

GROCERY MANAGEMENT SYSTEM



A Mini Project Report

Submitted by

NAVEEN RAJ P

241901070

SANJAY V

241901099

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

AND CYBER SECURITY

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI,

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “**GROCERY MANAGEMENT SYSTEM**” is the bonafide work of “**NAVEEN RAJ P (241901070), SANJAY V (241901099)**” who carried out the project work under my supervision.

Ms. R. Rupmala

ASSISTANT PROFESSOR

Department of Computer Science

Engineering And Cyber Security

Rajalakshmi Engineering College

This mini project report is submitted for the viva voce examination to be held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the mini project report **Grocery Management System**, submitted as part of the curriculum requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work carried out by us under the supervision of **Ms.R.Rupmala, Assistant Professor, Department of Computer Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.**

This submission represents our ideas in our own words, and wherever ideas or words of others have been used, we have duly cited and referenced the original sources with accuracy and completeness. We also declare that we have followed the principles of academic honesty and integrity and have not misrepresented or fabricated any data, ideas, facts, or sources in preparing this report.

We understand that any violation of the above statements will result in disciplinary action by the institute and/or the University and may also invite penal action from the respective copyright holders or authors whose works have been used without proper acknowledgment.

We further certify that this report **has not previously been submitted** for the award of any degree, diploma, or similar title at any other University.

Rajalakshmi Engineering College,

Chennai

November 2025

NAVEEN RAJ P

SANJAY V

ABSTRACT

The **Grocery Management System** is a robust desktop application, built using **Java Swing** for the intuitive graphical user interface and backed by **MySQL** for reliable data management, designed to fully automate and streamline daily grocery store operations. This comprehensive system replaces manual record-keeping by providing key functionalities like performing **CRUD operations** on detailed **product inventory** (including tracking stock, pricing, and reorder levels), processing quick and accurate **sales transactions** with automated **bill generation**, managing customer records, and producing essential **sales and inventory reports**, ultimately improving data accuracy, boosting staff productivity, and offering real-time operational insights for effective business decision-making.

KEYWORDS: JavaSwing, MySQL, JDBC, Inventory Management, Product Tracking, Billing System, Sales Management, Customer Management, CRUD Operations, Stock Monitoring, Data Security, User Interface Design

ACKNOWLEDGEMENT

We like to convey our sincere appreciation to all who have supported and mentored us during the successful completion of this project work. We express our profound gratitude to **Mr. Benedict J.N.**, Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project. We express our profound gratitude to **Ms. R. Rupmala**, Assistant Professor in the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for her unwavering support, invaluable insights, and collaboration throughout our endeavor. We would like to convey our gratitude to our faculty members and colleagues for their valuable feedback and encouragement. We express our gratitude to our families and friends for their steadfast support, patience, and encouragement, which were instrumental in the effective execution of this seminar.

NAVEENRAJ P

SANJAY V

TABLE OF CONTENTS	PAGE
DECLARATION	i
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Aim and Objectives	1
1.3 Scokpe of the Project	1
2 SYSTEM ARCHITECTURE	2
3 SYSTEM SPECIFICATIONS	3
3.1 Hardware Specifications	3
3.2 Software Specifications	3
4 MODULE DESCRIPTION	4
4.1 Product Management Module	4
4.2 Customer Management Module	4
4.3 Sales and Billing Module	4
4.4 ER Diagram	5
4.5 Database Schema	6
5 CODING	8
5.1 Key Java Implementation Snippets	8
6 SCREENSHOTS	11
7 CONCLUSION AND FUTURE ENHANCEMENT	
REFERENCES	14

LIST OF FIGURES

Figure No.	Title	Page
2.1	SYSTEM ARCHITECTURE	2
4.1	ER DIAGRAM	5
6.1	INTRODUCTION PAGE	14
6.2	PRODUCT MANAGEMENT	15
6.3	CUSTOMER MANAGEMENT	15
6.4	SALES AND BILLING	16

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In traditional grocery stores, operations are often handled manually, leading to errors in billing, stock maintenance, and record management. There is a lack of real-time inventory tracking and customer data management. The proposed system aims to overcome these limitations through automation and digital record-keeping.

