

# **INVENTORY MANAGEMENT SYSTEM**

**Project report submitted in partial fulfillment of the Requirements for the Award of the**

**Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

SRI PALLAVI      24KB1A05KA

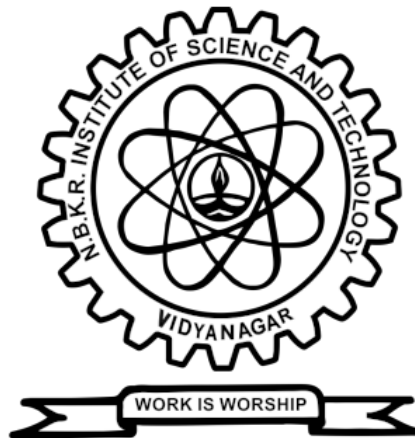
KAVYA REDDY      24KB1A05CU

JAHNAVI      24KB1A05DP

SNEHA LATHA      24KB1A05GA

**Under the Guidance of**

**SMT.SRUTHI**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# **INVENTORY MANAGEMENT SYSTEM**

**Project report submitted in partial fulfillment of the Requirements for the Award of the  
Degree of**

## **BACHELOR OF TECHNOLOGY**

**In**

## **COMPUTER SCIENCE AND ENGINEERING**

**By**

SRI PALLAVI      24KB1A05KA

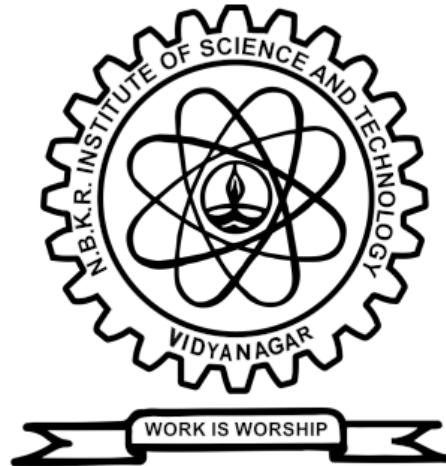
KAVYA REDDY      24KB1A05CU

JAHNAVI      24KB1A05DP

SNEHA LATHA      24KB1A05GA

**Under the Guidance of**

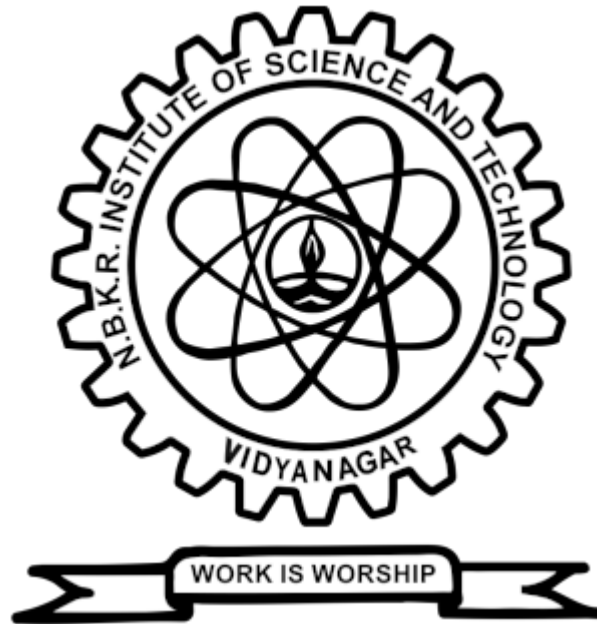
**SMT.SRUTHI**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**N.B.K.R.INSTUTUTE OF SCIENCE & TECHNOLOGY**

**N.B.K.R.INSTITUTE OF SCIENCE & TECHNOLOGY**  
**(AUTONOMOUS)**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project report entitled YOUR PROJECT TITLE being submitted by

SRI PALLAVI	24KB1A05KA
KAVYA REDDY	24KB1A05CU
JAHNAVI	24KB1A05DP
SNEHA LATHA	24KB1A05GA

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the N.B.K.R.INSTITUTE OF SCIENCE & TECHNOLOGY, VIDYANAGAR is a record of bonafied work carried out under my guidance and supervision.

