A PROJECT REPORT ON

# "Grocery  MANAGEMENT SYSTEM"

SUBMITTED BY:

Mr Pratik Abasaheb Hirve (2124UCEM1015)

SUBJECT:

CPP PROGRAMMING

Under the guidance

of

Miss Ishavari Tirse Mam

Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

# INDEX

# INTRODUCTION

The Grocery Management System is a software application designed to assist grocery store owners or managers in efficiently organizing and managing their inventory. The primary goal of the system is to automate the process of adding, updating, and tracking products, along with their pricing and stock levels, ensuring smooth operations in a grocery store. By eliminating manual record-keeping, this system reduces the chances of human error, improves the accuracy of stock control, and enhances the overall productivity of store operations.

## *CODE*

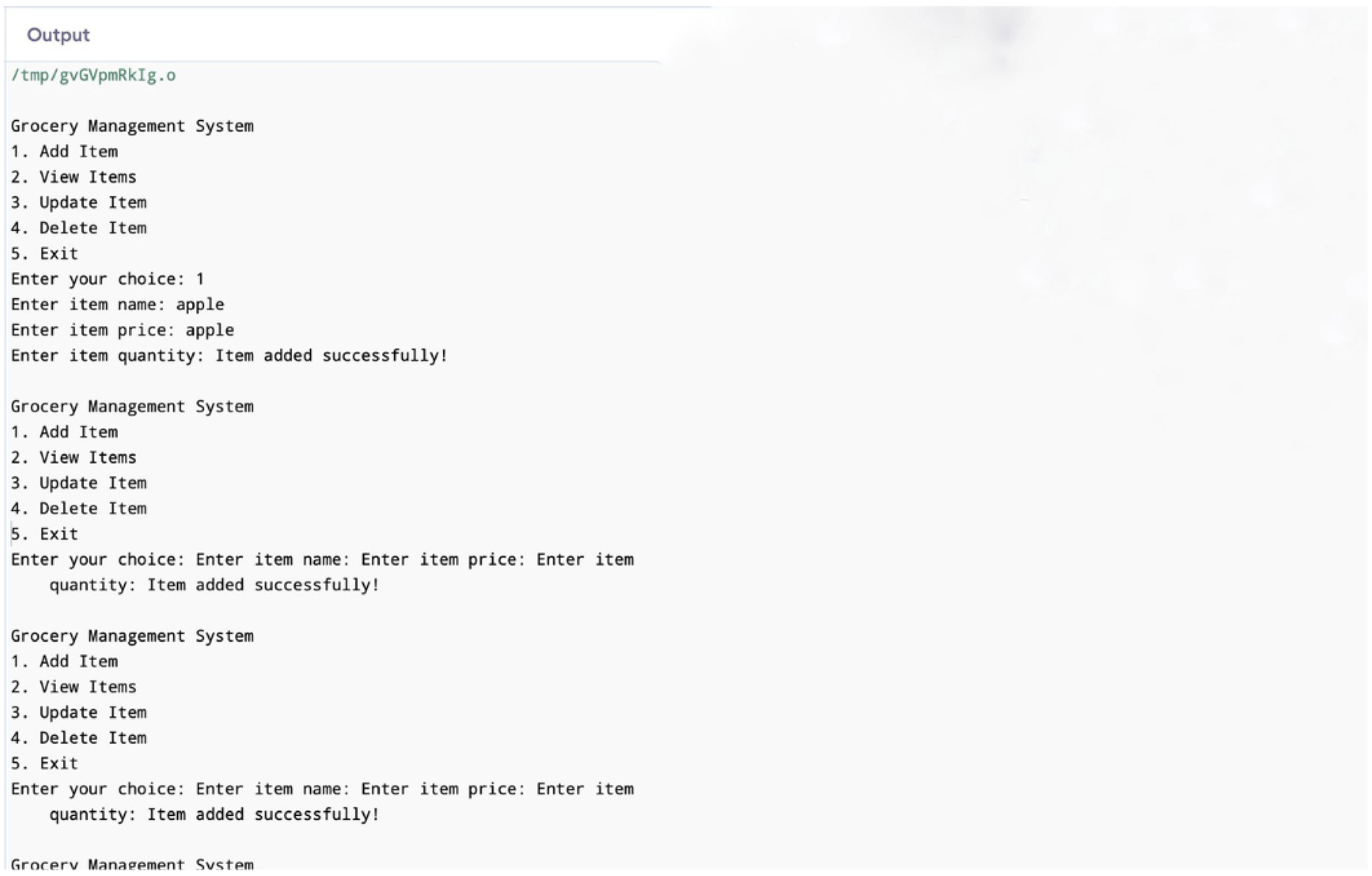
```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

// Structure to store grocery items
struct GroceryItem {
string name;
float price;
int quantity;#include <iostream>
#include <vector>
#include <string>

using namespace std;

// Structure to store product information
struct Product {
int id;
string name;
double price;
```

```
int quantity;
};

// Function to display all products
void displayProducts(const vector<Product>& products) {
cout << "\n--- Grocery Items ---\n";
cout << "ID\tName\t\tPrice\tQuantity\n";
for (const auto& product : products) {
cout << product.id << "\t" <<z product.name << "\t\t" << product.price << "\t" <<
product.quantity << endl;
}
}

// Function to add a new product
void addProduct(vector<Product>& products) {
Product newProduct;
cout << "Enter product ID: ";
cin >> newProduct.id;


cout << "Enter product name: ";
cin >> newProduct.name;
cout << "Enter product price: ";
cin >> newProduct.price;
cout << "Enter product quantity: ";
cin >> newProduct.quantity;
products.push_back(newProduct);
cout << "Product added successfully!\n";
}

// Function to update the quantity of a product
void updateQuantity(vector<Product>& products) {
int id, newQuantity;
cout << "Enter product ID to update: ";
cin >> id;
for (auto& product : products) {
if (product.id == id) {
cout << "Enter new quantity: ";
cin >> newQuantity;
```

```
"1. Display Products\n";
cout << "2. Add Product\n";
cout << "3. Update Product Quantity\n";
cout << "4. Exit\n";
cout << "Enter your choice: ";
cin >> choice;

switch (choice) {
case 1:
displayProducts(products);
break;
case 2:
addProduct(products);
break;
case 3:
updateQuantity(products);
break;
case 4:
cout << "Exiting system...\n";
break;
default:
```

