

RETAIL POINT OF SALE SYSTEM

A PROJECT REPORT

Submitted by

MEERA B

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING



**K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**

SAMAYAPURAM, TRICHY



ANNA UNIVERSITY CHENNAI 600025

JUNE 2025

RETAIL POINT OF SALE SYSTEM

Submitted by

MEERA B (8115U23EC062)

in partial fulfillment of requirements for the award of the course

EGB1221 – DATABASE MANAGEMENT SYSTEM

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Mrs.R.Nalini

Department of Computer Science And Engineering



**K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**

Under

ANNA UNIVERSITY, CHENNAI





**K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**



**Under
ANNA UNIVERSITY, CHENNAI**

BONAFIDE CERTIFICATE

Certified that this project report on “**RETAIL POINT OF SALE SYSTEM**” is the bonafide work of **MEERA B (8115U23EC062)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

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

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of CSE

K. Ramakrishnan College of Engineering
(Autonomous)

Samayapuram-621112.

SIGNATURE

Mrs. R. NALINI, M.E., (Ph.D.),

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K. Ramakrishnan College of Engineering
(Autonomous)

Samayapuram-621112.

Submitted for the viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER



**K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**



**Under
ANNA UNIVERSITY, CHENNAI**

DECLARATION BY THE CANDIDATE

I declare that to the best of my knowledge the work reported here in has been composed solely by me and that it has not been in whole or in part in any previous application for a degree.

Submitted for the project Viva- Voce held at K.Ramakrishnan College of Engineering
on _____

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Engineering (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 our 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. D. SRINIVASAN, B.E, M.E., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.**, Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I am grateful to **Mrs. R. NALINI, M.E., (Ph.D)** Assistant Professor,Computer Science And Engineering, K. Ramakrishnan College of Engineering (Autonomous), for her guidance and valuable suggestions during the course of study.

I render our 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 our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTITUTION

- M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.
- M3: To provide education for developing high-quality professionals to transform the society.

VISION OF DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successful career.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

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

A Retail Point of Sale (POS) system is a combination of hardware and software used by retailers to facilitate sales transactions and manage day-to-day business operations. At its core, the POS system allows for the processing of customer purchases, including scanning items, calculating totals, accepting payments, and printing receipts. Beyond basic transaction handling, modern POS systems often include advanced features such as inventory management, employee tracking, customer relationship management, and sales analytics. These features help businesses maintain accurate stock levels, monitor staff performance, and better understand customer behavior. With the rise of cloud computing and mobile technologies, POS systems have evolved to offer greater flexibility and accessibility. Cloud-based POS systems enable real-time data synchronization across multiple store locations, allowing managers to monitor operations remotely. This digital transformation has helped retailers improve operational efficiency, reduce human error, and enhance the overall customer experience. As a result, POS systems have become a vital tool for competitive advantage in the modern retail landscape.

ABSTRACT WITH POs AND PSOs MAPPING
CO 5: BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME
PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>A Retail Point of Sale (POS) system is a combination of hardware and software used to process sales and manage inventory in stores. It handles billing, payments, and customer data.</p> <p>Modern POS systems improve efficiency, reduce errors, and provide real-time data, helping retailers run their operations smoothly and serve customers better.</p>	<p>PO1 PO2 PO3 PO5 PO6 PO9 PO10 PO12</p>	<p>PSO1 PSO2</p>

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

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	ix
	LIST OF FIGURES	xiii
	LIST OF TABLES	xiii
	LIST OF ABBREVIATIONS	xiii
1	INTRODUCTION	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Technologies Used	2
2	PROJECT METHODOLOGY	4
	2.1 Problem Description	4
	2.2 Proposed Work	4
	2.3 ER Diagram	5
	2.4 ER Diagram Overview	5
3	MODULE DESCRIPTION	6
	3.1 User Management Module	6
	3.2 ProductManagement Module	6
	3.3 Sales & Billing Module	7
	3.4 Inventory Management Module	7
	3.5 Payment Module	8

4	Implementation And Test Cases	9
	4.1 Table Structures	9
	4.2 SQL Queries	11
	4.3 Test Cases and Results	14
5	CONCLUSION & FUTURE SCOPE	15
	Conclusion	15
	Future Scope	15
	APPENDIX A (SOURCE CODE)	16
	APPENDIX B (SCREENSHOTS)	33
	REFERENCES	36

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
2.3.1	ER Diagram of Retail Pos System	5

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
4.1	Customer	9
4.2	Product	9
4.3	Employee	9
4.4	Supplier	10
4.5	Sales	10
4.6	Payment	10
4.7	Receipt	11

LIST OF ABBREVIATIONS

POS	Point of Sale
DBMS	Database Management System
INV No	Invoice Number
UPI	Unified Payments Interface
ID	Identification
USERID	User ID
SQL	Structured Query Language
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 Objective

The primary objective of a Retail Point of Sale (POS) system is to simplify and speed up the sales process in retail environments. By automating tasks such as billing, applying discounts, processing payments, and printing receipts, the POS system helps ensure fast and accurate transactions. This leads to shorter checkout times, reduced human errors, and an overall better customer experience.

Beyond handling sales, a POS system also plays a critical role in inventory management. It tracks stock levels in real time, alerts staff when items are low, and helps prevent overstocking or stockouts. This level of control allows retailers to make informed purchasing decisions and reduce waste. Additionally, POS systems often include reporting and analytics tools that provide insights into sales trends, peak business hours, and employee performance.