Dr.A.Raja sekhar reddy

**SMT.STRUTHI**

**Designation**

**Dr. HOD Name**

**M.Tech, Ph.D**

**Head of the Department**

## **DECLARATION**

I hereby declare that the dissertation entitled **Inventory Management System** submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place : vidyanagar

Date: 05-05-2025

**SRI PALLAVI**

**24KB1A05KA**

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all those who contributed to the successful completion of the *Inventory Management System* project.

First and foremost, I would like to thank B.Sruthi, my guide and mentor, for their invaluable guidance, constructive feedback, and continuous support throughout the development of this project. Their expertise and encouragement greatly enhanced my learning experience.

I am also grateful to Head of the department Dr.A.raja shekhar reddy,for providing the necessary resources and a conducive environment to work on this project.

My heartfelt thanks go to my team members and peers for their collaboration, assistance, and insightful suggestions during various stages of development.

Lastly, I extend my deepest appreciation to my family and friends for their unwavering support and motivation, which played a crucial role in completing this project successfully.

This project has been a significant learning experience and has helped me deepen my understanding of inventory systems, database management, and software development processes.

***THANKYOU***

## **ABSTRACT OF THE PROJECT**

### **INVENTORY MANAGEMENT SYSTEM**

The *Inventory Management System* is a software solution designed to streamline the process of managing stock levels, orders, sales, and deliveries within an organization. This system aims to enhance accuracy, efficiency, and real-time monitoring of inventory, thereby minimizing human errors and optimizing resource usage.

This project focuses on automating key inventory operations, including item tracking, stock updates, supplier management, and report generation. By integrating a centralized database and a user-friendly interface, the system ensures that data is updated instantly and accessible to authorized users. The system is developed using [mention technology stack, e.g., PHP, MySQL, and JavaScript] and follows a modular structure to support scalability and easy maintenance.

The implementation of this system enables businesses to make informed decisions based on accurate inventory data, reduce operational costs, and improve overall productivity. It is particularly beneficial for retail stores, warehouses, and manufacturing units that require efficient inventory oversight.

# INTRODUCTION

## 1.1.INTRODUCTION

Inventory management plays a crucial role in ensuring the smooth operation of any business that deals with physical products. An effective inventory system helps track stock levels, sales, and restocking needs while reducing waste and optimizing costs. This project presents a simple yet functional Inventory Management System implemented in the C programming language. It allows users to add products, view inventory, process sales, and review sales records, making it a helpful tool for small businesses or educational purposes.

## 1.2.PROBLEM STATEMENT

You are tasked with developing a simple **Inventory and Sales Management System** in C for a small retail store. The system should manage a list of products, allow recording of sales, and provide the ability to view both inventory and sales history.

Create a simple **Inventory Management System** in C for a small shop. The system should:

- Allow the user to **add products** with ID, name, quantity, and price.
- Let the user **sell products**, update stock, and record each sale.
- Provide options to **list all products** and **view sales history**.
- Support up to 100 products and 100 sales.

## 1.3.SCOPE

- **Product Management:**

Add and store product details including ID, name, quantity, and price.

- **Sales Tracking:**

Record product sales, calculate total price, and update inventory stock levels.

- **Data Display:**

List all available products and view sales history.

- **User Interface:**

A simple menu-driven interface for easy operation by shopkeepers or staff.

#### 1.4.OBJECTIVE

The main objective of this program is to develop a simple, console-based **Inventory and Sales Management System** using the C programming language. It aims to:

- Help small shop owners **manage product details** like ID, name, quantity, and price.
- Enable users to **record and track sales** of products accurately.
- Maintain an **up-to-date stock level** after each sale.
- Provide an easy-to-use **menu interface** for listing products and viewing sales reports.

This program serves as a basic tool for improving manual inventory tracking and minimizing human errors in stock and sales management.



## Literature Survey / Existing System

Inventory and sales management is a critical function for businesses of all sizes. Traditionally, small shops and businesses have used **manual registers or spreadsheets** to record stock and sales data. These methods are time-consuming, prone to human error, and lack real-time updates, which can lead to inaccurate inventory levels and loss of sales data.

Existing Systems:

### 1. Manual Inventory Books:

- Still used in small retail stores.
- Difficult to update and maintain.
- Errors are common and tracking history is hard.

