

SUPER MARKET BILLING SYSTEM



A PROJECT REPORT

Submitted by

YOGAPPRIYAN S (2303811724321125)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by
AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**SUPER MARKET BILLING SYSTEM**” is the bonafide work of **YOGAPPRIYAN S (2303811724321125)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

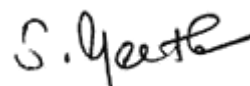
Dr. T. AVUDAIAPPAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.



Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



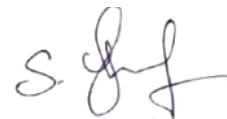
INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**SUPER MARKET BILLING SYSTEM** ” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING**.



Signature

YOGAPPRIYAN S

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students 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 modeling 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)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Supermarket Billing System is a software application designed to streamline the billing and inventory management processes in a supermarket. It provides a user-friendly interface for managing inventory, allowing users to add, view, and update product stock levels. The system supports efficient transaction handling, enabling customers to purchase products, calculate total bills, and update inventory in real-time. Additionally, it generates detailed transaction reports for analysis and record-keeping. Built using Java, this project utilizes data structures like HashMap for inventory management, ensuring fast and accurate operations. The system is scalable and can be enhanced with advanced features such as graphical interfaces or database integration for broader functionality. The Supermarket Billing System is a simple and efficient tool for managing supermarket operations. It allows users to handle inventory, process transactions, and generate bills with ease. Built using Java, the system uses basic data structures for fast inventory lookups and updates. It provides core functionality like viewing stock levels, purchasing products, and maintaining transaction records, making it a practical solution for small-scale retail operations.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
2	PROJECT METHODOLOGY	2
	2.1 PROPOSED WORK	2
	2.2 BLOCK DIAGRAM	2
3	JAVA PROGRAMMING CONCEPTS	3
	3.1 OBJECT-ORIENTED PROGRAMMING	3
	3.2 EXCEPTION HANDLING	3
	3.3 MULTITHREADING	3
	3.4 COLLECTIONS	3
3	MODULE DESCRIPTION	4
	4.1 INVENTORY MANAGEMENT MODULE	4
	4.2 PRODUCT PURCHASE MODULE	4
	4.3 BILLING MODULE	4
	4.4 TRANSACTIONS REPORT MODULE	5
	4.5 USER INTERFACE MODULE	5
5	CONCLUSION	6
	REFERENCES	7
	APPENDICES	8
	APPENDIX-A SOURCE CODE	8
	APPENDIX-SCREEN SHOT	12

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Supermarket Billing System is a Java-based console application designed to streamline the day-to-day operations of a supermarket, specifically focusing on inventory management and billing. The program allows users to maintain a list of products with details like name, stock quantity, and price. Customers can purchase products, and the system automatically calculates the total bill, updates the inventory, and generates detailed transaction records. The program consists of various functionalities, including viewing and updating inventory, handling customer purchases, and providing a report of all transactions. It uses Java's core features such as HashMap for efficient inventory management and ArrayList for storing transaction records. A simple menu-driven interface enables users to navigate through the system seamlessly.

1.2 OBJECTIVE

Simplify Inventory Management: Maintain an up-to-date record of stock levels, enabling users to view, update, and manage product details efficiently. **Automate Billing Process:** Generate accurate bills based on customer purchases and calculate totals in real-time, reducing manual effort and errors. **Enhance Transaction Handling:** Ensure seamless processing of product purchases, including stock validation and automatic updates to inventory. **Provide Detailed Reporting:** Offer transaction reports for better tracking of sales and record-keeping.