Overall, a Retail POS system is designed to increase operational efficiency, improve accuracy, and support better business decisions. It is an essential tool for modern retail businesses aiming to stay competitive and deliver a seamless shopping experience to customers.

1.2 Overview

A Retail Point of Sale (POS) system is a technology solution used by retail businesses to manage sales transactions and streamline store operations. It typically includes both hardware—such as cash registers, barcode scanners, receipt printers, and card readers—and software that handles functions like sales processing, inventory control, customer management, and reporting.

Modern POS systems go beyond basic checkout functionality. They provide real-time data on sales and inventory, support multiple payment methods, and often integrate with other business systems such as accounting or e-commerce platforms. Cloud-based POS systems offer additional benefits like remote access, automatic updates, and data backup, making them ideal for multi-location retail operations.

Overall, a POS system is a central component of retail management that improves efficiency, enhances customer service, and supports data-driven decision-making. It helps businesses operate smoothly, reduce errors, and adapt to changing market demands.

1.3 Technologies Used

A Retail Point of Sale (POS) system relies heavily on a Database Management System (DBMS) to store, organize, and manage the large volume of data generated during retail operations. The DBMS serves as the backbone of the POS software, ensuring data integrity, consistency, and security across all system functions.

Key DBMS concepts used in a POS system include:

- 1. Data Storage and Tables:** The DBMS stores data in structured tables such as Products, Sales, Employees, Customers, and Inventory. Each table contains records (rows) and fields (columns) representing specific attributes (e.g., product name, price, stock level).
- 2. Data Relationships:** POS systems use relational databases where tables are connected using keys. For example, a Sales table may reference a Customer and Product table using foreign keys, allowing the system to link transactions to specific customers and items.

- 3. Data Querying:** Structured Query Language (SQL) is used to query and manipulate data. POS systems use SQL queries to fetch sales reports, update inventory after a sale, or retrieve customer purchase history.
- 4. Transactions and Concurrency:** A POS system handles many transactions simultaneously. The DBMS ensures ACID properties (Atomicity, Consistency, Isolation, Durability) to maintain data accuracy during multiple operations, especially in multi-user environments.
- 5. Security and Access Control:** The DBMS manages user access, ensuring only authorized users (e.g., admin, cashier, manager) can view or modify specific data. This protects sensitive business and customer information.
- 6. Backup and Recovery:** To prevent data loss, POS systems use DBMS features for data backup and recovery, ensuring the system can be restored in case of failure.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Problem Description

In traditional retail environments, managing sales, inventory, billing, and customer data manually is time-consuming, error-prone, and inefficient. Small and medium-sized businesses often rely on outdated methods such as handwritten receipts or standalone billing systems, which do not offer real-time inventory updates, accurate sales tracking, or warranty management.

This lack of automation can lead to stock mismanagement, incorrect billing, delays in service, and poor customer satisfaction. Therefore, a centralized Retail Point of Sale (POS) System is required to simplify the process of adding products, handling purchases through a shopping cart, calculating bills, processing payments, updating inventory, and generating digital receipts – all while maintaining a proper database structure for smooth business operations.

2.2 Proposed Work

The proposed work for the Retail Point of Sale (POS) System aims to design and develop a robust, database-driven application that automates the core operations of a retail store. The system will include essential modules such as user authentication, customer and product management, inventory tracking, sales processing, payment handling, and reporting. The goal is to replace manual billing and stock management processes with a streamlined digital workflow to enhance speed, accuracy, and customer satisfaction.

This POS system will allow cashiers to scan products, retrieve item details from a centralized database, and generate bills efficiently. Customers can choose from multiple payment options such as cash, card, or UPI. After every transaction, stock levels will be updated in real time to prevent inventory mismatches. Administrators will have access to various analytical reports including daily sales, low-stock alerts, and best-selling products. The entire system will be built using a

modular approach, making it scalable and easy to maintain, with the possibility of integrating future enhancements like mobile app support and online order synchronization.