1.2 Aim and Objectives

Aim: To develop an efficient and user-friendly software solution to manage grocery store operations.

Objectives:

- Simplify product, customer, and sales management.
- Automate billing and stock updates.
- Provide accurate and up-to-date records.
- Reduce manual errors and operational time.
- Improve the reliability of data storage using MySQL.

1.3 Scope of the Project

The system can be used in small and medium-sized grocery stores for managing products, sales, and customers. It supports data storage, search operations, and report generation. The architecture allows for scalability and can be extended to include online ordering or vendor management in the future.

CHAPTER 2

SYSTEM ARCHITECTURE

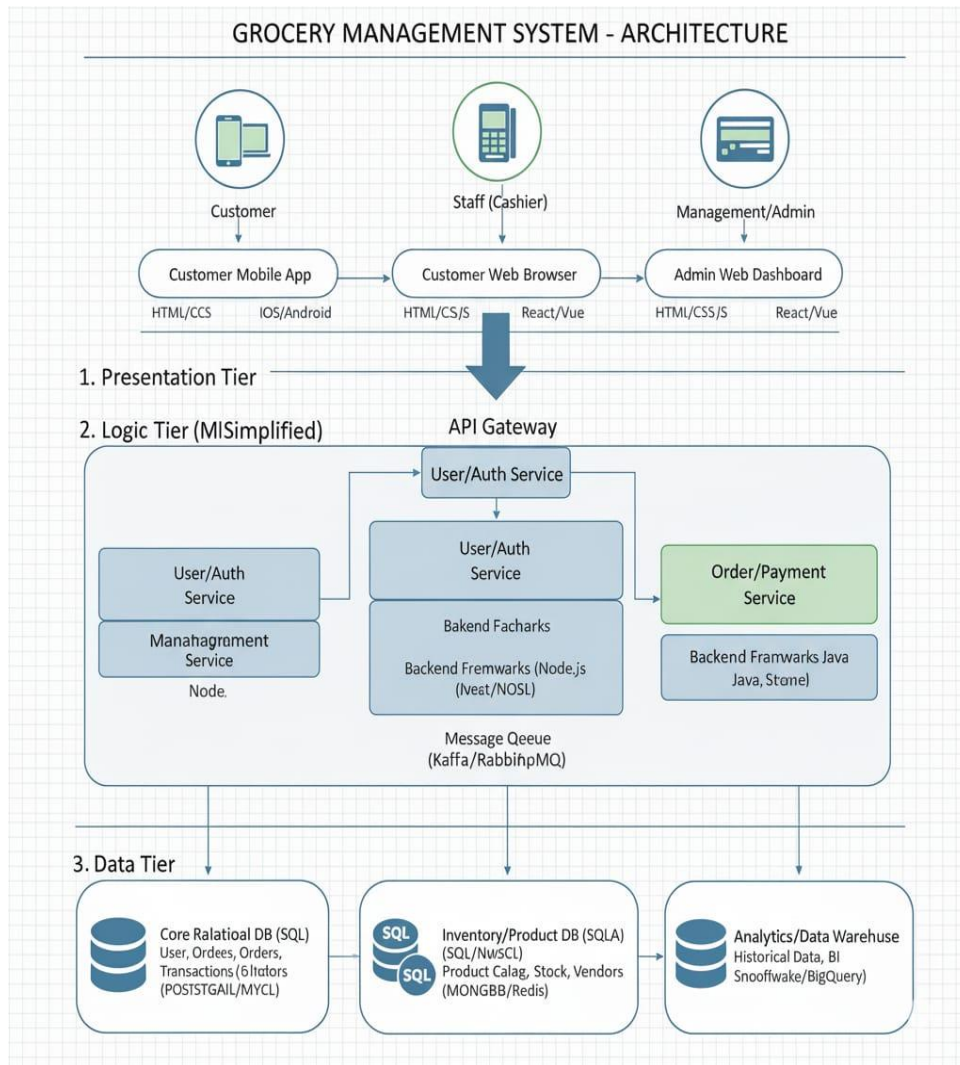


Fig 2.1 System Architecture

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 Hardware Requirements

- Processor : Intel Core i3 or above
- RAM : Minimum 4 GB
- Hard Disk : 500 MB free space
- Display : 1024 × 768 resolution or higher

