SUPERMARKET BILLING SYSTEM

A PROJECT REPORT

Submitted by

TAMILDEEPAA A(2303811724322116)

in partial fulfillment of requirements for the award of the course

CGB1221-DATABASE MANAGEMENT SYSTEMS

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

JUNE- 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"SUPERMARKET BILLING SYSTEM"** is the bonafide work of **TAMILDEEPAA A(2303811724322116)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE	**SIGNATURE**
Dr.T. AVUDAIAPPAN, M.E.,Ph.D.,	Mrs.S. GEETHA, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
ASSOCIATE PROFESSOR	ASSISTANT PROFESSOR
Department of Artificial Intelligence	Department of Artificial Intelligence
K.Ramakrishnan College of Technology (Autonomous)	K.Ramakrishnan College of Technology (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on …04.06.2025…….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **"SUPERMARKET BILLING SYSTEM"** is the result of original work done by me and best of my 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 completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS.**

.

Signature

.

TAMILDEEPAA A

Place: Samayapuram

Date:04.06.2025

ACKNOWLEDGEMENT

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

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

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

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

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep expression and sincere gratitude to my project supervisor **Mrs.S.GEETHA, M.E.,** Department of **ARTIFICIAL INTELLIGENCE,** for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out this project.

I render my sincere thanks to 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 my departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

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

Mission:

- Be a center 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.

DEPARTMENT

Vision:

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

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

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.

ABSTRACT

The Supermarket Billing System is a computerized solution designed to automate the billing process in retail supermarket environments. This system enables efficient management of sales transactions by recording customer purchases, generating invoices, calculating totals, and handling various payment modes. It integrates key components such as product inventory, customer and employee databases, billing counters, and payment records to streamline operations and reduce human error. Through a relational database structure, it supports real-time updates of stock levels and ensures data consistency with triggers and stored procedures. The system enhances operational efficiency, improves customer service, and provides a reliable framework for managing sales data, inventory control, and financial reporting in a retail setting.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Supermarket Billing System is a database-driven application developed to efficiently manage the operations of a retail supermarket. It automates essential functions such as customer billing, product inventory control, employee tracking, and payment processing. By leveraging structured query language (SQL) and relational database concepts, the system ensures real-time data consistency, accuracy, and scalability. It incorporates triggers, constraints, and stored procedures to support transactional integrity and minimize manual intervention. This project demonstrates the ability to model real-time problem-solving through systematic database design, integrating multiple modules to provide a reliable and comprehensive solution for retail management.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The objective of the Supermarket Billing System project is to develop a comprehensive and automated solution that streamlines the billing and inventory processes within a supermarket. This system is intended to replace traditional manual billing methods with a digital platform that ensures speed, accuracy, and ease of use. It aims to facilitate efficient handling of customer transactions, automatically calculate billing amounts, and manage real-time inventory updates based on product sales. Additionally, the system maintains detailed records of customers, employees, products, and payments, thereby supporting better operational control and customer service. By integrating key features such as secure payment processing, user authentication, and database integrity mechanisms like triggers and stored procedures, the project seeks to reduce human errors, enhance data accuracy, and provide a scalable foundation for future business needs.

1.2 OVERVIEW

The Supermarket Billing System is a database-driven application designed to handle the core operations of a retail supermarket. It offers an end-to-end solution for managing the billing process, tracking inventory, and maintaining detailed records of customers, employees, and transactions. The system automates the generation of bills by recording the products selected by customers, calculating the total payable amount, and updating the inventory accordingly. The system also enhances data management through the use of constraints and relationships between tables, ensuring the integrity and reliability of the stored information. Overall, the Supermarket Billing System provides a scalable and efficient framework for handling supermarket transactions, improving both customer service and operational efficiency.

