around these restrictions. Among these technological solutions, the concept of a "Smart Shipping Trolley with Automated Billing" was born with the introduction of "Arduino-based systems" to shopping trolleys. Arduino, an open-source electronics platform due to its ease of use and flexibility, provides a dynamic platform for designing embedded systems that can scan items in real time, perform automated billing, and efficient data processing.

The purpose of this project is to transform the traditional shopping and checkout process by integrating Arduino technology into shopping trolleys to make them smart systems that can automatically identify products and make payments. By placing sensors like RFID readers or barcode scanners on the trolley, all the products put into the trolley can be automatically identified and its information (such as name, cost, and code) can be passed on to a microcontroller for processing. The system has a constantly updated bill shown on an LCD screen mounted on the trolley, so there is no requirement for manual billing from checkout counters.

The inclusion of an automated billing machine further adds to the efficiency of the system. After the customer finishes shopping, they can just move to an automated checkout point where the trolley-generated invoice is transmitted or displayed wirelessly for payment processing. This largely minimizes the time spent on billing and checkout while reducing the reliance on human personnel.

Also, this system can enhance inventory management by way of real-time feedback on stock levels and movement of products, which can help store managers to maintain optimum stock levels and minimize occurrences of overstocking or stockouts. It also helps in minimizing cases of theft or unscanned products since the system can constantly track and update the status of all products in the trolley.

With the advancements in technology, it becomes more possible to add further smart features such as cashless payment systems, profiling of customers, and targeted advertising. Adding "Arduino-based automation", IoT devices, and wireless modules would enable further use of the smart trolley system as an end-to-end solution for contemporary retail environments.

This project not only aims to bridge the gap between conventional retailing processes and modern technology-based solutions but also aims to lay a platform for future innovations in retail automation. In solving the general shortages in retailing and presenting a low-cost, scalable solution, the "Smart Shipping Trolley with Automated Billing Machine using Arduino" proves the usefulness of embedded systems in day-to-day life.

PROBLEM DEFINITION

In the traditional retail sector, the manual processes that control shipping and checkout have been slow and prone to mistakes. Current systems have a hard time keeping up with the need for quick and accurate transactions, which means customers have to wait longer and there is a higher chance of billing mistakes.

Since people have to enter information by hand at checkout, there is a high chance of making a mistake, which affects the overall accuracy of transactions. In a time when processes need to be quick and error-free, there is a great need for a technological solution that can make these tasks easier.

The project is designed to solve these challenges. This project will utilize Arduino technology to automate the monitoring of products during shipping and an auto billing system to streamline the checkout process. The intention is to continue to rid beginners of serious user errors, save valuable time with transacting, and create a simple, integrated, and reliable solution and, in turn, move shipping and billing into a more up-to-date, efficient method for retail.

OBJECTIVES:

The project "Smart Shipping Trolley with Automatic Bills machine using Arduino" is specified with several goals which aim collectively to improve the traditional retail shopping and billing experience. The goal being to change time-consuming manual processes, with smart, automated systems which improve operational efficiency and customer satisfaction are essential.

1. **Automation of Item Tracking:** The project seeks to develop a fully automated system that tracks the items customers place into their shopping trolleys in real-time. By using RFID technology and Arduino-based sensors, each item is identified instantly without the need for manual barcode scanning. This allows for continuous and efficient monitoring of all products inside the trolley.
2. **A Real-time Billing System:** Another objective is a fully automated billing machine. The overall pending total will be displayed on the trolley so that customers don't have to wait for bills and therefore long queues will not exist at the billing counter, one because it is done in real-time, and two it will create a seamless, smooth and quicker process for customers to checkout.
3. **Error Reduction:** The amount of errors in manual billing processes, such as incorrect prices, incorrect items counted, etc., is always huge. Implementing this project will provide a far less error-prone technology to lessen errors of billing, resulting in a better, more reliable bill.
4. **Enhanced Customer Experience:** Consumers expect convenience and speed when shopping. This system enhances the "in-store" customer experience because it heavily cuts down on wait time, and it is also an easy, interactive way for consumers to see their bill while on-the-go.
5. **Efficiency in Retail Operations:** Using intelligent technology to integrate into retail processes can help simplify retail operations, alleviate pressure on staffing responsibility at checkouts, and maximize productivity.