2.3 ER Diagram

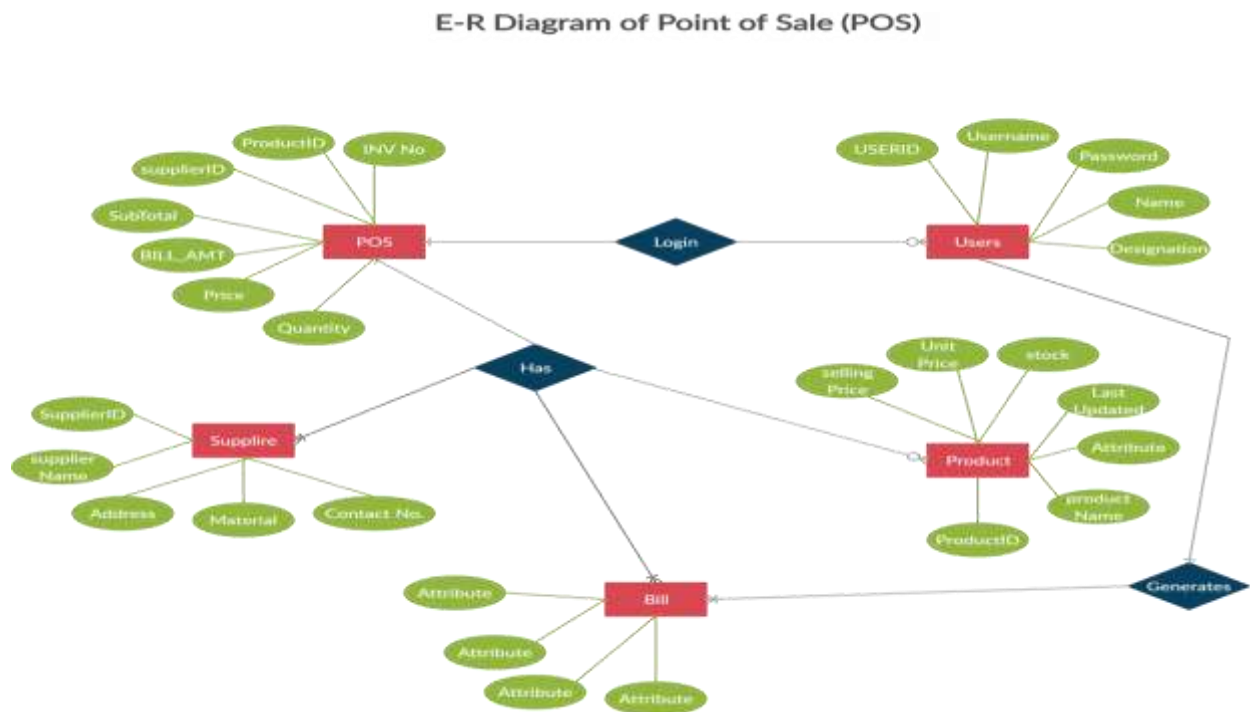


Figure 2.1: Retail POS System

2.4 ER DIAGRAM OVERVIEW

The E-R diagram for the Retail Point of Sale (POS) System shows the key entities involved in sales transactions. The central entity POS is connected to Users, Products, Suppliers, and Bills. Each user logs in to operate the system, and every sale involves selecting a product from the inventory, which is linked to its supplier. The transaction generates a bill and updates stock levels. Attributes like product price, quantity, and bill amount help manage sales and inventory. This diagram ensures efficient data flow and organization within the POS system.

CHAPTER 3

MODULE DESCRIPTION

3.1 USER MANAGEMENT MODULE

The User Management Module is crucial for securing and organizing access within the POS system. It allows authorized users to log in using their unique credentials and supports various roles such as Admin, Manager, and Cashier. Each role is associated with a specific set of permissions to ensure users only access functions relevant to their job responsibilities. This module not only protects sensitive data but also prevents unauthorized access to key areas of the system.

In addition to login and logout features, the module maintains a database of all users, storing information such as username, password (encrypted), role, and last login time. Admin users can add new users, assign roles, and deactivate users when needed. This structured approach improves accountability, as all user activities can be tracked, making it easier to audit system usage and detect any unusual activity.

3.2 PRODUCT MANAGEMENT MODULE

The Product Management Module handles everything related to products available for sale in the store. It allows the staff to add new products by entering attributes such as name, barcode, price, stock quantity, and category. Admins or authorized users can also update existing product details or remove obsolete products from the system, keeping the catalog current and accurate.

This module plays a vital role in both the billing and inventory systems, as it directly affects how products are scanned, displayed, and updated. Integration with barcode scanners ensures quick product identification at the time of sale, speeding up checkout.

Maintaining a clean and organized product database not only streamlines operations but also enhances reporting and stock management.

3.3 SALES & BILLING MODULE

The Sales & Billing Module forms the heart of the POS system by handling all real-time transactions. When a product is scanned, this module retrieves its details, calculates the price based on quantity, and applies any active discounts or promotions. It generates a subtotal, includes applicable taxes, and calculates the grand total. The module is designed for speed and accuracy, helping cashiers process purchases efficiently.

Once the bill is finalized, it can be printed or sent digitally to the customer. It also logs each transaction into the database for recordkeeping and future reporting. The Sales & Billing Module often works closely with the Payment and Inventory modules, updating stock levels and recording payment methods used. This integration ensures accurate inventory data and helps generate useful sales reports for business analysis.

3.4 INVENTORY MANAGEMENT MODULE

The Inventory Management Module monitors and updates product stock in real time. Each time a sale is processed, the stock count for that product is automatically reduced in the system. Similarly, when new stock arrives, it is added to the inventory via a purchase or restock entry. This continuous updating helps maintain an accurate picture of available products in the store.

The module also includes alert systems for low stock levels, allowing store managers to reorder items before they run out. Detailed inventory reports can be generated to assess fast-moving items, slow stock, or overstocked products. This helps improve stock rotation, avoid unnecessary purchases, and keep the supply chain running efficiently.

3.5 PAYMENT MODULE

The Payment Module handles all monetary transactions between the customer and the POS system. After billing is completed, this module offers various payment options such as cash, credit/debit card, and UPI. The cashier selects the customer's preferred method, enters the amount received, and the system either completes the transaction or calculates the change to be returned in case of cash payments.

It also records each payment with details like transaction time, amount, mode of payment, and reference number. These records help in reconciliation and reporting at the end of the day or during audits. The module may also support refunds or cancellations, ensuring flexibility and convenience in handling different transaction scenarios.

