

# Grocery Store Management System - Documentation

## Overview

The Grocery Store Management System is designed to automate the operations of a grocery store, providing features for inventory management, billing, and employee management. The system allows

an admin and staff to manage items, track sales, generate bills, and calculate salaries, making the store's operations more efficient and organized.

## Features

### 1. Item Management

- Add New Items: The admin can add new items to the inventory with their cost price and selling price.
- Display Items: The system shows the current stock of items, including their quantities and prices.
- Update Quantities: Item quantities can be increased or decreased based on sales or stock replenishment.

### 2. Billing System

- Add Items to Bill: Items can be added to a bill with their quantity and unit price.
- Total Bill Calculation: The system calculates the total cost of the items in the bill.
- Profit Calculation: It calculates the profit by comparing the selling price with the cost price of the items sold.

### 3. Employee Management

- Add Employee Records: The admin can add new employees, including information like employee ID, name, gender,

work shift, and salary.

- Leave Management: The system tracks employee leave days and adjusts their salary accordingly.
- Salary Calculation: The system calculates the final salary of employees after deducting leave days.

#### 4. Admin Operations

- Admin Login: Only authorized admins can log into the system with a secure username and password.
- Admin Access: The admin has full control over managing items, employees, and viewing profits and expenditures.

### **Technologies Used**

The system is implemented using C++ and follows Object-Oriented Programming (OOP) principles to ensure code modularity and reusability.

#### Key Concepts Implemented:

- Classes: Represent different entities like items, employees, bills, etc.
- Inheritance: Used for creating specialized classes (e.g., employee management).
- Constructors: Used to initialize object properties when objects are created.
- Member Functions: Functions associated with the class that define behaviors and actions (e.g., adding items, updating quantities, calculating totals).

### **System Requirements**

To run this system, ensure that you have the following:

- C++ Compiler: Any modern C++ compiler (e.g., GCC, MSVC).

- Operating System: Compatible with Windows, macOS, and Linux.

## **Usage**

### 1. Starting the System:

- Compile the program using a C++ compiler.
- Run the compiled executable to start the system.

### 2. Login:

- Admin will need to enter a secure username and password to log in.

### 3. Managing Items:

- Admin can add, view, and update item quantities.
- The system automatically updates the total cost and profit for each item sold.

### 4. Managing Employees:

- Admin can add employee details, track leave days, and adjust salaries.

### 5. Billing:

- The cashier can add items to the bill, update quantities, and generate the total bill, which includes the profit.

## **Classes Overview**

### 1. Item Class

Represents each item in the store.

#### Member Variables:

- string itemName: Name of the item.

- int itemID: Unique identifier for the item.
- float costPrice: Purchase cost of the item.
- float sellingPrice: Price at which the item is sold.
- int quantity: Current stock quantity.

#### Member Functions:

- void addItem(): Adds a new item to the system.
- void updateQuantity(): Updates the quantity of the item.
- void displayItem(): Displays item details.

## 2. Employee Class

Represents each employee in the store.

#### Member Variables:

- string name: Name of the employee.
- int employeeID: Unique identifier for the employee.
- float salary: Monthly salary.
- int leaves: Number of leave days taken.

#### Member Functions:

- void addEmployee(): Adds a new employee.
- void calculateSalary(): Calculates the employee's salary after deductions.

## 3. Bill Class

Represents the sales bill for a customer.

#### Member Variables:

- `vector<Item> items`: A list of items in the bill.
- `float totalAmount`: The total cost of items.
- `float profit`: The profit generated from the sale.