3.2 Software Requirements

- Operating System : Windows 10/11
- Frontend : Java (Swing)
- Backend : MySQL Database
- IDE : IntelliJ IDEA Community Edition
- Connector : MySQL Connector/J

CHAPTER 4

MODULE DESCRIPTION

The project is divided into functional modules to simplify development and debugging. Each module handles a specific feature of the system, ensuring modularity and maintainability.

4.1 Product Management Module

This module handles product operations such as adding, updating, deleting, and displaying products. It connects directly to the products table in MySQL and automatically updates stock after sales.

4.2 Customer Management Module

This module stores and manages customer data including name, contact number, and email ID. It is linked with the sales module to track purchases made by each customer.

4.3 Sales and Billing Module

This module processes sales transactions and generates bills. It calculates totals automatically and updates product quantities in the database after every transaction.

4.4 ER Diagram

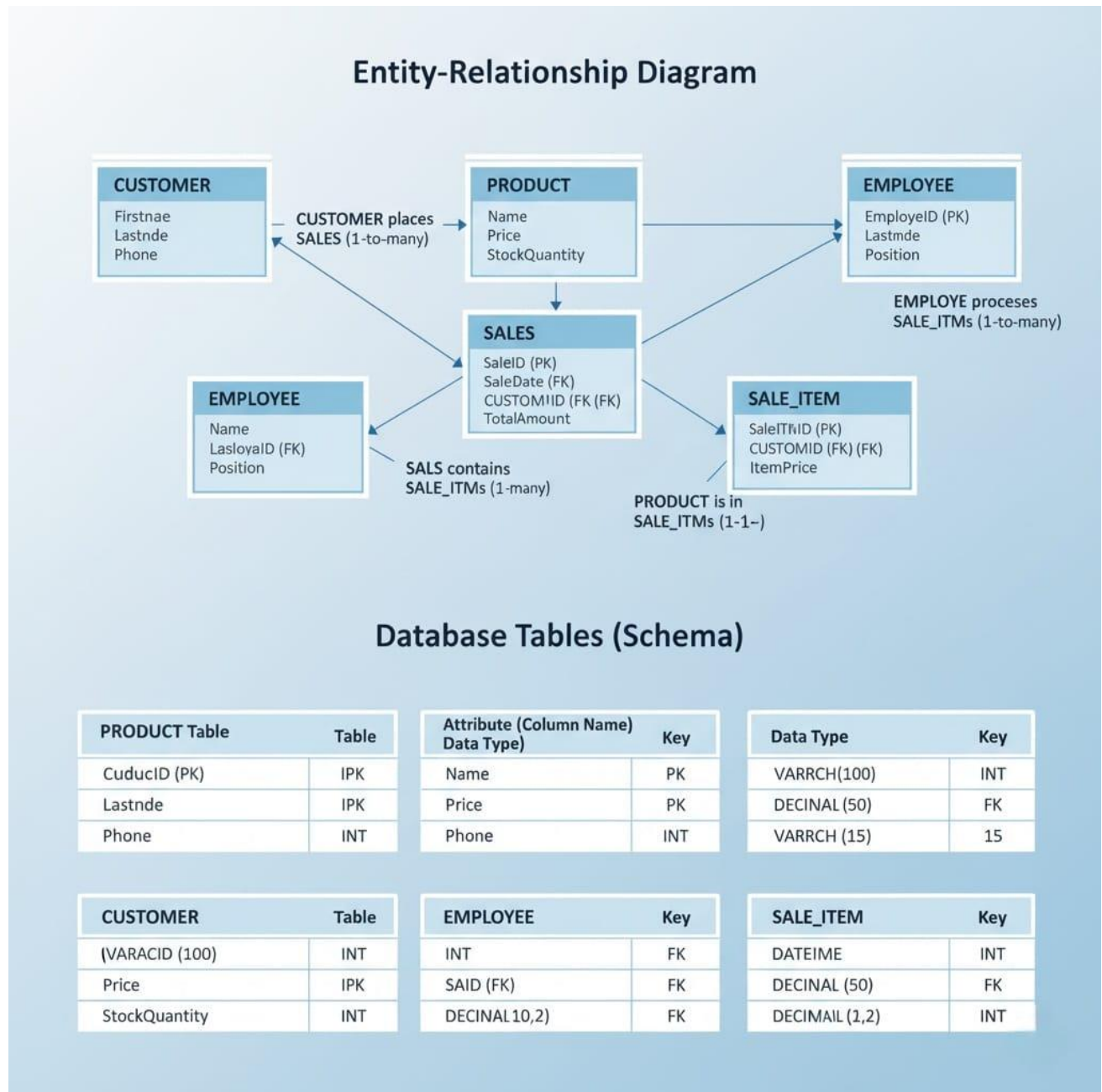


Fig 4.1 ER Diagram

4.5 Database Schema

The database structure for the Grocery Management System is designed to efficiently manage product inventory, customer information, sales transactions, and item-wise billing. It ensures referential integrity, modularity, and scalability for institutional or commercial retail use. The system is organized into four main tables: **products**, **customers**, **sales**, and **sale_items**. Each table is connected using primary keys and foreign keys, enabling optimized searching, faster joins, and consistent relational behavior.

Entity Descriptions:

Products: Stores all product-related information such as product name, unit price, and available stock quantity. Each product is uniquely identified using a **product_id**.

Customers: Contains customer details including their name, mobile number, and email address. Each customer is identified by a unique **customer_id**.

Sales: Represents each billing transaction made at the grocery store. It links every sale to a specific customer and records details such as sale date and total bill amount. Each sale is identified by a unique **sale_id**.

Sale item: Stores item-wise details for every sale. Each record represents a product purchased in a particular sale, along with quantity and subtotal. This table connects **sales** and **products** through foreign keys, ensuring accurate billing and stock updates.