1.3 SQL AND DATABASE CONCEPTS

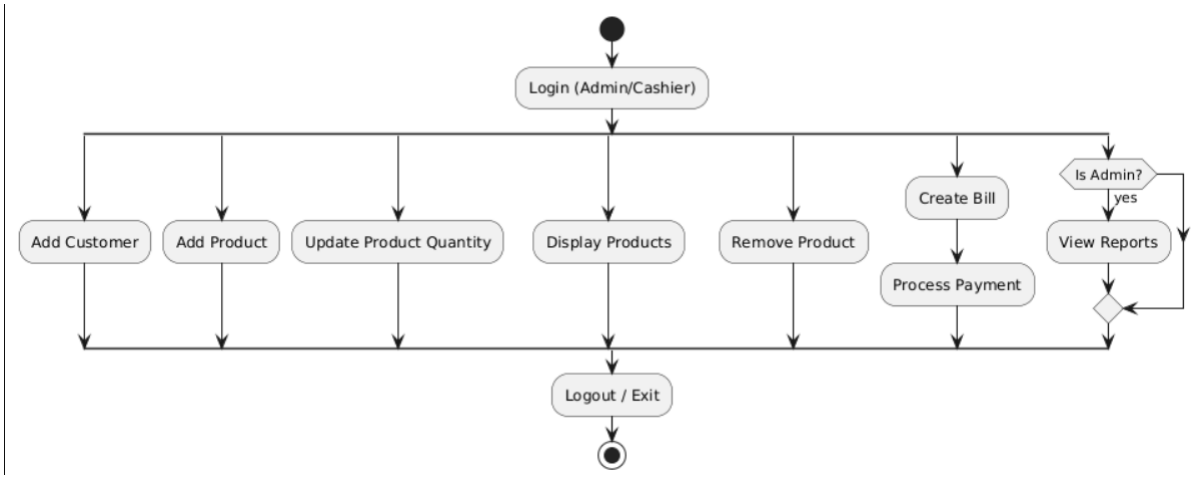
- The system is built using **Structured Query Language (SQL)** to manage and manipulate relational data efficiently.
- A **relational database model** is used, where data is stored in interrelated tables such as customer, employee, product, cart, billing_counter, and payment.
- **Primary keys** and **foreign keys** are used to enforce relationships between tables, ensuring **referential integrity**.
- **Constraints** like NOT NULL, UNIQUE, and AUTO_INCREMENT ensure data validity and consistency.
- **Stored procedures** (e.g., totalAmount) are used to perform predefined operations, such as calculating total bill amounts.
- **Triggers** are implemented to automate tasks—e.g., updating product inventory automatically when a purchase is made.
- **Normalization** is applied to organize data efficiently, reduce redundancy, and improve integrity.
- **JOIN operations** are used to retrieve related data from multiple tables, enabling complex queries and reports.
- The database design supports **real-time data processing**, making the system responsive and reliable.

CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work focuses on the development of a comprehensive Supermarket Billing System that automates and streamlines the billing, inventory management, and customer handling processes within a retail supermarket. The system is designed to replace manual billing procedures with a reliable, accurate, and time-efficient digital platform. It will enable billing personnel to quickly generate invoices, calculate totals, and accept multiple modes of payment while maintaining real-time updates of product inventory. The system is expected to offer features like dynamic billing generation, stock level tracking, customer history maintenance, and reporting functionalities. The ultimate goal of the proposed work is to provide a robust and scalable solution that enhances operational efficiency, minimizes human errors, and improves the overall customer experience in the supermarket environment.

2.2 BLOCK DIAGRAM



CHAPTER 3
MODULE DESCRIPTION

3.1 CUSTOMER MANAGEMENT MODULE

This module is responsible for maintaining complete records of customers including their names, contact information, and addresses linked via zip codes. This module ensures that each customer is uniquely identified and can be associated with their respective purchases and payment history. It supports personalized service, helps track customer loyalty, and enables targeted promotions or offers based on buying behavior.

3.2 PRODUCT AND INVENTORY MODULE

This module manages all details related to products such as type, model ID, name, price, and available quantity. It ensures real-time inventory updates by automatically adjusting stock levels when purchases are made, using triggers. This module prevents overselling, supports product categorization, and plays a crucial role in maintaining accurate inventory data, thereby ensuring efficient stock management.

3.3 BILLING AND CART MODULE

This module is central to the system, handling the addition of selected items to a virtual cart and calculating the total billing amount. It generates invoices by linking the bill to the respective customer and the employee who processed the sale. With real-time timestamping, this module ensures billing transparency and accuracy, while supporting error-free transactions and smooth checkout operations.

3.4 PAYMENT MODULE

This module manages the financial transactions associated with each bill. It records payment details such as the amount, mode of payment (e.g., cash or card), and the time of payment. Each payment is tied to a specific bill and customer, ensuring traceability. This module also supports financial tracking, facilitates daily sales reports, and ensures data consistency through relational integrity.