### 2. Spreadsheet-Based Systems (e.g., Excel):

- Require basic computer knowledge.
- Offer simple formulas but are not ideal for real-time or large-scale operations.
- No automated alerts for low stock or invalid sales.

### 3. Commercial Inventory Software (e.g., Tally, Zoho Inventory):

- Feature-rich, supports invoicing, GST, and reporting.
- Suitable for medium to large businesses.
- Expensive and may be complex for small shopkeepers.
- Given the limitations of existing manual or complex systems, this project proposes a **basic C program** that:

Need for a New System:

Given the limitations of existing manual or complex systems, this project proposes a **basic C program** that:

- Helps manage products and sales efficiently.
- Is easy to use and lightweight.
- Serves as an educational tool or prototype for further development.

### 3. Software Requirement Analysis

#### **Functional Requirements:**

These are the specific behaviors and functions the system must perform:

**1. Add Product:**

- The system should allow the user to input and store product ID, name, quantity, and price.

**2. List Products:**

- Display all stored products with their details (ID, name, quantity, price).

**3. Sell Product:**

- Allow the user to sell a product by entering its ID and quantity.
- Reduce the product's quantity based on the sale.
- Calculate and display the total sale price.
- Store the sale details (product ID, quantity sold, total price).

**4. View Sales:**

- Display a list of all recorded sales with product ID, quantity sold, and total price.

**5. Exit:**

- Allow the user to exit the program gracefully.

#### **Non-Functional Requirements:**

These define the system's quality attributes:

**1. Usability:**

- The program should have a simple, text-based interface suitable for non-technical users.

**2. Performance:**

- The system should handle up to 100 products and 100 sales efficiently in memory.

**3. Reliability:**

- Basic error handling should be present for invalid inputs (e.g., insufficient stock, product not found).

**4. Maintainability:**

- The code should be modular and structured, making it easy to modify or extend.