CHAPTER 4

IMPLEMENTATION & TEST CASES

4.1 TABLE STRUCTURES

Table 1: Customer

Column Name	Data Type	Description
customer_id	INT (PK)	Unique customer ID
name	VARCHAR(100)	Customer's full name
phone	VARCHAR(20)	Contact number
email	VARCHAR(100)	Email address
address	TEXT	Residential address

Table 2 : Product

Column Name	Data Type	Description
product_id	INT (PK)	Unique product ID
name	VARCHAR(100)	Product name
category	VARCHAR(100)	Category of product
price	DECIMAL(10,2)	Unit price
quantity_in_stock	INT	Quantity available in inventory
description	TEXT	Product details
warranty	VARCHAR(100)	Warranty period

Table 3 : Employee

Column Name	Data Type	Description
employee_id	INT (PK)	Unique employee ID
name	VARCHAR(100)	Full name
role	VARCHAR(50)	Role/position (e.g., cashier)
phone	VARCHAR(20)	Contact number
email	VARCHAR(100)	Email address

Table 4 : Supplier

Column Name	Data Type	Description
supplier_id	INT (PK)	Unique supplier ID
name	VARCHAR(100)	Supplier name
phone	VARCHAR(20)	Contact number
email	VARCHAR(100)	Email address
address	TEXT	Company address

Table 5 : Sales

Column Name	Data Type	Description
sale_id	INT (PK)	Unique sale transaction ID
customer_id	INT (FK)	Linked to <code>customer.customer_id</code>
employee_id	INT (FK)	Handled by which employee
sale_date	DATETIME	Date and time of the sale
total_amount	DECIMAL(10,2)	Total sale amount

Table 6 :Payment

Column Name	Data Type	Description
payment_id	INT (PK)	Unique payment ID
sale_id	INT (FK)	Linked to <code>sales.sale_id</code>
payment_type	VARCHAR(50)	Mode of payment (Cash, Card, UPI)
amount_paid	DECIMAL(10,2)	Amount paid
payment_date	DATETIME	Date and time of payment

Table 7 : Receipt

Column Name	Data Type	Description
receipt_id	INT (PK)	Unique receipt ID
sale_id	INT (FK)	Linked to sales.sale_id
receipt_date	DATETIME	Date receipt was generated
issued_by	VARCHAR(100)	Name or ID of employee

4.2 SQL QUERIES

1. CUSTOMER

```
CREATE TABLE customer (  
    customer_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    name VARCHAR(100),  
    phone VARCHAR(20),  
    email VARCHAR(100),  
    address TEXT  
);
```

2. PRODUCT

```
CREATE TABLE product (  
    product_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    name VARCHAR(100),  
    category VARCHAR(100),  
    price DECIMAL(10,2),  
    quantity_in_stock INT,  
    description TEXT,  
    warranty VARCHAR(100)  
);
```


3. EMPLOYEE

```
CREATE TABLE employee (  
    employee_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    name VARCHAR(100),  
    role VARCHAR(50),  
    phone VARCHAR(20),  
    email VARCHAR(100)  
);
```

4. SUPPLIER

```
CREATE TABLE supplier (  
    supplier_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    name VARCHAR(100),  
    phone VARCHAR(20),  
    email VARCHAR(100),  
    address TEXT  
);
```

5. SALES

```
CREATE TABLE sales (  
    sale_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    customer_id INT,  
    employee_id INT,  
    sale_date DATETIME,  
    total_amount DECIMAL(10,2),  
    FOREIGN KEY (customer_id)  
    REFERENCES customer(customer_id),  
    FOREIGN KEY (employee_id)  
    REFERENCES employee(employee_id)  
);
```

6. PAYMENT

```
CREATE TABLE payment (  
    payment_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    sale_id INT,  
    payment_type VARCHAR(50),  
    amount_paid DECIMAL(10,2),  
    payment_date DATETIME,  
    FOREIGN KEY (sale_id) REFERENCES  
    sales(sale_id)  
);
```

7. RECEIPT

```
CREATE TABLE receipt (  
    receipt_id INT PRIMARY KEY  
    AUTO_INCREMENT,  
    sale_id INT,  
    receipt_date DATETIME,  
    issued_by VARCHAR(100),  
    FOREIGN KEY (sale_id) REFERENCES  
    sales(sale_id)  
);
```

4.3 TEST CASES & RESULTS

Test Case ID	Module	Test Description	Input Data	Expected Output	Status
TC01	Product Add	Add a new product to inventory	Name, Price, Quantity, Warranty	Product added to product table	Pass
TC02	Product Add	Add product with missing name	Price, Quantity, Warranty only	Error: Name required	Pass
TC03	Add to Cart	Add available product to cart	Product ID with available quantity	Product added to cart	Pass
TC04	Add to Cart	Add product with zero stock	Product ID with 0 stock	Error: Out of stock	Pass
TC05	Billing	Calculate total for cart items	Multiple items in cart	Total = sum of item prices	Pass
TC06	Checkout	Checkout cart and generate receipt	Cart with items	Receipt generated and cart cleared	Pass
TC07	Payment	Process payment for a sale	Sale ID, payment type, amount	Payment recorded in payment table	Pass
TC08	Receipt View	View generated receipt after checkout	Sale ID	Shows list of items with total price	Pass
TC09	Inventory Update	Ensure stock reduces after adding to cart	Add to cart 1 unit	Product quantity reduced by 1	Pass
TC10	Data Validation	Add product with negative price or quantity	Negative values	Error: Invalid input	Pass