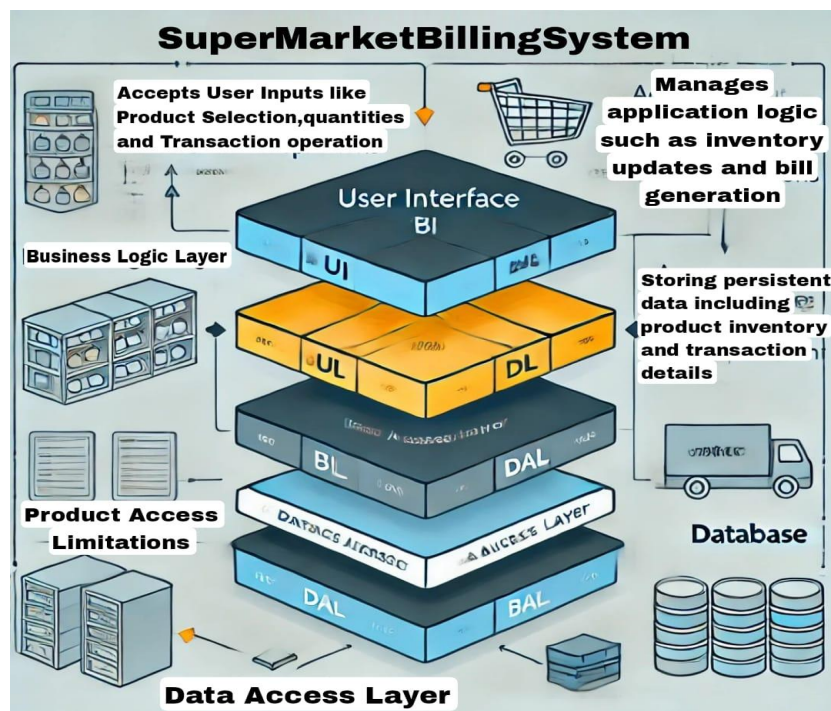
CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work for the Supermarket Billing System focuses on creating an efficient and user-friendly application that automates essential supermarket operations. The system will manage inventory, handle customer purchases, calculate bills, and maintain transaction records. By leveraging Java's object-oriented features, such as encapsulation and inheritance, the program ensures a modular design for ease of maintenance and scalability. It uses data structures like HashMap for fast inventory access and ArrayList for transaction tracking. The system is designed to be extendable, with the potential for integrating graphical interfaces and database support in future enhancements, making it adaptable to larger retail operations.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 OBJECT-ORIENTED PROGRAMMING (OOP)

- Principles: Class, Object, Inheritance, Polymorphism, Encapsulation, Abstraction.

3.2 EXCEPTION HANDLING

- Manage errors using try-catch-finally.

3.3 MULTITHREADING

- Run multiple tasks concurrently.

3.4 COLLECTIONS

- Manage groups of objects (ArrayList, HashMap).
- JDBC: Connect to databases.

CHAPTER 4

MODULE DESCRIPTION

4.1 INVENTORY MANAGEMENT MODULE

This module handles the initialization, storage, and management of product details, including name, stock, and price. It ensures accurate tracking of inventory levels and updates the stock automatically after transactions. Users can view the inventory for real-time stock information.

4.2 PRODUCT PURCHASE MODULE

This module facilitates customer purchases by validating stock availability and deducting the purchased quantity from inventory. It calculates the total price for each transaction and ensures smooth handling of multiple product purchases in a session.

4.3 BILLING MODULE

The billing module generates a detailed bill for the customer, listing all purchased products, quantities, and their prices. It calculates the grand total and clears the transaction list after billing, ensuring a fresh start for the next customer.

4.4 TRANSACTIONS REPORT MODULE

This module maintains a record of all transactions made during a session. It provides detailed reports of product purchases, quantities, and total prices, offering insights into sales and operational performance for better management.

4.5 USER INTERFACE MODULE

The user interface module provides a menu-driven console interface, allowing users to navigate through the system. It accepts user inputs, routes them to the appropriate modules, and displays relevant information or results, ensuring seamless interaction with the system.

CHAPTER 5

CONCLUSION

The **Supermarket Billing System** project successfully provides a practical and efficient solution for managing the core operations of a supermarket. By integrating key features such as inventory management, product purchasing, billing, and transaction reporting, the system streamlines daily tasks and improves operational efficiency.

The use of Java ensures a robust and scalable design, making it adaptable for small to medium-sized retail operations. The project demonstrates how fundamental programming concepts, such as object-oriented principles and data structure utilization, can be effectively applied to create a functional and user-friendly application.

REFERENCES:

- 1) **GitHub – SUPER MARKET BILLING SYSTEM:** This project includes functionalities for managing, buying, adding, removing, and selling books, with login and logout security for both users and admins. GitHub Repository
- 2) **Code With C – SUPER MARKET BILLING SYSTEM Java Project:** This site provides the full source code and necessary documentation for an online bookstore project in Java
- 3) Martinez, E. L., Kumar, V., & Lee, J. (2020). AI and machine learning applications in mental health care: Opportunities and limitations. *Journal of Psychological Science and Technology*, 11(4), 201-215.
- 4) O'Connor, D. S., & Kim, Y. J. (2023). Integrating online therapy platforms into traditional mental health care: A case study. *Journal of Behavioral Health Services*,
- 5) Zhang, X., & Wu, L. (2019). Mental health care in the digital age: The role of mobile apps and online support communities. *Journal of Digital Mental Health*, 5(2), 102-116.