3.5 EMPLOYEE MANAGEMENT MODULE

This module stores comprehensive information about supermarket employees, including their names, roles, contact details, and associated locations. It helps track which employee handles which transactions, allowing for role-based access and operational accountability. This module also supports human resource functions and internal security by ensuring that only authorized personnel can access certain system features.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

CONCLUSION:

The Supermarket Billing System offers a reliable, efficient, and scalable solution for managing day-to-day retail operations within a supermarket environment. By automating key processes such as customer billing, inventory updates, and payment tracking, the system significantly reduces human errors and improves operational speed. Its modular design ensures that various components like product management, employee tracking, and user authentication work seamlessly together through a well-structured relational database. Through the use of SQL features such as stored procedures, triggers, and constraints, the system maintains data accuracy, integrity, and real-time updates. The integration of different modules provides a holistic approach to retail management, ensuring both customer satisfaction and administrative efficiency. In conclusion, the system not only simplifies the billing process but also lays a strong foundation for future enhancements such as data analytics, mobile integration, and real-time business intelligence.

FUTURE SCOPE:

The future scope of supermarket billing systems is vast and promising, with opportunities to integrate advanced technologies like artificial intelligence and machine learning for better customer behavior analysis and inventory optimization. Cloud-based solutions will enable centralized data management and real-time access across multiple store locations, improving operational efficiency. The adoption of mobile and contactless payment methods will enhance customer convenience and speed up checkout processes. Automation through self-checkout kiosks and RFID scanning can reduce labor costs and waiting times. Additionally, enhanced inventory management with real-time stock updates and automated reorder alerts will help maintain optimal stock levels. Integration with supply chain and vendor management systems will streamline procurement and delivery tracking.