Table 4.4: Test Cases & Results

CHAPTER 5

CONCLUSION & FUTURE SCOPE

5.1 CONCLUSION

The Retail Point of Sale (POS) System is a powerful and reliable solution developed to handle the essential operations of a retail store. It integrates multiple functionalities such as user login, product and inventory management, billing, payment processing, and report generation. By automating these processes, the system reduces manual effort, increases transaction speed, and ensures data accuracy. It allows cashiers and admins to perform their roles efficiently through a secure, role-based login system. The use of a centralized database ensures that all records—such as sales, payments, and inventory updates—are stored securely and can be accessed in real time.

This system not only simplifies daily business operations but also provides valuable insights through reports and analytics, helping store owners make better decisions. With features like receipt generation, multi-payment support, and low-stock alerts, it improves customer satisfaction and inventory control.

5.2 FUTURE SCOPE

The Retail POS System has strong potential for future enhancements to meet growing business needs and improve user experience. One major improvement could be integrating real-time cloud-based data storage, allowing multiple store branches to sync and share data instantly. Mobile app integration for sales tracking, stock alerts, or remote management could make the system more flexible and accessible. Additionally, adding support for QR code-based billing and loyalty rewards systems would enhance customer satisfaction and engagement. Another important area of development is the integration of AI and machine learning to analyze customer behavior, predict sales trends, and optimize inventory management.

Overall, the POS system can be continuously improved to support larger businesses, enhance automation, and keep up with evolving retail technologies.

APPENDIX A

(SOURCE CODE)

```
<!DOCTYPE html>
<html>
<head>
<title>WELCOME TO RETAIL POS SYSTEM</title>
<style>
body {
font-family: 'Segoe UI', sans-serif; background-color: #f0f2f5;
text-align: center; margin: 0;
padding: 0;
}
h1 {
background-color: #4CAF50; color: white;
padding: 20px; margin: 0;
}
.container {
margin-top: 30px; display: flex;
flex-wrap: wrap; justify-content: center;
}
.card {
width: 220px; margin: 15px; background: white; border-radius: 10px;
box-shadow: 0 2px 8px rgba(0,0,0,0.2); padding: 15px;
width: 220px; margin: 15px; background: white; border-radius: 10px;
box-shadow: 0 2px 8px rgba(0,0,0,0.2); padding: 15px;
}
.card img { width: 100%; height: 130px;
object-fit: contain;
}
.card a {
display: block; margin-top: 10px;
```

```

padding: 10px;
background-color: #4CAF50; color: white;
text-decoration: none; border-radius: 5px; font-weight: bold;
}
.card a:hover {
background-color: #388e3c;
}
</style>
</head>
<body>
<h1>WELCOME TO RETAIL POS SYSTEM</h1>
<div class="container">
<div class="card">

<a href="insert_customer.php">Add Customer</a>
<a href="view_customers.php">View Customers</a>
</div>
<div class="card">

<a href="insert_product.php">Add Product</a>
<a href="view_products.php">View Products</a>
</div>
<div class="card">

<a href="insert_employee.php">Add Employee</a>
<a href="view_employees.php">View Employees</a>
</div>
<div class="card">

<a href="insert_supplier.php">Add Supplier</a>
<a href="view_suppliers.php">View Suppliers</a>
</div>
<div class="card">

```

```


<a href="insert_sales.php">Add Sale</a>
<a href="view_sales.php">View Sales</a>
</div>
<div class="card">

<a href="insert_receipt.php">Add Receipt</a>
<a href="view_receipts.php">View Receipts</a>
</div>
</div>
</body>
</html>

```

```
SELECT * FROM customers
```

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$name = $_POST["name"];
$phone = $_POST["phone"];
$email = $_POST["email"];
$stmt = $conn->prepare("INSERT INTO customer
VALUES (?, ?, ?)");
$stmt->bind_param("sss", $name, $phone, $email); if ($stmt->execute()) {
echo "Customer inserted successfully!";
} else {
echo " Error: " . $stmt->error;
(name, phone, email)
}
$conn->close();
}
?>

```

```

<h2>Add Customer</h2>
<form method="POST">
Name: <input type="text" name="name" required><br><br> Phone: <input
type="text" name="phone" required><br><br> Email: <input type="email"
name="email" required><br><br>
<input type="submit" value="Add Customer">
</form>
SELECT * FROM employee
<!DOCTYPE html>
<html>
<head>
<title>Insert Employee</title>
</head>
<body>
<h2>Add Employee</h2>
<form method="post">
Name: <input type="text" name="name" required><br><br> Position: <input
type="text" name="position" required><br><br>
Salary: <input type="number" name="salary" step="0.01" required><br><br>
<input type="submit" name="submit" value="Add Employee">
</form>
<?php
if (isset($_POST['submit'])) {
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$name = $_POST['name'];
$position = $_POST['position'];
$salary = $_POST['salary'];
$stmt = $conn->prepare("INSERT INTO employee (name, position, salary)
VALUES (?, ?, ?)");
$stmt->bind_param("ssd", $name, $position, $salary);
if ($stmt->execute()) {

```