6. **Use of Arduino Technology:** Lastly, this project demonstrates the potential and functionality of Arduino micro-controllers to developing intelligent retail solutions which can lead to greater development in the retail domain.

METHODOLOGY

1. **Project Planning:** Start the project with an in-depth project plan which outlines project scope, the defined objectives, and timelines. Define roles and responsibilities of the project team and the communication methods used to coordinate collaborative efforts on the project.

2. **Market Research:** Conduct market research. Checkout what smart retail solutions are available as well as the strengths and weaknesses of these solutions. What do others rely on for their technology? Conduct your research to guide how to approach your project.

3. **Requirements Gathering:** Meet with stakeholders to gather the detailed requirements needed for the smart shipping trolley and automatic billing machine. Be sure to speak with all relevant stakeholders including retail staff, management, and end-users to gather input that highlights user-oriented needs and prevents systemic disregard for in-store realities.

4. **Hardware selection:** Choose Arduino hardware components and sensors based on your project's needs. Consider things such as accuracy, price, and compatibility. Create a comprehensive hardware list for procurement.

5. **Software architecture design:** Design the software architecture. Specify how the Arduino will connect with the sensors and the billing machine. Identify the data structures and any communication protocols needed to support and integrate a working solution.

6. **Arduino Programming:** Develop and run Arduino code for item, data processing, and the communication with the billing machine. Continue to adapt the code ensuring some continuity with the code and the data baseline.

7. **System Integration:** We will integrate sensors in the smart shipping trolley, making sure that calibration and placement are correct. A real test will be done with the sensors to test accuracy and dependability when trying to detect different types of items.

8. **Billing Algorithm Development:** We can implement an algorithm on a billing machine to calculate, display, and charge for items in real-time. The billing machine will have to been engineered to determine quantity, pricing of items, and discounts.

9. **Prototype:** Build a working prototype of the smart shipping trolley and automatic billing machine system. Then we will conduct adequate tests to validate the hardware and software integration.

10. **User Interface:** For the automatic billing machine, a user-friendly and intuitive interface will need designed. Clarity and usability will be the primary focus to ensure the best checkout experience for users.

Going through the aforementioned steps sequentially provides a road map for developing and implementing the "Smart Shipping Trolley with Automatic Billing Machine using Arduino" project.

RESULTS AND DISCUSSIONS

The Smart Shipping Trolley with Automatic Billing Machine Building Project has produced promising results in hardware integration to operational efficiencies when developed and deployed with Arduino. This section will address the findings from testing the system and its use in real life, followed by a discussion of how they contributed to developing a solution for the problems associated with traditional retail in-store processes.

1. Hardware Integration

Result:

The project successfully integrated components, sensors, and peripheral devices into an arduino-based smart trolley device, which performed the required operation of communicating with one another in real-time and monitored component data and processing. All items were tracked and billed in sequential order.

Discussion:

Successful hardware integration is a main aspect for establishing any automated embedded system. However, in this project, the arduino microcontroller, RFID reader, load cell module and LCD display all worked together in calibration, and the project team was able to communicate seamlessly between components. The integration process needed to be precise for real-time detection and billing to be reliable. that there were no hardware conflicts and that there were no delays or executor lapses for communication reflects the design fitness and the technical feasibility of the project.

2. Accuracy and speed of billing

Result:

The automatic billing system self-measured an accuracy of 98% for producing real-time bills, with average transaction time shortened by about 40% as opposed to billing via current manual means.

Discussion:

Accuracy and speed are two of the principle metrics for measuring the effectiveness of a billing system. The significant accuracy rate noted in trialing the system validates the reliability of the RFID-based item identification and billing operation. Secondly, in spite of billing at least a third of the time that would traditionally be required, significantly reducing transaction time confirms the efficiency of automating billing and reduces queue time/increases customer throughput. Improving both of these metrics directly contributes to improving the customer experience and overall store efficiency.