Relational Schema

Field Name	Data Type	Constraints	Description
customer_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique customer ID
customer_name	VARCHAR(100)	NOT NULL	Name of the customer
phone	VARCHAR(15)	NULL	Contact number
email	VARCHAR(100)	NULL	Email ID

Table 1: Products Table

Field Name	Data Type	Constraints	Description
customer_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique customer ID
customer_name	VARCHAR(100)	NOT NULL	Name of the customer
phone	VARCHAR(15)	NULL	Contact number
email	VARCHAR(100)	NULL	Email ID

Table 2: Customers Table

Field Name	Data Type	Constraints	Description
sale_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique sale ID
customer_id	INT	FOREIGN KEY	References customers table
sale_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date & time of sale
total_amount	DECIMAL(10,2)	NULL	Total bill amount

Table 3: Sales Table

CHAPTER 5

CODING

5.1.Key Java Implementation Snippets

5.1.1.MAINDASHBOARD

```
package ui;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MainDashboard extends JFrame {
    public MainDashboard() {
        setTitle("Grocery Management System - Dashboard");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4, 1, 10, 10));
        JLabel title = new JLabel("GROCERY MANAGEMENT SYSTEM", JLabel.CENTER);
        title.setFont(new Font("Arial", Font.BOLD, 20));
        add(title);
        JButton productBtn = new JButton("🛒 Manage Products");
        JButton customerBtn = new JButton("👤 Manage Customers");
        JButton salesBtn = new JButton("💰 Sales & Billing");
        JButton exitBtn = new JButton("✕ Exit");
        productBtn.setFont(new Font("Arial", Font.PLAIN, 16));
        customerBtn.setFont(new Font("Arial", Font.PLAIN, 16));
        salesBtn.setFont(new Font("Arial", Font.PLAIN, 16));
        exitBtn.setFont(new Font("Arial", Font.PLAIN, 16));
        productBtn.addActionListener(e -> new ProductFrame());
        customerBtn.addActionListener(e -> new CustomerFrame());
        salesBtn.addActionListener(e -> new SalesFrame());
        exitBtn.addActionListener(e -> System.exit(0));
        add(productBtn);
        add(customerBtn);
        add(salesBtn);
        add(exitBtn);
        setLocationRelativeTo(null); // center window
        setVisible(true);
    }
    public static void main(String[] args) {
        new MainDashboard();
    }
}
```