```

echo "<p style='color:green;'>Employee inserted successfully!</p>";
} else {
echo "<p style='color:red;'>Error: " . $stmt->error . "</p>";
}
$stmt->close();
$conn->close();
}
?>
</body>
</html>
SELECT * FROM payment
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$customers = $conn->query("SELECT id, name FROM customer"); if
($_SERVER["REQUEST_METHOD"] == "POST") {
$customer_id = $_POST['customer_id'];
$amount = $_POST['amount'];
$payment_method = $_POST['payment_method'];
$stmt=$conn->prepare("INSERT INTO payment (customer_id,
payment_method) VALUES (?, ?, ?)");
$stmt->bind_param("ids", $customer_id, $amount, $payment_method); if
($stmt->execute()) {
echo "Payment inserted successfully!";
} else {
echo " Error: " . $stmt->error;
amount,
}
$stmt->close();
}
?>
<h2>Insert Payment</h2>

```

```

<form method="POST">
Customer:
<select name="customer_id" required>
<option value="">Select Customer</option>
<?php while ($row = $customers->fetch_assoc()): ?>
<option value="<?= $row['id'] ?>"><?= $row['name'] ?></option>
<?php endwhile; ?>
</select><br><br>
Amount: <input type="number" name="amount" step="0.01"
required><br><br> Payment Method:
<select name="payment_method" required>
<option value="Cash">Cash</option>
<option value="Card">Card</option>
<option value="UPI">UPI</option>
</select><br><br>
<input type="submit" value="Insert Payment">
</form>
SELECT * FROM product
?php
$conn = new mysqli("localhost", "root", "", "retail_pos");
if ($conn->connect_error) { die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
$name = $_POST['name'];
$price = $_POST['price'];
$quantity = $_POST['quantity'];
if (!empty($name) && !empty($price) && !empty($quantity)) {
$stmt = $conn->prepare("INSERT INTO product (name, price, quantity)
VALUES (?, ?, ?)");
$stmt->bind_param("sdi", $name, $price, $quantity);
if ($stmt->execute()) { echo "Product added successfully!"; } else { echo "Error:
" . $stmt->error; }
$stmt->close();
} else { echo "Please fill all fields!"; }

```

```

}
$conn->close();
?>
<form method="post" action="">
Name: <input type="text" name="name" required><br><br>
Price: <input type="number" step="0.01" name="price" required><br><br>
Quantity: <input type="number" name="quantity" required><br><br>
<input type="submit" value="Add Product">
</form>
SELECT * FROM receipt
<!DOCTYPE html>
<html>
<head>
<title>Insert Receipt</title>
</head>
<body>
<h2>Insert Receipt</h2>
<form method="POST" action="insert_receipt.php">
Customer ID: <input type="number" name="customer_id" required><br><br>
Payment Method: <input type="text" name="payment_method"
required><br><br>
Amount: <input type="number" step="0.01" name="amount"
required><br><br>
Date: <input type="date" name="date" required><br><br>
<input type="submit" value="Insert">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$customer_id = $_POST["customer_id"];
$payment_method = $_POST["payment_method"];

```

```

$amount = $_POST["amount"];
$date = $_POST["date"];
$sql = "INSERT INTO receipt (customer_id, payment_method, amount, date)
VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("isds", $customer_id, $payment_method, $amount, $date);
if ($stmt->execute()) {
    echo "Receipt inserted successfully.";
} else {
    echo "Error: " . $stmt->error;
}
$stmt->close();
$conn->close();
}
?>
</body>
</html>
SELECT * FROM sales
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$productResult = $conn->query("SELECT id, name, price FROM product"); if
($_SERVER["REQUEST_METHOD"] == "POST") {
    $product_id = $_POST['product_id'];
    $quantity = $_POST['quantity'];
    $stmt = $conn->prepare("SELECT price FROM product WHERE id=?");
    $stmt->bind_param("i", $product_id);
    $stmt->execute();
    $stmt->bind_result($price);
    $stmt->fetch();
    $stmt->close();
    $total_price = $price * $quantity;

```

```

$stmt = $conn->prepare("INSERT INTO sales (product_id, quantity,
total_price)
VALUES (?, ?, ?)");
$stmt->bind_param("iid", $product_id, $quantity, $total_price); if ($stmt-
>execute()) {
echo "Sale recorded successfully!";
} else {
echo "Error: " . $stmt->error;
}
$stmt->close();
}
?>

<h2>Record Sale</h2>
<form method="POST">
Product:
<select name="product_id" required>
<option value="">Select Product</option>
<?php
if ($productResult->num_rows > 0) {
while ($row = $productResult->fetch_assoc()) {
echo
?>
</select><br><br>
Quantity: <input type="number" name="quantity" min="1" required><br><br>
<input type="submit" value="Record Sale">
</form>
SELECT * FROM supplier
<!DOCTYPE html>
<html>
<head>
<title>Insert Supplier</title>
</head>
<body>
<h2>Add Supplier</h2>

```

```

<form method="post">
Name: <input type="text" name="name" required><br><br> Contact: <input
type="text" name="contact" required><br><br> Email: <input type="email"
name="email" required><br><br>
<input type="submit" name="submit" value="Add Supplier">
</form>
<?php
if (isset($_POST['submit'])) {
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);