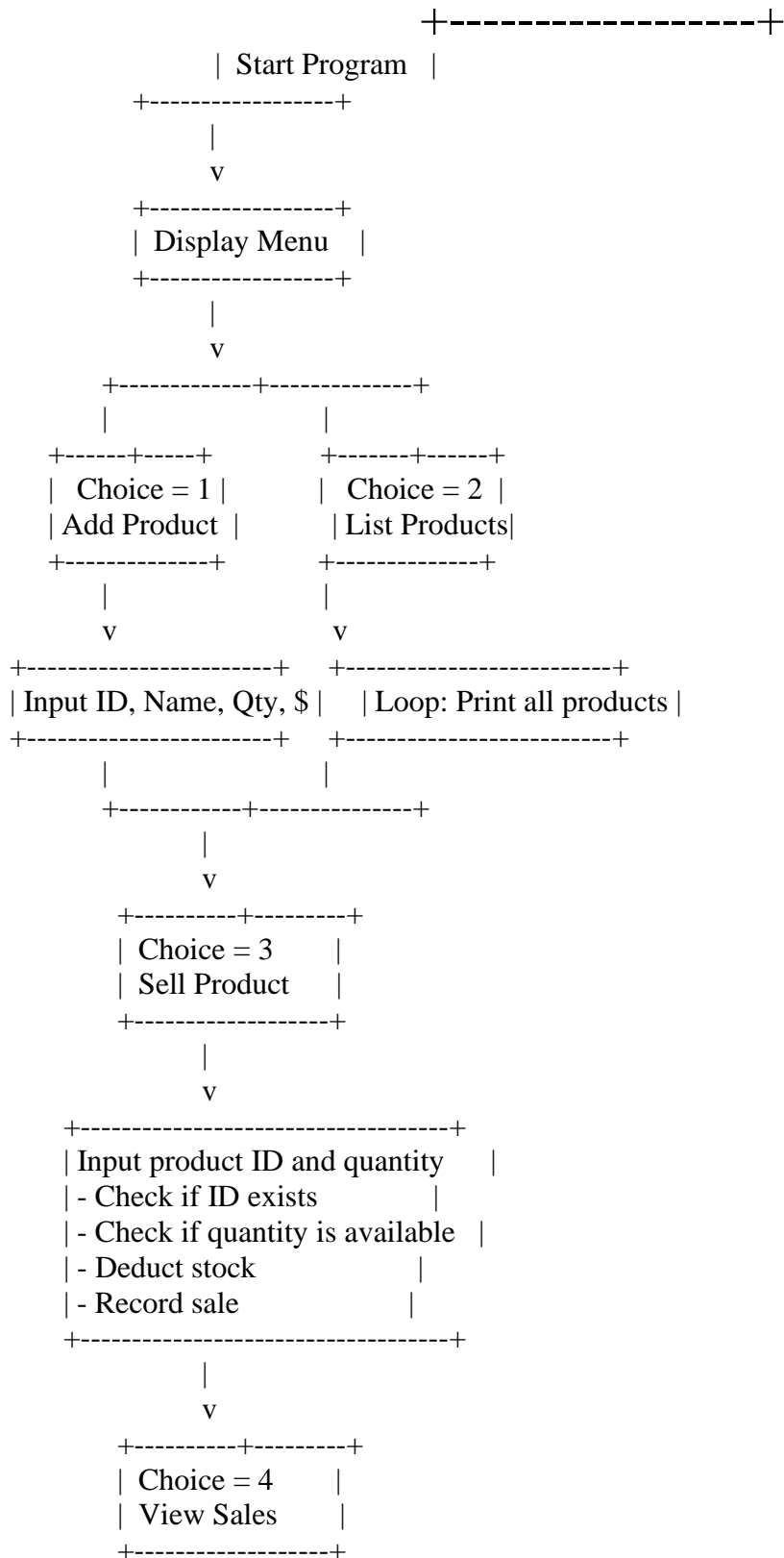
**5. Portability:**

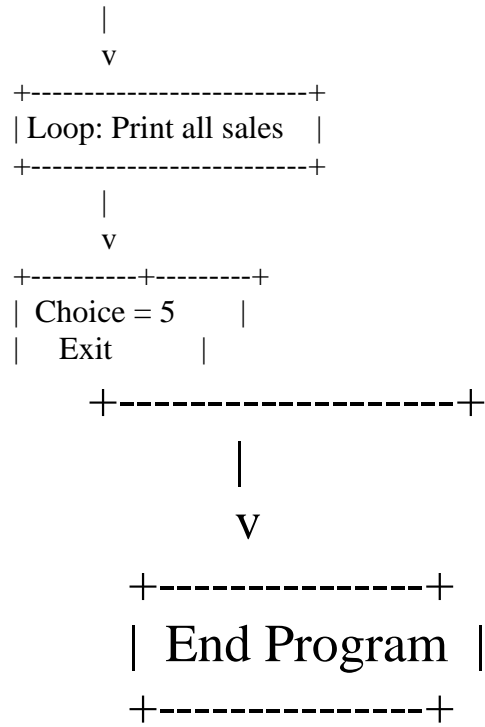
- The program should compile and run on any standard C compiler without platform dependency.

**6. Limitations:**

- Data is not saved permanently; all data is lost when the program exits.
- No user authentication or access control.

## 4. Software Design





## 5.PROPOSED SYSTEM

Module	Functionality
Product Management	Add product details to the inventory List all available products.
Sales Management	Sell products and update inventory Record sales transactions and list them.
Inventory Management	Check and update product quantities. Ensure enough stock is available for each sale.
User Interface	Display interactive menus and prompts. Provide feedback messages to the user.
Error Handling and Validation	Validate user inputs. Handle errors like invalid product IDs or quantities.