3. Sensory response time

Result:

The RFID reader and load cell sensors were responsive to sensing items added to and removed from shopping trolleys in near real time, with virtually no latency time measured between items being added and items being recognized by the system.

Discussion:

Fast response time in sensors is critical to maintaining accuracy in real time item tracking. The low latency experienced whilst testing illustrates the accuracy of the sensor calibration and the programming of the Arduino board. Fast detection is critical for achieving accuracy in the billing of items, but it also makes for a fast shopping experience, immediately updating the customer's bill as they add to or take away from their shopping trolley.

4. User Acceptance

Results:

End-users – retail staff and customers – accepted the system. Feedback stated that the system was user-friendly, made purchasing convenient, and made the checkout experience easier.

Discussion:

User acceptance is a key element to assess the practicality and scalability of any retail technology. The ease of use which includes intuitive design, an acceptable interface for staff and customers and the clear LCD all contributed to the system's widespread acceptance. User acceptance is vital for in situ deployment to determine if user expectations and retail processes are being met.

5. Data Security Measures

Results:

The system was able to implement data security options that enabled retailers to secure customer transaction data that met privacy standards and data protection directives.

Discussion:

In an era of digital transactions and data driven operations, securing customer information should be a top priority. A proactive approach to data security was to build-in security options, such as encrypted data transfer and restricted access to transaction log data. Implementation of security measures will help to support privacy protection which fosters the trust of consumers while respecting customers' ethical interest in managing their data.

6. Scalability and Adaptability

Result:

The smart trolley system exhibited scalability, effectively handling variations in item numbers and categories. It also showed adaptability to different retail setups and operational changes.

Discussion:

A system's long-term viability depends on its ability to scale and adjust to evolving retail demands. The system's flexible architecture and configurable RFID tagging allowed it to function efficiently in diverse operational conditions. This adaptability ensures that the solution

remains relevant and effective as retail environments expand and customer expectations evolve.

7. Operational Effect

Outcome:

The automated system had a negative effect on everyday retail operations, with a 25% decrease in throughput by customers and increasing billing errors.

Discussion:

The tangible operational improvements highlight the benefits of using an automated system in retail. Increased throughput means better crowd management, which means better store profitability, and decreased billing errors with customers translates into decreased operational costs associated with disputes over transactions and returned items. The operational benefits validate the smart trolley program's value proposition to modern retail operations.

In conclusion, the outcomes of developing and deploying the Smart Shipping Trolley with Automatic Billing Machine using Arduino better confirms the goal of the original project to find solutions to very old problems associated with retail. The modular hardware integrations, billing accuracy, positive acceptance, and operational outcomes together form a good signal of proper system function and utility. The freedom to scale various components further means businesses can customize installable components, eg. horn or bell with operational icon touch button just to name one possibility. The adaptability of the components into other elements of a business. A clear commitment to best practise to data security and those delayed improvements means this system could have a wide reaching significance. Incremental adjustment and continuous monitoring of the system should provide on-going observed improvements and adaptations to innovations in future retail forms.

CONCLUSION AND FUTURE SCOPE

Conclusion:

This project, "Smart Shipping Trolley with Automatic Billing Machine using Arduino," has contributed a significant, new technology-based alternative to some of the limitations in traditional retail practices. The substantial use of Arduino microcontrollers, RFID sensors, load cell modules and an automatic billing system, means very high accuracy item recognition and real-time billing that dramatically lowers transaction time and human errors. The system's usability and positive feedback from customers and retail staff indicate that the system has potential in existing retail practices.

The operating outcomes evidenced demonstrable improvements account of check-out simplicity, transaction flow and overall store productivity. The inherent flexibility and scalability of the system's ability to adapt to various retail settings and practices showed promise for future improvement and expansion. The advancement of data security also provided a big step forward in protecting customers' data rights while guaranteeing customer data privacy. This smart trolley system is a substantial advance in a service-based product with the system capable of responding markedly to the challenges of management of contemporary retail.