}
$name = $_POST['name'];
$contact = $_POST['contact'];
$email = $_POST['email'];
$stmt = $conn->prepare("INSERT INTO supplier (name, contact, VALUES (?,
?, ?)");
$stmt->bind_param("sss", $name, $contact, $email);
if ($stmt->execute()) {
echo "<p style='color:green;'>Supplier inserted successfully!</p>";
} else {
echo "<p style='color:red;'>Error: " . $stmt->error . "</p>";
email)
}
$stmt->close();
$conn->close();
}
?>
</body>
</html>
<!DOCTYPE html>
<html>
<head>

```

```

<title>Login - POS System</title>
<style>
body {
font-family: Arial, sans-serif; background: #e0e0e0; display: flex;
justify-content: center;
align-items: center; height: 100vh;
}
.login-box { background: white; padding: 30px;
border-radius: 12px;
box-shadow: 0 0 10px #aaa; width: 300px;
}
h2 {
text-align: center; margin-bottom: 20px;
}
input[type="text"], input[type="password"] { width: 100%;
padding: 10px; margin-bottom: 15px; border: 1px solid #ccc; border-radius:
6px;
}
input[type="submit"] { width: 100%; padding: 10px;
background: #4CAF50; border: none;
color: white; border-radius: 6px;
cursor: pointer;
}
input[type="submit"]:hover { background: #45a049;
}
</style>
</head>
<body>
<div class="login-box">
<h2>Login</h2>
<form action="index.php" method="post">
<input type="text" name="username" placeholder="Enter Username" required>
<input type="password" name="password" placeholder="Enter Password"
required>

```

```

<input type="submit" value="Login">
</form>
</div>
</body>
</html>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$result = $conn->query("SELECT id, name, phone, email FROM customer"); if
($result->num_rows > 0) {
echo "<h2>Customer List</h2>";
echo "<table border='1' cellpadding='10'>
<tr><th>ID</th><th>Name</th><th>Phone</th><th>Email</th></tr>"; while
($row = $result->fetch_assoc()) {
echo "<tr>
<td>{$row['id']}</td>
<td>{$row['name']}</td>
<td>{$row['phone']}</td>
<td>{$row['email']}</td>
</tr>";
}
echo "</table>";
} else {
echo "No customers found.";
}
$conn->close();
?>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos");
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

```



```

$result = $conn->query("SELECT * FROM employee"); echo "<h2>Employee
List</h2>";
echo "<table border='1' cellpadding='10'>";
echo
"<tr><th>ID</th><th>Name</th><th>Position</th><th>Salary</th></tr>";
while ($row = $result->fetch_assoc()) {
echo "<tr>
<td>{$row['id']}</td>
<td>{$row['name']}</td>
<td>{$row['position']}</td>
<td>{$row['salary']}</td>
</tr>";
}
echo "</table>";
$conn->close();
?>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos");
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql="SELECT payment.id, customer.namecustomer_name,
payment_method, payment_date
FROM payment
JOIN customer ON payment.customer_id = customer.id ORDER BY
payment_date DESC";
$result = $conn->query($sql);
echo "<h2>Payment Records</h2>"; if ($result->num_rows > 0) {
echo "<table border='1' cellpadding='10'>
<tr>
<th>ID</th>
<th>Customer</th>
<th>Amount</th>
<th>Method</th>

```

```

amount,
<th>Date</th>
</tr>";
while ($row = $result->fetch_assoc()) { echo "<tr>
<td>{$row['id']}</td>
<td>{$row['customer_name']}</td>
<td>₹{$row['amount']}</td>
<td>{$row['payment_method']}</td>
<td>{$row['payment_date']}</td>
</tr>";
}
echo "</table>";
} else {
echo "No payment records found.";
}
?>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$result = $conn->query("SELECT * FROM product");
echo "<h2>Product List</h2>";
echo "<table border='1' cellpadding='10'>";
echo "<tr><th>ID</th><th>Name</th><th>Price</th><th>Quantity</th></tr>";
while ($row = $result->fetch_assoc()) { echo "<tr>
<td>{$row['id']}</td>
<td>{$row['name']}</td>
<td>{$row['price']}</td>
<td>{$row['quantity']}</td>
</tr>";
}
echo "</table>";

```

```

$conn->close();
?>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "retail_pos"; // or store_management
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT r.id, c.name as customer_name, r.amount, r.date FROM receipt
r
JOIN customer c ON r.customer_id = c.id";
$result = $conn->query($sql); echo "<h2>Receipt List</h2>"; if ($result-
>num_rows > 0) {
    echo "<table border='1' cellpadding='10'>
    <tr>
    <th>ID</th>
    <th>Customer</th>
    <th>Amount</th>
    <th>Date</th>
    </tr>";
    while($row = $result->fetch_assoc()) { echo "<tr>
    <td>{$row['id']}</td>
    <td>{$row['customer_name']}</td>
    <td>{$row['amount']}</td>
    <td>{$row['date']}</td>
    </tr>";
    }
    echo "</table>";
    } else {
    echo "No receipts found.";
    }
}

```