APPENDIX A-SOURCE CODE

```
-- phpMyAdmin SQL Dump

-- version 4.9.1

-- https://www.phpmyadmin.net/

-- Host: 127.0.0.1

-- Generation Time: Nov 28, 2019 at 08:00 PM

-- Server version: 10.4.8-MariaDB

-- PHP Version: 7.3.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;

/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;

/*!40101 SET NAMES utf8mb4 */;

-- Database: `supermarketdb`

DELIMITER $$
```

```
-- Procedures

CREATE DEFINER=`root`@`localhost` PROCEDURE `totalAmount` (IN
`billno` INT(9) UNSIGNED)  SELECT SUM(amount) FROM cart WHERE
bill_no = billno$$

DELIMITER ;

-- Table structure for table `address`

CREATE TABLE `address` (

  `zipcode` int(6) NOT NULL,

  `state` varchar(255) NOT NULL,

  `district` varchar(255) NOT NULL,

  `city` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `address`


INSERT INTO `address` (`zipcode`, `state`, `district`, `city`) VALUES

(561203, 'Karnataka', 'Bangalore Rural', 'Doddaballapur'),

(562157, 'Karnataka', 'Bangalore', 'Bangalore'),

(584128, 'Karnataka', 'Raichur', 'SINDHANUR'),

(587101, 'Karnataka', 'Bagalkot', 'Old Bagalkot'),

(591242, 'Karnataka', 'Belgaum', 'Athni');

-- Table structure for table `billing_counter`

CREATE TABLE `billing_counter` (

  `bill_no` int(9) NOT NULL,
```

```
  `customer_id` int(4) NOT NULL,
  `employee_id` int(3) NOT NULL,
  `bdate` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `billing_counter`

INSERT INTO `billing_counter` (`bill_no`, `customer_id`, `employee_id`,
`bdate`) VALUES
(70, 1111, 111, '2019-11-23 23:53:34'),
(72, 1112, 111, '2019-11-24 11:43:07'),
(79, 2226, 111, '2019-11-24 14:49:20'),
(80, 2227, 111, '2019-11-24 16:06:14'),
(115, 2227, 111, '2019-11-29 00:05:18'),
(120, 2229, 111, '2019-11-29 00:22:49');

-- Table structure for table `cart`

CREATE TABLE `cart` (
  `cart_id` int(4) NOT NULL,
  `model_id` int(5) NOT NULL,
  `quantity` int(3) NOT NULL,
  `amount` int(8) NOT NULL,
  `bill_no` int(9) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `cart`
```

```
INSERT INTO `cart` (`cart_id`, `model_id`, `quantity`, `amount`, `bill_no`) VALUES

(44, 12111, 1, 90, 70),

(47, 12111, 1, 90, 72),

(48, 12122, 2, 100, 72),

(66, 12111, 2, 200, 79),

(67, 12122, 1, 50, 79),

(68, 12111, 2, 100, 80),

(69, 12122, 5, 500, 80),

(118, 12111, 5, 1140, 115),

(125, 12111, 2, 456, 120),

(126, 12122, 1, 50, 120);

-- Triggers `cart`

DELIMITER $$

CREATE TRIGGER `amounts` BEFORE INSERT ON `cart` FOR EACH ROW BEGIN

UPDATE p_name,product

SET p_name.quantity=p_name.quantity-NEW.QUANTITY

WHERE NEW.MODEL_ID=product.model_id AND

product.name_id=p_name.name_id;

END

$$

DELIMITER ;
```

```
-- Table structure for table `customer’

CREATE TABLE `customer` (
  `customer_id` int(4) NOT NULL,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone_no` bigint(10) NOT NULL,
  `zipcode` int(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `customer`

INSERT INTO `customer` (`customer_id`, `name`, `email`, `phone_no`, `zipcode`) VALUES
(1111, 'Vinayak S P', 'vinay@gmail.com', 9632587410, 562157),
(1112, 'Basu', 'basu@gmail.com', 8965231470, 584128),
(2226, 'Sammed', 'sammed@gmail.com', 8523987410, 591242),
(2227, 'Bhuvan', 'bhuvan@gmail.com', 8563256989, 562157),
(2229, 'Sachin Koppad', 'sachin@gmail.com', 6532568956, 587101)

-- Table structure for table `employee`

CREATE TABLE `employee` (
  `employee_id` int(3) NOT NULL,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone_no` bigint(10) NOT NULL,
  `role` varchar(255) NOT NULL,
```

```
  `zipcode` int(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- Dumping data for table `employee`
INSERT INTO `employee` (`employee_id`, `name`, `email`, `phone_no`, `role`, `zipcode`) VALUES
(111, 'NBPatil', 'patil@gmail.com', 9955117755, 'Biller', 584128),
(112, 'Chandru', 'chand@gmail.com', 9658325614, 'Volunteer', 562157),
(114, 'Sachin', 'sachin@gmail.com', 8695352665, 'Security guard', 587101),
(116, 'Abhi', 'abhi@gmail.com', 6369856325, 'Biller', 561203);
                -- Table structure for table `payment`
CREATE TABLE `payment` (
  `payment_id` int(8) NOT NULL,
  `customer_id` int(4) NOT NULL,
  `pdate` datetime NOT NULL DEFAULT current_timestamp(),
  `amount` int(8) NOT NULL,