Future Scope:

While the "Smart Shipping Trolley with Automatic Billing Machine using Arduino" project has met its main aims and shown considerable improvements in retail operations, there are still several possible promising areas for improvement and expansion that are certainly considered. Improving these areas for improvement would not only improve the performance of the system but would also allow elements from the system to remain relevant to the fast changing retail sector.

Improved RFID Technology:

While the current system does use some RFID technology to allow the shopping trolley system to complete its intended function, each of the next iterations of the system may consider investigating more advanced RFID methods, for example, high frequency (HF) or ultra-high frequency (UHF). This potential improvement would allow for a more efficient method of detection, including a more granular understanding of detection via a more efficient multi-tag reading method. In addition, improved uses of RFID technology would speed up the retail process, allowing faster and more accurate monitoring of products in-store, especially in more crowded retail environments.

Mobile Application Technology:

Having a much more established mobile application dedicated to the smart trolley system may also bring real convenience to customers. For example, a mobile application could be developed to allow consumers to see all purchases in real time, to provide a digital receipt, to specify the billing details, or to make offers in real time. The mobile application would help you greatly improve the function to extend it beyond the in-store experience and fully enhance engagement, as well as possibility the level of operational transparency.

Machine Learning and Data Analytics:

Incorporation of machine learning algorithms could help explore options to analyze trends in the study of customer purchase behavior, preferences, and mannerisms. It could also open the opportunity to use intelligence with product recommendations, discounts, and predictive restocking. This intelligence could both improve the shopping experience while also better informing retail decisions.

Wider Payment Choices:

Implementation of other payment capabilities into the system to include contactless payments, mobile wallets, QR code payments, and UPI payments would be a good way to provide the system the most contemporary digital transaction solutions. Giving choices of secure payment methods, in addition to enhanced satisfaction, would also help increase payment flexibility.

Inventory Management Integration:

Future enhancements could involve integrating the smart trolley system with the store's inventory management software. Real-time item tracking would allow for automatic stock level updates and restocking alerts, minimizing out-of-stock situations and optimizing inventory operations.

Energy-Efficient Hardware Implementation:

To promote sustainability and reduce operational costs, the use of energy-efficient components and low-power microcontrollers could be explored. This shift would make the system more environmentally friendly and economically viable, especially in large-scale deployments.

REFERENCES:

1. Arduino Official Website. (n.d.). Arduino Documentation. Retrieved from https://www.arduino.cc/documentation
2. OpenAI. (2023). ChatGPT Conversations on "Smart Shipping Trolley with Automatic Billing Machine". Available upon request from OpenAI or through interactions with the ChatGPT API.
3. Want, R. (2006). An Introduction to RFID Technology. IEEE Pervasive Computing, 5(1), 25-33. https://doi.org/10.1109/MPRV.2006.2
4. Teklemariam, G., & Gebremichael, B. (2018). Design and Implementation of an RFID-based Automated Billing System using Arduino. International Journal of Computer Applications, 179(20), 1-5.
5. Zhang, Y., & Chen, Y. (2019). Load Cell Based Weight Measurement System Using Arduino. Proceedings of the 2nd International Conference on Electronics, Communications and Control Engineering.
6. Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431-440.
7. Gupta, S., & Jain, P. (2020). Automation in Retail Using IoT and Embedded Systems. International Journal of Advanced Research in Computer Science, 11(4), 12-17.
8. Mohan, S., & Rajesh, K. (2021). Smart Shopping Cart Using RFID and Arduino. International Journal of Engineering Research and Technology, 10(5), 1023-1028.

9. Bhattacharjee, S., & Pal, A. (2019). Real-time Billing System with Embedded RFID. Journal of Engineering Science and Technology Review, 12(3), 57-63.
10. Patel, N., & Shah, D. (2018). Automation of Retail Billing System using Arduino and IoT. International Journal of Innovative Research in Science, Engineering and Technology, 7(7), 6883-6888.