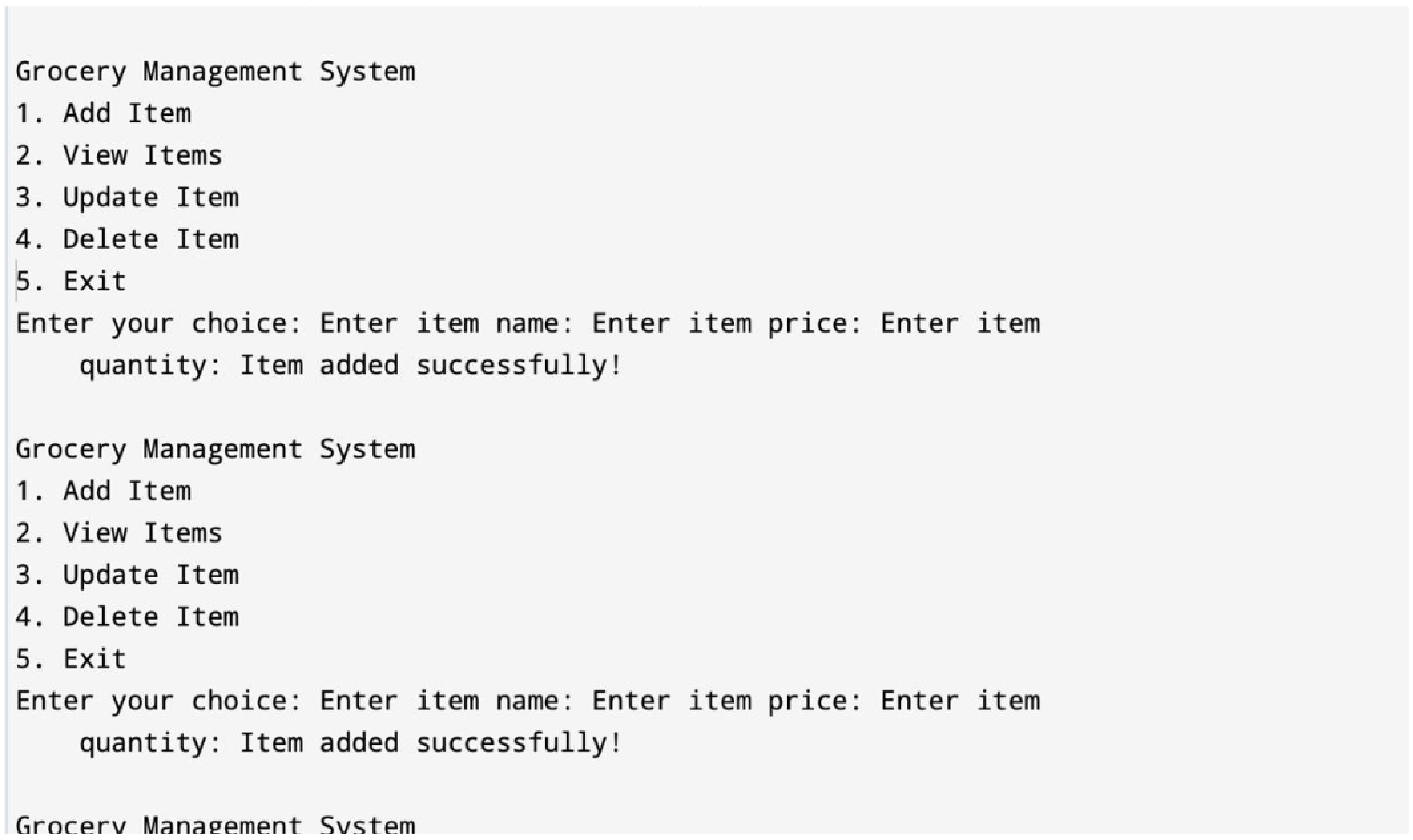
```
cout << "Invalid choice! Please try again.\n";
}
} while (choice != 4);

return 0;
}
};

// Function to display available items
void displayItems(const vector<GroceryItem>& items) {
if (items.empty()) {
cout << "No items in the inventory.\n";
return;
}

cout << "Available grocery items:\n";
cout << "-----------------------------\n";
cout << "S.No\tItem\tPrice\tQuantity\n";
int i = 1;
for (const auto& item : items) {
cout << i << "\t" << item.name << "\t" << item.price << "\t" << item.quantity << "\n";
i++;
}
cout << "-----------------------------\n";
}

// Function to add items to the inventory
void addItem(vector<GroceryItem>& items) {
GroceryItem item;
cout << "Enter the name of the item: ";
cin >> item.name;
cout << "Enter the price of the item: ";
cin >> item.price;
cout << "Enter the quantity of the item: ";
cin >> item.quantity;

items.push_back(item);
cout << "Item added successfully!\n";
cout << "Invalid item number!\n";
}
}

int main() {
vector<GroceryItem> groceryItems;
int choice;

do {
cout << "\nGrocery Management System\n";
cout << "1. Display items\n";
cout << "2. Add new item\n";
cout << "3. Delete an item\n";
cout << "4. Exit\n";
cout << "Enter your choice: ";
cin >> choice;

switch (choice) {
case 1:
displayItems(groceryItems);
break;
case 2:
addItem(groceryItems);
break;
case 3:
deleteItem(groceryItems);
break;
case 4:
cout << "Exiting the system.\n";
break;
default:
cout << "Invalid choice! Please try again.\n";
}
} while (choice != 4);

return 0;
}
```