5.1.2.PRODUCT FRAME

```
package ui;
import javax.swing.*;
import dao.ProductDAO;
import model.Product;
import java.awt.*;
import java.awt.event.*;
import java.util.List;
public class ProductFrame extends JFrame {
    JTextField nameField, categoryField, qtyField, priceField;
    JTextArea displayArea;
    ProductDAO dao = new ProductDAO();
    public ProductFrame() {
        setTitle("Product Management");
        setSize(500, 400);
        setLayout(new FlowLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        add(new JLabel("Name:"));
        nameField = new JTextField(10);
        add(nameField);
        add(new JLabel("Category:"));
        categoryField = new JTextField(10);
        add(categoryField);
        add(new JLabel("Quantity:"));
        qtyField = new JTextField(5);
        add(qtyField);
        add(new JLabel("Price:"));
        priceField = new JTextField(7);
        add(priceField);
        JButton addBtn = new JButton("Add Product");
        JButton viewBtn = new JButton("View All");
        displayArea = new JTextArea(10, 40);
        addBtn.addActionListener(e -> addProduct());
        viewBtn.addActionListener(e -> viewProducts());
        add(addBtn);
        add(viewBtn);
        add(new JScrollPane(displayArea));
        setVisible(true);
    }
    private void addProduct() {
        String name = nameField.getText();
        String cat = categoryField.getText();
        int qty = Integer.parseInt(qtyField.getText());
        double price = Double.parseDouble(priceField.getText());
        dao.addProduct(new Product(name, cat, qty, price));
        JOptionPane.showMessageDialog(this, "Product added!");
    }
}
```



```

    }
    private void viewProducts() {
        List<Product> list = dao.getAllProducts();
        displayArea.setText("");
        for (Product p : list) {
            displayArea.append(p.getProductId() + " - " + p.getName() +
                " (" + p.getCategory() + "), Qty: " + p.getQuantity() +
                ", Price: ₹" + p.getPrice() + "\n");
        }
    }
    public static void main(String[] args) {
        new ProductFrame();
    }
}