  `mode` varchar(255) NOT NULL,
  `bill_no` int(9) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
                -- Dumping data for table `payment`
INSERT INTO `payment` (`payment_id`, `customer_id`, `pdate`, `amount`, `mode`, `bill_no`) VALUES
(6, 1111, '2019-11-23 23:53:53', 90, 'Cash', 70),
(7, 1112, '2019-11-24 11:44:01', 190, 'Cash', 72),
```

```
(10, 2226, '2019-11-24 14:49:52', 250, 'Cash', 79),
(11, 2227, '2019-11-24 16:06:50', 600, 'Card', 80),
(20, 2227, '2019-11-29 00:05:54', 1140, 'Card', 115),
(25, 2229, '2019-11-29 00:23:21', 506, 'Cash', 120);
-- Table structure for table `product`
CREATE TABLE `product` (
  `model_id` int(5) NOT NULL,
  `type` varchar(255) NOT NULL,
  `name_id` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- Dumping data for table `product`
INSERT INTO `product` (`model_id`, `type`, `name_id`) VALUES
(12111, 'Tea Powder', 1),
(12122, 'Toothpaste', 14),
(12123, 'Mobile', 15);
-- Table structure for table `product_shelves`
CREATE TABLE `product_shelves` (
  `shelf_id` int(2) NOT NULL,
  `type` varchar(255) NOT NULL,
  `category` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-- Dumping data for table `product_shelves`
```

```
INSERT INTO `product_shelves` (`shelf_id`, `type`, `category`) VALUES
(1, 'Tea Powder', 'Daily Needs'),
(10, 'Toothpaste', 'Daily Needs'),
(18, 'Perfumes', 'Daily needs'),
(19, 'Mobile', 'Electronics');
-- Table structure for table `p_name`
CREATE TABLE `p_name` (
  `name_id` int(5) NOT NULL,
  `name` varchar(255) NOT NULL,
  `price` int(8) NOT NULL,
  `quantity` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Dumping data for table `p_name`
INSERT INTO `p_name` (`name_id`, `name`, `price`, `quantity`) VALUES
(1, 'Red Label Natural Care 500g', 228, 193),
(14, 'Colgate', 50, 199),
(15, 'Redmi Note8 Pro 6GB and 64GB', 14999, 200);
-- Table structure for table `users`
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
```

```
  `password` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Dumping data for table `users`
INSERT INTO `users` (`user_id`, `username`, `password`, `name`) VALUES
(2, 'admin', 'admin', 'admin'),
(3, 'user', 'user', 'user');
-- Indexes for dumped tables
-- Indexes for table `address`
ALTER TABLE `address`
  ADD PRIMARY KEY (`zipcode`),
  ADD UNIQUE KEY `zipcode` (`zipcode`);
-- Indexes for table `billing_counter`
ALTER TABLE `billing_counter`
  ADD PRIMARY KEY (`bill_no`),
  ADD UNIQUE KEY `bill_no` (`bill_no`),
  ADD KEY `fk_bcid` (`customer_id`),
  ADD KEY `fk_beid` (`employee_id`);
-- Indexes for table `cart`
ALTER TABLE `cart`
  ADD PRIMARY KEY (`cart_id`),
  ADD KEY `fk_cbno` (`bill_no`);
```

```
-- Indexes for table `customer`
ALTER TABLE `customer`
  ADD PRIMARY KEY (`customer_id`),
  ADD KEY `fk_zipcode` (`zipcode`);
-- Indexes for table `employee`
ALTER TABLE `employee`
  ADD PRIMARY KEY (`employee_id`),
  ADD KEY `fk_ezip` (`zipcode`);
-- Indexes for table `payment`
ALTER TABLE `payment`
  ADD PRIMARY KEY (`payment_id`),
  ADD KEY `fk_pbno` (`bill_no`),
  ADD KEY `fk_pcid` (`customer_id`);
-- Indexes for table `product`
ALTER TABLE `product`
  ADD PRIMARY KEY (`model_id`),
  ADD KEY `fk_pnid` (`name_id`),
  ADD KEY `fk_ptype` (`type`);
-- Indexes for table `product_shelves`
ALTER TABLE `product_shelves`
  ADD PRIMARY KEY (`shelf_id`,`type`),
  ADD UNIQUE KEY `type` (`type`);
```

```
-- Indexes for table `p_name`

ALTER TABLE `p_name`

  ADD PRIMARY KEY (`name_id`),

  ADD UNIQUE KEY `name` (`name`);

-- Indexes for table `users`

ALTER TABLE `users`

  ADD PRIMARY KEY (`user_id`);

-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table `billing_counter`

ALTER TABLE `billing_counter`

  MODIFY `bill_no` int(9) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=121;

-- AUTO_INCREMENT for table `cart`

ALTER TABLE `cart`

  MODIFY `cart_id` int(4) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=127;

-- AUTO_INCREMENT for table `customer`

ALTER TABLE `customer`

  MODIFY `customer_id` int(4) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2230;

-- AUTO_INCREMENT for table `employee`

ALTER TABLE `employee`

  MODIFY `employee_id` int(3) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=117;
```