## 6.coding

```
#include <stdio.h>
#include <string.h>

#define MAX_PRODUCTS 100
#define MAX_SALES 100
#define MAX_NAME_LENGTH 50

// Product structure
typedef struct {
    int id;
    char name[MAX_NAME_LENGTH];
    int quantity;
    float price;
} Product;

// Sale structure
typedef struct {
    int product_id;
    int quantity;
    float total_price;
} Sale;

Product products[MAX_PRODUCTS];
Sale sales[MAX_SALES];
int product_count = 0;
int sale_count = 0;

// Add a new product
void addProduct() {
    if (product_count >= MAX_PRODUCTS) {
        printf("Product list is full.\n");
```

```

        return;
    }

    printf("Enter product ID: ");
    scanf("%d", &products[product_count].id);

    printf("Enter product name: ");
    scanf(" %[^\\n]s", products[product_count].name);

    printf("Enter quantity: ");
    scanf("%d", &products[product_count].quantity);

    printf("Enter price: ");
    scanf("%f", &products[product_count].price);

    product_count++;
    printf("Product added successfully.\\n");

    }

    // List all products
    void listProducts() {
        if (product_count == 0) {
            printf("No products available.\\n");
            return;
        }

        printf("ID\\tName\\tQuantity\\tPrice\\n");
        for (int i = 0; i < product_count; i++) {
            printf("%d\\t%s\\t%d\\t%.2f\\n", products[i].id, products[i].name, products[i].quantity, products[i].price);
        }

    }

    // Sell a product
    void sellProduct() {
        int id, qty;
        printf("Enter product ID to sell: ");
        scanf("%d", &id);

        printf("Enter quantity: ");
        scanf("%d", &qty);

        for (int i = 0; i < product_count; i++) {
            if (products[i].id == id) {
                if (products[i].quantity >= qty) {
                    products[i].quantity -= qty;

                    sales[sale_count].product_id = id;
                    sales[sale_count].quantity = qty;
                    sales[sale_count].total_price = products[i].price * qty;
                    sale_count++;

                } else {
                    printf("Not enough quantity.\\n");
                }
            }
        }
    }

```

```

        }
        return;
    }
}

printf("Product not found.\n");

}

// List all sold items
void listSales() {
    if (sale_count == 0) {
        printf("No sales recorded yet.\n");
        return;
    }

    printf("Product ID\tQuantity\tTotal Price\n");
    for (int i = 0; i < sale_count; i++) {
        printf("%d\t%d\t%.2f\n", sales[i].product_id, sales[i].quantity, sales[i].total_price);
    }

}

int main() {
    int choice;
    do {
        printf("\n--- Inventory System ---\n");
        printf("1. Add Product\n");
        printf("2. List Products\n");
        printf("3. Sell Product\n");
        printf("4. View Sales\n");
        printf("5. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addProduct(); break;
            case 2: listProducts(); break;
            case 3: sellProduct(); break;
            case 4: listSales(); break;
            case 5: printf("Goodbye!\n"); break;
            default: printf("Invalid option.\n");
        }
    } while (choice != 5);

    return 0;
}

```



## Output Screens / Result

```
---- Inventory System ----
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 1
Enter product ID: 101
Enter product name: APPLE
Enter quantity: 50
Enter price: 2.5
Product added successfully.
```

```
---- Inventory System ----
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 1
Enter product ID: 102
Enter product name: BANANA
Enter quantity: 100
Enter price: 1.0
Product added successfully.
```

```
---- Inventory System ----
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 2
ID      Name      Quantity  Price
101     APPLE      50        2.50
102     BANANA     100       1.00
```

```
--- Inventory System ---
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 3
Enter product ID to sell: 101
Enter quantity: 10
Product sold. Total price: 25.00
```

```
--- Inventory System ---
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 4
Product ID      Quantity      Total Price
101             10           25.00
```

```
--- Inventory System ---
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 2
ID      Name      Quantity      Price
101     APPLE      40           2.50
102     BANANA     100          1.00
```

```
--- Inventory System ---
1. Add Product
2. List Products
3. Sell Product
4. View Sales
5. Exit
Enter choice: 5
Goodbye!
```

## 8.Conclusion

The Inventory Management System presented in this program provides a simple yet functional solution for managing product inventories and tracking sales in a small retail or business environment. It enables users to:

- **Add new products** with unique IDs, names, quantities, and prices.
- **List all available products** in the inventory.
- **Sell products** and automatically update stock levels.
- **Record and display sales transactions** including quantities sold and total revenue generated.

The system is implemented using basic C programming concepts such as structures, arrays, and control statements, making it easy to understand and maintain. Although it currently handles only in-memory data (without file storage), it forms a solid foundation for more advanced features like data persistence, user authentication, or report generation.

Overall, this program demonstrates effective modular design and provides a useful tool for managing basic inventory and sales operations in a console-based environment.