```

CHAPTER 6

SCREEN SHOTS

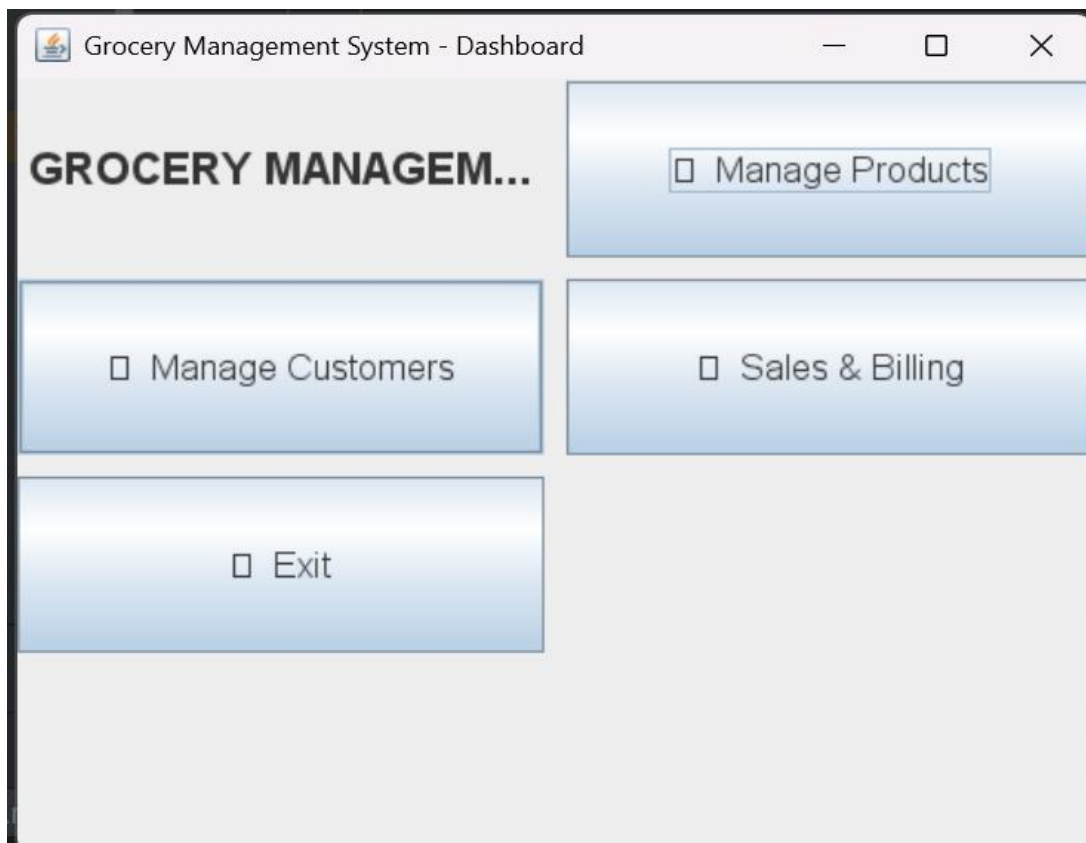
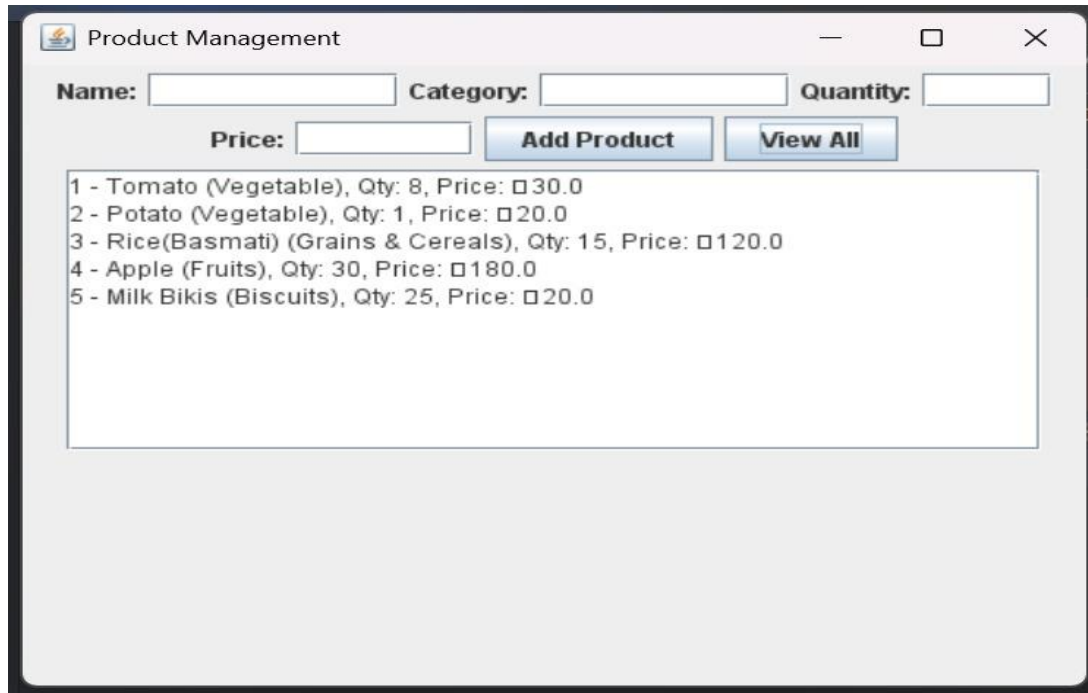