```

$conn->close();
?>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql="SELECT sales.id,product.nameAS sales.sale_date, sales.total_price
FROM sales
JOIN product ON sales.product_id = product.id ORDER BY sales.sale_date
DESC";
$result = $conn->query($sql);
product_name, sales.quantity,
if ($result->num_rows > 0) {
echo "<h2>Sales Records</h2>";
echo "<table border='1' cellpadding='10'>
<tr>
<th>ID</th>
<th>Product</th>
<th>Quantity</th>
<th>Sale Date</th>
<th>Total Price</th>
</tr>";
while ($row = $result->fetch_assoc()) { echo "<tr>
<td>{$row['id']}</td>
<td>{$row['product_name']}</td>
<td>{$row['quantity']}</td>
<td>{$row['sale_date']}</td>
<td>₹{$row['total_price']}</td>
</tr>";
}
echo "</table>";
} else {
echo "No sales found.";

```

```

}
$conn->close();
?>
<?php
$conn = new mysqli("localhost", "root", "", "retail_pos"); if ($conn-
>connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$result = $conn->query("SELECT * FROM supplier"); echo "<h2>Supplier
List</h2>";
echo "<table border='1' cellpadding='10'>";
echo "<tr><th>ID</th><th>Name</th><th>Contact</th><th>Email</th></tr>";
while ($row = $result->fetch_assoc()) {
echo "<tr>
<td>{$row['id']}</td>
<td>{$row['name']}</td>
<td>{$row['contact']}</td>
<td>{$row['email']}</td>
</tr>";
}
echo "</table>";
$conn->close();
?>

```

APPENDIX B (SCREENSHOTS)

STEP 1: LOGIN PAGE

Login

Login

Login

Login

STEP 2: INDEX PAGE

WELCOME TO RETAIL POS SYSTEM

Customer

Add Customer

View Customers

Product

Add Product

View Products

Employee

Add Employee

View Employees

Supplier

Add Supplier

View Suppliers

Sales

Add Sale

View Sales

Receipts

Add Receipt

View Receipts

STEP 3: CUSTOMER

Customer

Add Customer

View Customers

Add Customer

Name:

Phone:


Email:

Add Customer

Customer List

ID	Name	Phone	Email
1	Meera. B	9655605336	b2332@kroc.ac.in
2	dhruva	9655605336	b2332@kroc.ac.in
3	Meera. B	9655605336	a2321@kroc.ac.in
4	Meera. B	9655605336	a2321@kroc.ac.in
5	dhruva	9655605336	a2314@kroc.ac.in
6	MEERA B	09655605336	meerabas.com@gmail.com

STEP 4: PRODUCT

 **Product**

Add Product

View Products

Product List

ID	Name	Price	Quantity
1	pen	10.00	100
2	pencil	5.00	20
3	pen	22.00	12
4	pen	22.00	12
5	remote	30.00	10

Name:

Price:

Quantity:

Add Product

STEP 5: EMPLOYEE

 **Employee**

Add Employee

View Employees

Add Employee

Name:

Position:


Salary:

Add Employee

Employee List

ID	Name	Position	Salary
1	Meera B	manager	20000.00
2	Meera B	manager	20000.00
3	Meera B	manager	20000.00
4	Meera B	manager	20000.00
5	dhruva	manager	20000.00
6	MEERA BASKAR	manager	20000.00

STEP 6 : SUPPLIER

 **Supplier**

Add Supplier

View Suppliers

Add Supplier

Name:

Contact:


Email:

Add Supplier

Supplier List

ID	Name	Contact	Email
1	meera	9876543212	a2334@kroce.ac.in
2	dhruva	9876543212	b2332@kroce.ac.in
3	motu	7865432198	abc123@gmail.com

STEP 7: SALES

Sales

Add Sale

View Sales

Record Sale

Product: pencil - ₹5.00

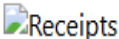
Quantity: 3

Record Sale

Sales Records

ID	Product	Quantity	Sale Date	Total Price
3	pencil	3	2025-05-18 21:29:25	₹15.00
2	pencil	23	2025-05-18 17:20:27	₹115.00
1	pen	25	2025-05-18 17:20:15	₹250.00

STEP 8: RECEIPTS

Receipts

Add Receipt

View Receipts

Insert Receipt

Customer ID: 1

Payment Method: cash

Amount: 2000

Date: 05/18/2025

Insert

Receipt List

ID	Customer	Amount	Date
2	Meeva R	20000.00	2025-05-18
3	dhanya	2000.00	2025-05-18
4	Meeva B	20000.00	2025-05-18
5	dhanya	2000.00	2025-05-18
6	Meeva B	2000.00	2025-05-18
7	dhanya	12300.00	2025-05-18
8	Meeva B	2000.00	2025-05-18

REFERENCES

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.
2. Laudon, K. C., & Laudon, J. P. (2019). *Management Information Systems: Managing the Digital Firm* (16th ed.). Pearson.
3. Ullman, J. D., & Widom, J. (2008). *A First Course in Database Systems* (3rd ed.). Pearson Education.
4. <https://www.geeksforgeeks.org/dbms/> – GeeksforGeeks DBMS Tutorials
5. <https://www.tutorialspoint.com/dbms/index.htm> – Tutorialspoint DBMS Guide
6. <https://developer.mozilla.org/> – Mozilla Developer Network (for front-end integration references)
7. <https://www.oracle.com/database/> – Oracle Documentation for SQL and DBMS
8. <https://www.postgresql.org/docs/> – PostgreSQL Official Documentation
9. POS System Workflow Diagrams – <https://creatly.com/>
10. <https://www.softwareadvice.com/resources/what-is-a-pos-system/>