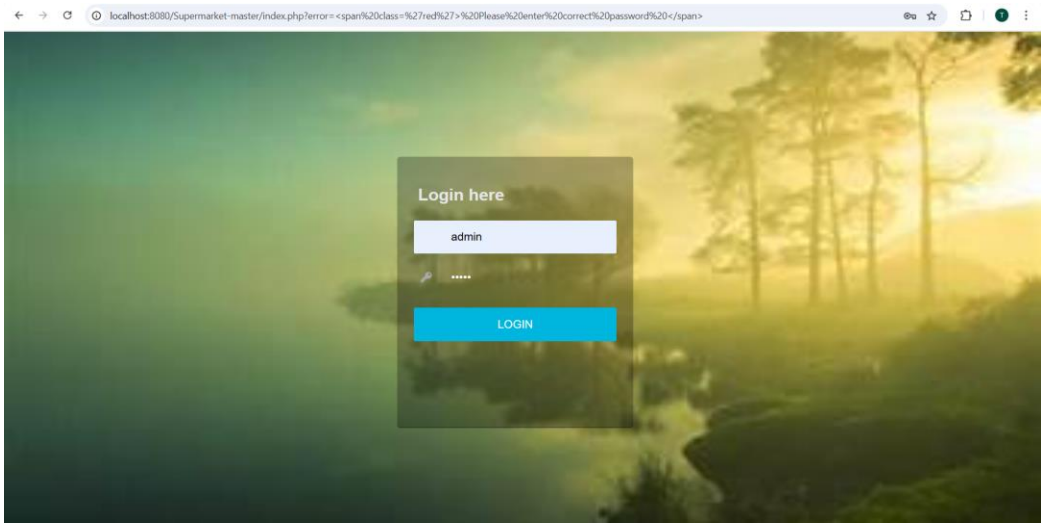
```
-- AUTO_INCREMENT for table `payment`

ALTER TABLE `payment`

  MODIFY `payment_id` int(8) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=26;

-- AUTO_INCREMENT for table `product`

ALTER TABLE `product`

  MODIFY `model_id` int(5) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=12124;

-- AUTO_INCREMENT for table `product_shelves`

ALTER TABLE `product_shelves`

  MODIFY `shelf_id` int(2) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=20;

-- AUTO_INCREMENT for table `p_name`

ALTER TABLE `p_name`

  MODIFY `name_id` int(5) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=16;

-- AUTO_INCREMENT for table `users`

ALTER TABLE `users`

  MODIFY `user_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=4;

-- Constraints for dumped tables

-- Constraints for table `billing_counter`

ALTER TABLE `billing_counter`
```

```
  ADD CONSTRAINT `fk_bcid` FOREIGN KEY (`customer_id`)
REFERENCES `customer` (`customer_id`) ON DELETE CASCADE ON
UPDATE CASCADE,

  ADD CONSTRAINT `fk_beid` FOREIGN KEY (`employee_id`)
REFERENCES `employee` (`employee_id`) ON DELETE CASCADE ON
UPDATE CASCADE;

-- Constraints for table `cart`

ALTER TABLE `cart`

  ADD CONSTRAINT `fk_cbno` FOREIGN KEY (`bill_no`) REFERENCES
`billing_counter` (`bill_no`) ON DELETE CASCADE ON UPDATE
CASCADE;

-- Constraints for table `customer`

ALTER TABLE `customer`

  ADD CONSTRAINT `fk_zipcode` FOREIGN KEY (`zipcode`)
REFERENCES `address` (`zipcode`) ON DELETE CASCADE ON UPDATE
CASCADE;

-- Constraints for table `employee`

ALTER TABLE `employee`

  ADD CONSTRAINT `fk_ezip` FOREIGN KEY (`zipcode`) REFERENCES
`address` (`zipcode`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `payment`

ALTER TABLE `payment`

  ADD CONSTRAINT `fk_pbno` FOREIGN KEY (`bill_no`) REFERENCES
`billing_counter` (`bill_no`) ON DELETE CASCADE ON UPDATE
CASCADE,

  ADD CONSTRAINT `fk_pcid` FOREIGN KEY (`customer_id`)
REFERENCES `customer` (`customer_id`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

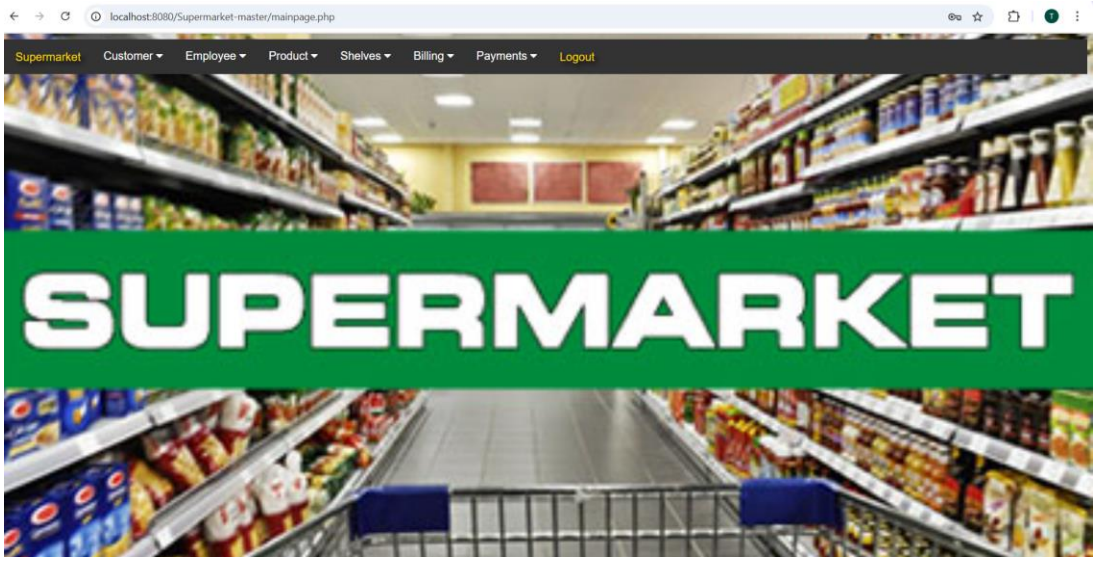
```
-- Constraints for table `product`