Fig 6.1 Introduction page



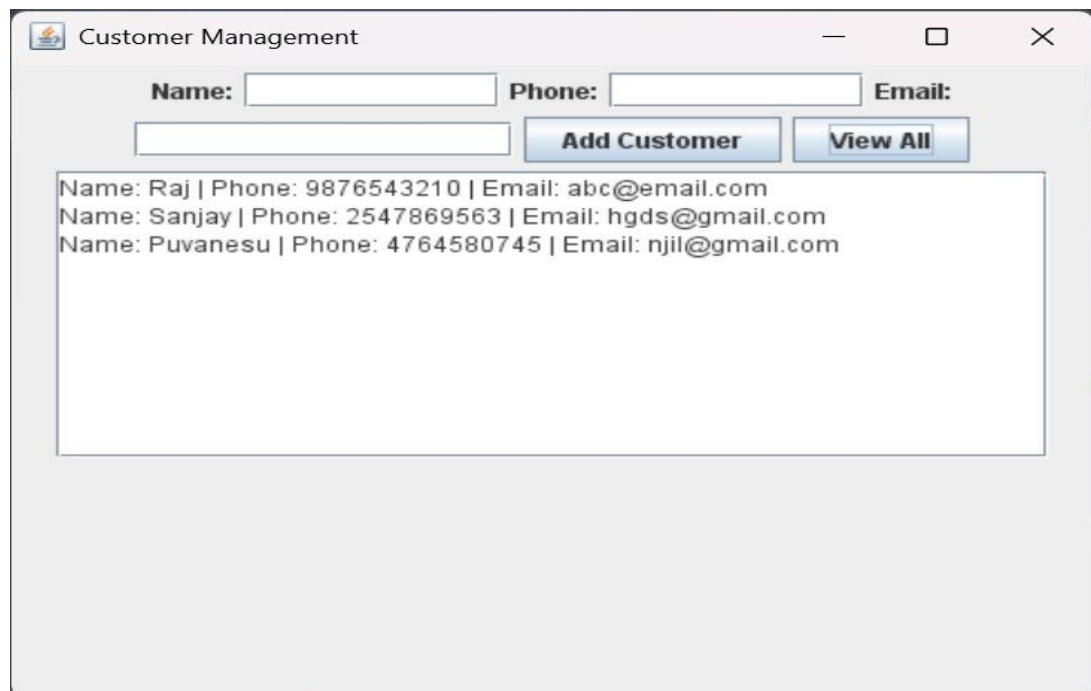
Product Management

Name: Category: Quantity:

Price:

1 - Tomato (Vegetable), Qty: 8, Price: ₹30.0
 2 - Potato (Vegetable), Qty: 1, Price: ₹20.0
 3 - Rice(Basmati) (Grains & Cereals), Qty: 15, Price: ₹120.0
 4 - Apple (Fruits), Qty: 30, Price: ₹180.0
 5 - Milk Bikis (Biscuits), Qty: 25, Price: ₹20.0