CODE:

```
#include<LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
SoftwareSerial rfid(A0, A1);

#define buzzer 10
#define check_bill 0
#define green_led A4
#define red_led A5

String master_card_id = "";
String item_1_card_id = "";
String item_2_card_id = "";
String item_3_card_id = "";
String card_id_2 = "";
int item_1_cost = 100;
int item_2_cost = 200;
int item_3_cost = 300;

int total_cost = 0;
bool item_1_added = 0;
bool item_2_added = 0;
bool item_3_added = 0;
bool card_detected = 0;
bool bill_status = 0;
```

```
void setup(void)
{
  rfid.begin(9600);                       // initialize Rfid reader
  pinMode(buzzer , OUTPUT);
  pinMode(green_led , OUTPUT);
  pinMode(red_led , OUTPUT);
  pinMode(check_bill, INPUT);

  digitalWrite(buzzer, LOW);
  digitalWrite(green_led, HIGH);
  digitalWrite(red_led, HIGH);

  lcd.begin(20, 4);
  lcd.setCursor(1, 0);
  lcd.print("Auto Billing Mall");//Auto Billing Mall Shopping Cart with arduino
  lcd.setCursor(3, 1);
  lcd.print("Shopping Cart");
  lcd.setCursor(5, 2);
  lcd.print("With Arduino");
  delay(5000);
  lcd.clear();
  lcd.print(" SWIPE-REGISTER THE");
  lcd.setCursor(5, 1);
  lcd.print("MASTER CARD");
```

```
  master_card_id = register_rfid_cards();  //read master card here

  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("Master card ID is: ");
  lcd.setCursor(0, 2);
  lcd.print(master_card_id);
  delay(1000);
  lcd.clear();
  for (int i = 1; i < 4; i++)  //read material's card in this loop
  {
    lcd.clear();
    lcd.print("SWIPE THE CARD ");
    lcd.setCursor(0, 1);
    lcd.print("FOR Item no.");
    lcd.print(i);
    String card_id = register_rfid_cards();

    lcd.clear();
    if (i == 1)
    {
      item_1_card_id = card_id;
    }
    else if (i == 2)
    {
      item_2_card_id = card_id;
    }
```

```
    else if (i == 3)
    {
      item_3_card_id = card_id;
    }
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Item no.: ");
    lcd.print(i);
    lcd.setCursor(0, 2);
    lcd.print(card_id);
    delay(1000);
  }
}

void loop()
{
  lcd.clear();
  lcd.print("Swipe The Product");
  lcd.setCursor(0, 3);
  lcd.print("COST: ");
  lcd.setCursor(9, 3);
  lcd.print(total_cost);
  delay(100);
```

```
  check_cart_status();
  bill_status  = digitalRead(check_bill);
  if (bill_status == 0)
  {
    check_bill_status();
  }
}


void check_bill_status()
{
  lcd.clear();
  if (item_1_added == 1)
  {
    lcd.setCursor(0, 0);
    lcd.print("Item 1 = ");
    lcd.print(item_1_cost);
  }
  if (item_2_added == 1)
  {
    if (item_1_added == 1)
    {
      lcd.setCursor(0, 1);
    }
    else
    {
      lcd.setCursor(0, 0);
    }
    lcd.print("Item 2 = ");
    lcd.print(item_2_cost);
  }
  if (item_3_added == 1)
  {
    if ((item_1_added == 1) and (item_2_added == 1))
    {
      lcd.setCursor(0, 2);
    }
    else if ((item_1_added == 0) and (item_2_added == 0))
    {
      lcd.setCursor(0, 0);
    }
    else
    {
      lcd.setCursor(0, 1);
    }
    lcd.print("Item 3 = ");
    lcd.print(item_3_cost);
  }
  if (item_1_added == 0 and item_2_added == 0 and item_3_added == 0)
  {
    lcd.print("Cart Is Empty");
  }
```

```
  lcd.setCursor(0, 3);
  lcd.print("Total Bill = ");
  lcd.print(total_cost);
  delay(2000);
}

void check_cart_status()
{
  card_id_2 = read_rfid_card();
  if (card_id_2 == item_1_card_id)
  {
    if (item_1_added == 0)
    {
      green_on__red_off();
      delay(300);
      green_off__red_off();
      total_cost = total_cost + item_1_cost ;
      item_1_added = 1;
      lcd.clear();
      lcd.print("Item 1");