ALTER TABLE `product`

  ADD CONSTRAINT `fk_pnid` FOREIGN KEY (`name_id`) REFERENCES
`p_name` (`name_id`) ON DELETE CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `fk_ptype` FOREIGN KEY (`type`) REFERENCES
`product_shelves` (`type`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

APPENDIX B-SCREENSHOTS

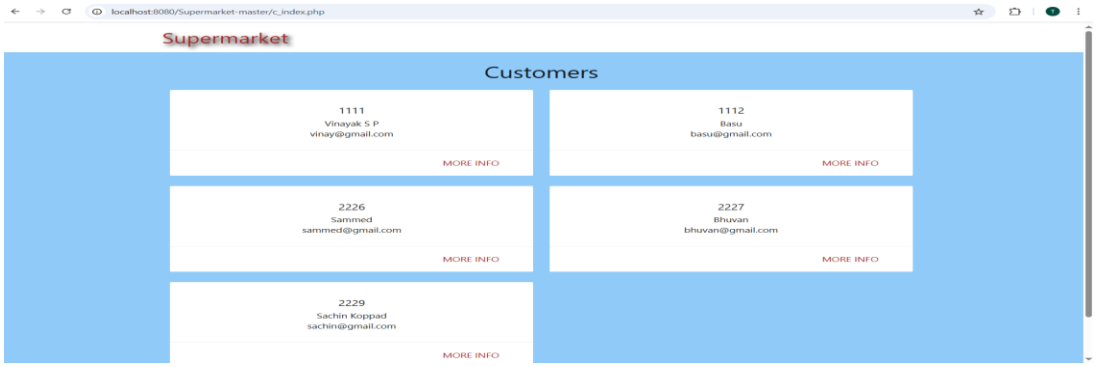
LOGIN PAGE



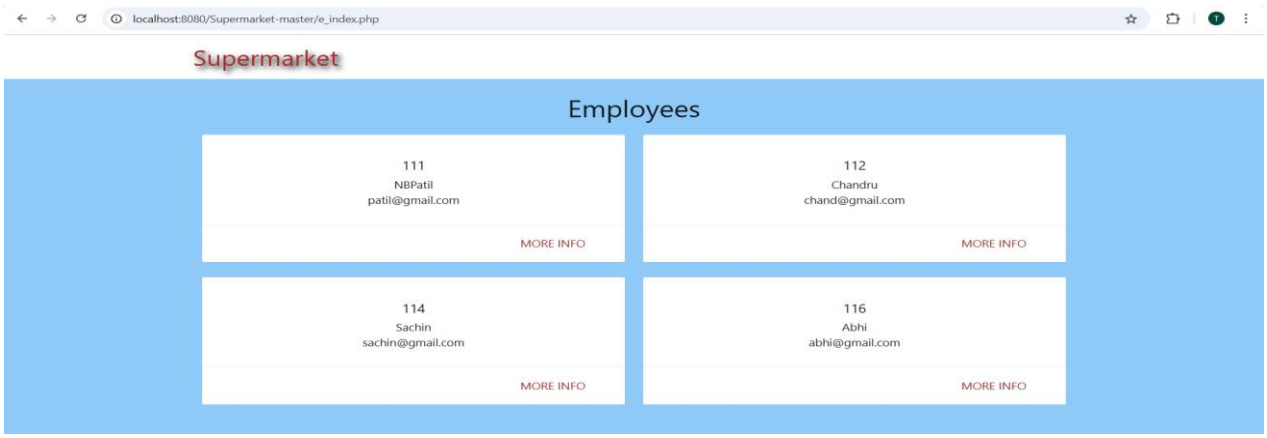
MAIN PAGE



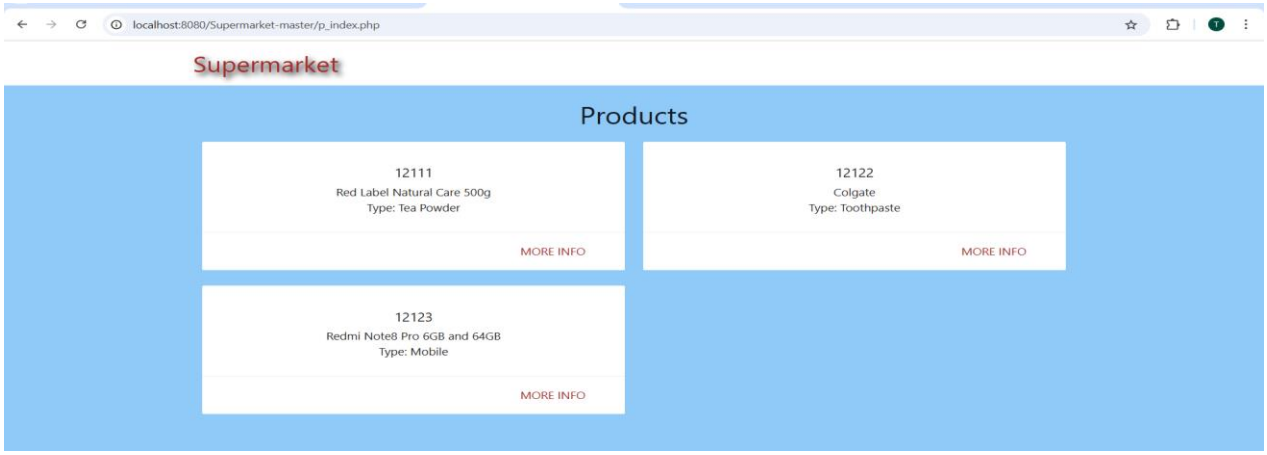
CUSTOMER DETAILS



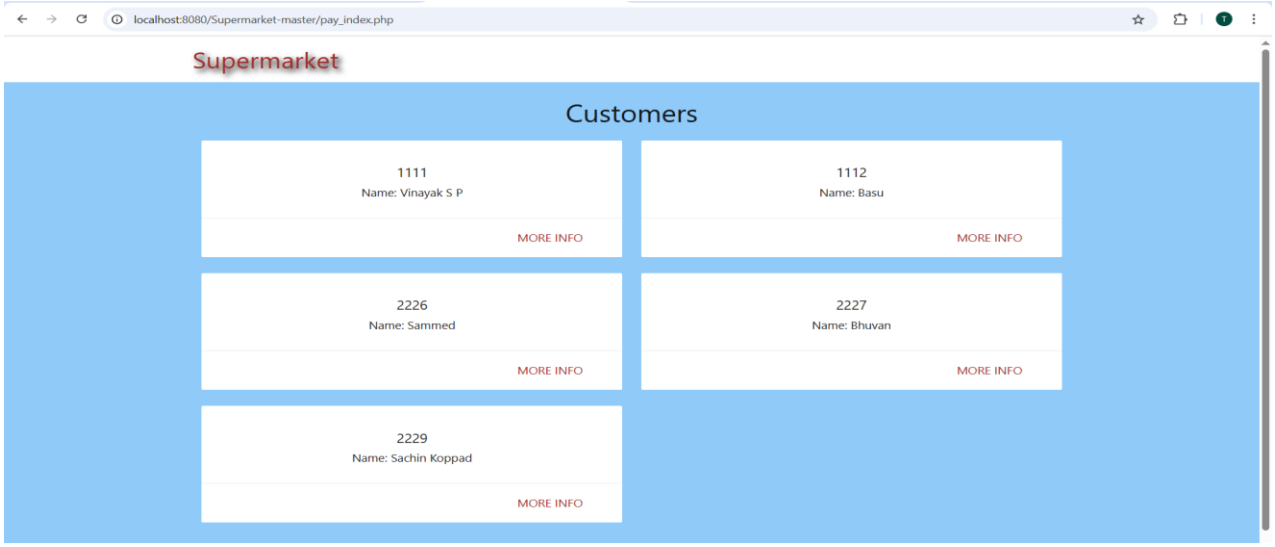
EMPLOYEE DETAILS



PRODUCTS



PAYMENTS



REFERENCES

1. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed., McGraw-Hill, 2020.
2. Elmasri, Ramez and Shamkant B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson Education, 2016.
3. GeeksforGeeks, "Introduction to SQL Triggers," [Online]. Available: https://www.geeksforgeeks.org/sql-triggers/. [Accessed: 01-Jun-2025].
4. Oracle, "What is a Stored Procedure?", Oracle Documentation. [Online]. Available: https://docs.oracle.com/. [Accessed: 01-Jun-2025].
5. PHPMyAdmin, "phpMyAdmin Official Documentation," [Online]. Available: https://www.phpmyadmin.net/. [Accessed: 01-Jun-2025].