Fig 6.2 Product Management

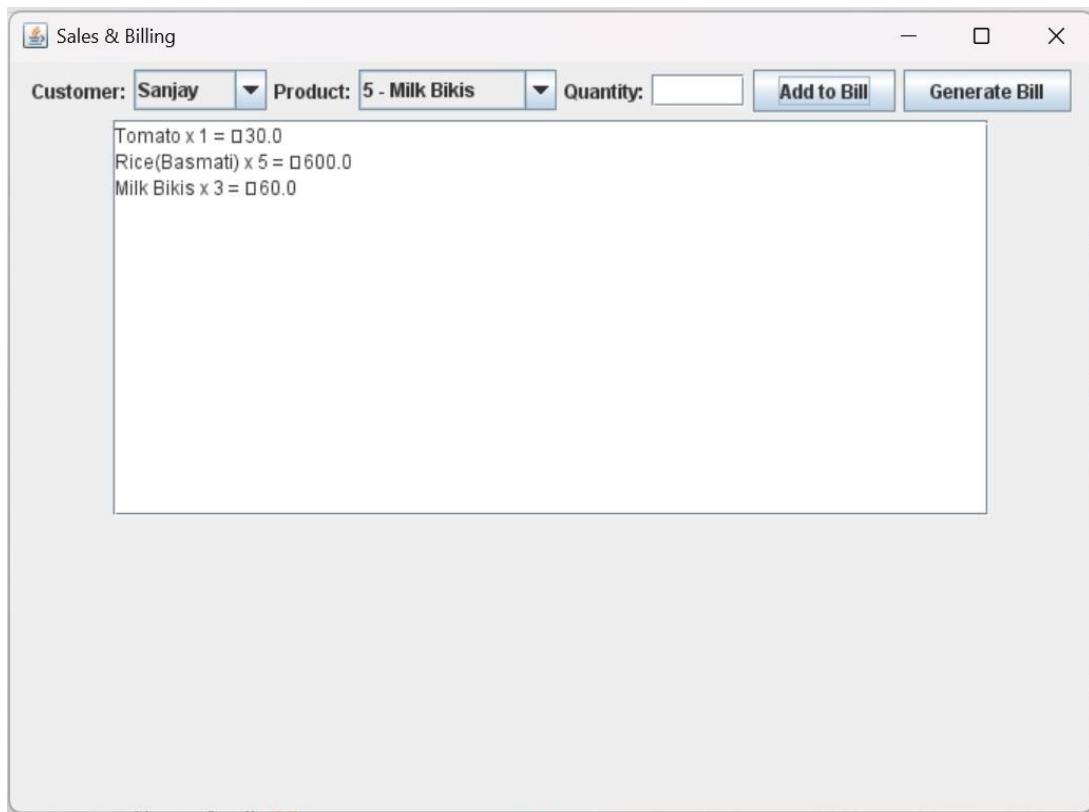


Customer Management

Name: Phone: Email:

Name: Raj | Phone: 9876543210 | Email: abc@email.com
 Name: Sanjay | Phone: 2547869563 | Email: hgds@gmail.com
 Name: Puvanesu | Phone: 4764580745 | Email: njil@gmail.com

Fig 6.3 Customer Management



The image shows a software window titled "Sales & Billing". At the top, there are three input fields: "Customer:" with a dropdown menu showing "Sanjay", "Product:" with a dropdown menu showing "5 - Milk Bikis", and "Quantity:" with an empty text box. To the right of these fields are two buttons: "Add to Bill" and "Generate Bill". Below the input fields is a large rectangular text area containing the following text:

Tomato x 1 = ₹ 30.0
Rice(Basmati) x 5 = ₹ 600.0
Milk Bikis x 3 = ₹ 60.0

Fig 6.4 Sales And Billing

CHAPTER 7

CONCLUSION

This project successfully demonstrates the development of a **database-driven application** using **Java Swing** for the frontend and **MySQL** for robust data persistence. It efficiently automates core grocery management tasks, ensuring **faster billing** processes and significantly **improved inventory accuracy** through real-time stock tracking. The application provides an intuitive desktop interface for **CRUD operations** on products, customers, and sales records, replacing error-prone manual systems. Future improvements are planned, including implementing a secure **user authentication** layer, enabling **PDF bill export** functionality for professional documentation, and exploring **online synchronization** for multi-terminal support and remote monitoring. **The successful integration of the Java GUI with the backend database validates skills in full-stack desktop application development.** The system establishes a reliable digital foundation, **making it scalable for small to medium-sized grocery businesses.**

REFERENCES

1. Oracle Java Documentation – <https://docs.oracle.com/javase/>
2. MySQL Documentation – <https://dev.mysql.com/doc/>
3. TutorialsPoint – JDBC and MySQL tutorials
4. Stack Overflow – Community coding references