# OUTPUT



Output

```
/tmp/gvGVpmRkIg.o

Grocery Management System
1. Add Item
2. View Items
3. Update Item
4. Delete Item
5. Exit
Enter your choice: 1
Enter item name: apple
Enter item price: apple
Enter item quantity: Item added successfully!

Grocery Management System
1. Add Item
2. View Items
3. Update Item
4. Delete Item
5. Exit
Enter your choice: Enter item name: Enter item price: Enter item
    quantity: Item added successfully!

Grocery Management System
1. Add Item
2. View Items
3. Update Item
4. Delete Item
5. Exit
Enter your choice: Enter item name: Enter item price: Enter item
    quantity: Item added successfully!

Grocery Management System
```



```
Grocery Management System
1. Add Item
2. View Items
3. Update Item
4. Delete Item
5. Exit
Enter your choice: Enter item name: Enter item price: Enter item
    quantity: Item added successfully!

Grocery Management System
1. Add Item
2. View Items
3. Update Item
4. Delete Item
5. Exit
Enter your choice: Enter item name: Enter item price: Enter item
    quantity: Item added successfully!

Grocery Management System
```

# CONCLUSION

Overall, the Grocery Management System demonstrates the potential for technology to optimize retail processes, enhancing both productivity and customer service within the grocery sector. Further enhancements could include features like user authentication, detailed sales reports, and support for discounts and promotions, adding even more value to the system.