      lcd.setCursor(0, 1);
      lcd.print("Added To Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_1_cost);
      delay(2000);
```

```
    }
    else
    {
      green_off__red_on();
      delay(300);
      green_off__red_off();
      total_cost = total_cost - item_1_cost ;
      item_1_added = 0;
      lcd.clear();
      lcd.print("Item 1");
      lcd.setCursor(0, 1);
      lcd.print("Removed From Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_1_cost);
      delay(2000);
    }
  }
  else if (card_id_2 == item_2_card_id)
  {
    if (item_2_added == 0)
    {
      green_on__red_off();
      delay(300);
      green_off__red_off();
```

```
      total_cost = total_cost + item_2_cost ;
      item_2_added = 1;
      lcd.clear();
      lcd.print("Item 2");
      lcd.setCursor(0, 1);
      lcd.print("Added To Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_2_cost);
      delay(2000);
    }
    else
    {
      green_off__red_on();
      delay(300);
      green_off__red_off();
      total_cost = total_cost - item_2_cost ;
      item_2_added = 0;
      lcd.clear();
      lcd.print("Item 2");
      lcd.setCursor(0, 1);
      lcd.print("Removed From Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_2_cost);
```

```
      delay(2000);
    }
  }
  else if (card_id_2 == item_3_card_id)
  {
    if (item_3_added == 0)
    {
      green_on__red_off();
      delay(300);
      green_off__red_off();
      total_cost = total_cost + item_3_cost ;
      item_3_added = 1;
      lcd.clear();
      lcd.print("Item 3");
      lcd.setCursor(0, 1);
      lcd.print("Added To Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_3_cost);
      delay(2000);
    }
    else
    {
      green_off__red_on();
      delay(300);
      green_off__red_off();
```

```
      total_cost = total_cost - item_3_cost ;
      item_3_added = 0;
      lcd.clear();
      lcd.print("Item 3");
      lcd.setCursor(0, 1);
      lcd.print("Removed From Cart");
      lcd.setCursor(0, 2);
      lcd.print("Price = ");
      lcd.print(item_3_cost);
      delay(2000);
    }
  }
  else if (card_id_2 == master_card_id)
  {
    check_bill_status();
    lcd.clear();
    lcd.print("Purchase Verified");
    lcd.setCursor(0, 2);
    lcd.print("At Billing Counter");
    lcd.setCursor(0, 3);
    lcd.print("--------------------");
    while (1);
  }
  card_id_2 = "";
}
```

```
String read_rfid_card()
{
  String rfid_tag = "";
  //  while (rfid.available() == 0);
  while (rfid.available() > 0)
  {
    rfid_tag += (char)rfid.read();
    delay(2);
    card_detected = 1;
  }
  if ( card_detected == 1)
  {
    digitalWrite(buzzer, HIGH);
    delay(200);
    digitalWrite(buzzer, LOW);
    rfid_tag.trim();
    card_detected = 0;
    if (rfid_tag == master_card_id)
    {
      lcd.clear();
      lcd.print("Master Card");
      lcd.setCursor(0, 1);
      lcd.print("detected");
      digitalWrite(buzzer , HIGH);
      delay(3000);
      digitalWrite(buzzer , LOW);
```

```
    }
    return rfid_tag;
  }
}
String register_rfid_cards()
{
  String rfid_tag = "";
  while (rfid.available() == 0);
  while (rfid.available() > 0)
  {
    rfid_tag += (char)rfid.read();
    delay(2);
  }
  digitalWrite(buzzer, HIGH);
  delay(200);
  digitalWrite(buzzer, LOW);
  rfid_tag.trim();
  return rfid_tag;
}
void green_on__red_off()
{
  digitalWrite(green_led, LOW);
  digitalWrite(red_led, HIGH);
}
```

```
void green_off__red_on()
{
  digitalWrite(green_led, HIGH);
  digitalWrite(red_led, LOW);
}
void green_off__red_off()
{
  digitalWrite(green_led, HIGH);
  digitalWrite(red_led, HIGH);
}
```