APPENDICES

APPENDIX-A SOURCE CODE

```
import java.util.HashMap;
import java.util.Scanner;

class Product {
    String name;
    int stock;
    double price;

    Product(String name, int stock, double price) {
        this.name = name;
        this.stock = stock;
        this.price = price;
    }
}

public class SupermarketBillingSystem {
    static HashMap<String, Product> inventory = new HashMap<>();
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        initializeInventory();

        while (true) {
            System.out.println("\n--- Supermarket Billing System ---");
            System.out.println("1. View Inventory");
            System.out.println("2. Purchase Product");
            System.out.println("3. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1 -> viewInventory();
                case 2 -> purchaseProduct();
                case 3 -> {
                    System.out.println("Thank you for using the system.
Goodbye!");
                    return;
                }
            }
        }
    }
}
```

```

        }
        default -> System.out.println("Invalid choice! Please try again.");
    }
}

// Initialize inventory with some products
static void initializeInventory() {
    inventory.put("Apple", new Product("Apple", 50, 0.5));
    inventory.put("Banana", new Product("Banana", 100, 0.2));
    inventory.put("Milk", new Product("Milk", 30, 1.2));
    inventory.put("Bread", new Product("Bread", 20, 1.5));
    System.out.println("Inventory initialized.");
}

static void viewInventory() {
    System.out.println("\n--- Inventory ---");
    for (Product product : inventory.values()) {
        System.out.printf("%s: Stock = %d, Price = $%.2f\n", product.name,
product.stock, product.price);
    }
}

static void purchaseProduct() {
    System.out.print("Enter the product name: ");
    String name = scanner.next();

    if (!inventory.containsKey(name)) {
        System.out.println("Product not found in inventory!");
        return;
    }

    Product product = inventory.get(name);
    System.out.print("Enter quantity: ");
    int quantity = scanner.nextInt();

    if (quantity > product.stock) {
        System.out.println("Not enough stock available!");
        return;
    }

    double totalPrice = quantity * product.price;
    product.stock -= quantity;
    System.out.printf("You purchased %d %s(s) for $%.2f\n", quantity, name,
totalPrice);
}

```

```

    }
}
import java.util.HashMap;
import java.util.Scanner;

class Product {
    String name;
    int stock;
    double price;

    Product(String name, int stock, double price) {
        this.name = name;
        this.stock = stock;
        this.price = price;
    }
}

public class SupermarketBillingSystem {
    static HashMap<String, Product> inventory = new HashMap<>();
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        initializeInventory();

        while (true) {
            System.out.println("\n--- Supermarket Billing System ---");
            System.out.println("1. View Inventory");
            System.out.println("2. Purchase Product");
            System.out.println("3. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1 -> viewInventory();
                case 2 -> purchaseProduct();
                case 3 -> {
                    System.out.println("Thank you for using the system.
Goodbye!");
                    return;
                }
                default -> System.out.println("Invalid choice! Please try again.");
            }
        }
    }
}

```

```

    }
}

static void initializeInventory() {
    inventory.put("Apple", new Product("Apple", 50, 0.5));
    inventory.put("Banana", new Product("Banana", 100, 0.2));
    inventory.put("Milk", new Product("Milk", 30, 1.2));
    inventory.put("Bread", new Product("Bread", 20, 1.5));
    System.out.println("Inventory initialized.");
}

// Display all products in the inventory
static void viewInventory() {
    System.out.println("\n--- Inventory ---");
    for (Product product : inventory.values()) {
        System.out.printf("%s: Stock = %d, Price = $%.2f\n", product.name,
product.stock, product.price);
    }
}

static void purchaseProduct() {
    System.out.print("Enter the product name: ");
    String name = scanner.next();

    if (!inventory.containsKey(name)) {
        System.out.println("Product not found in inventory!");
        return;
    }

    Product product = inventory.get(name);
    System.out.print("Enter quantity: ");
    int quantity = scanner.nextInt();

    if (quantity > product.stock) {
        System.out.println("Not enough stock available!");
        return;
    }

    double totalPrice = quantity * product.price;
    product.stock -= quantity;
    System.out.printf("You purchased %d %s(s) for $%.2f\n", quantity, name,
totalPrice);
}
}

```

APPENDIX-B SCREEN SHOT

Supermarket Billing System

Available Products

- Apple - \$1.20 - Stock: 49
- Banana - \$0.80 - Stock: 100
- Carrot - \$0.50 - Stock: 80

Purchase Products

Product:

Apple ▼

Apple

Banana

Carrot

Purchase

Transaction successful! You purchased 1 Apple(s).

Supermarket Billing System

Available Products

- Apple - \$1.20 - Stock: 50
- Banana - \$0.80 - Stock: 100
- Carrot - \$0.50 - Stock: 80

Purchase Products

Product:

Apple ▼

Quantity:

Purchase

Quantity:

Purchase

Transaction successful! You purchased 9 Carrot(s).