Members:

Muhammad Sudais Khalid

Hashir Khan

Naaz Ahmad

Program: BS Artificial Intelligence

Semester #01 (2023)

Subject: Programming Fundamental

Project: Retail Shop management system

Submitted to: Ma’am Samina Mushtaq & Sir Ghulam Mustafa.

Contents

Retail Shop Management System Report

1. Introduction:

The provided C++ program is a simple Retail Shop Management System designed to manage product records within a retail store. The system offers various functionalities such as adding new products, displaying product details, calculating the total value of the stock, searching for a product, updating product information, and deleting products from the inventory.

2. Program Structure:

The program is structured as a console-based application.
It utilizes a struct named `Product` to store information about each product, including name, price, and quantity. The main functionality is encapsulated within functions, making the code modular and easy to understand.

3. Functions:

Adding a Product:

Users can add new products to the inventory by providing the product name, price, and quantity. A limit of 100 products is set to prevent overflow.

Displaying Product Details:

Displays a tabular format of product details, including name, price, and quantity. If the inventory is empty, a message is displayed.

Calculating Total Value:

Computes and displays the total value of the stock, considering the price and quantity of each product.

Searching for a Product:

Allows users to search for a specific product by name. Displays product details if found; otherwise, notifies the user that the product is not in the inventory.

Updating Product Information:

Enables users to update the price and quantity of an existing product by providing the product name. Notifies the user if the product is not found.

Deleting a Product:

Permits users to remove a product from the inventory by specifying the product name.
Adjusts the product array to maintain a contiguous list of products. Notifies the user if the product is not found.

4. User Interface:

The program presents a simple menu-driven interface for users to choose from different operations. Utilizes console text attributes to enhance the visual presentation, such as different colors for various outputs.

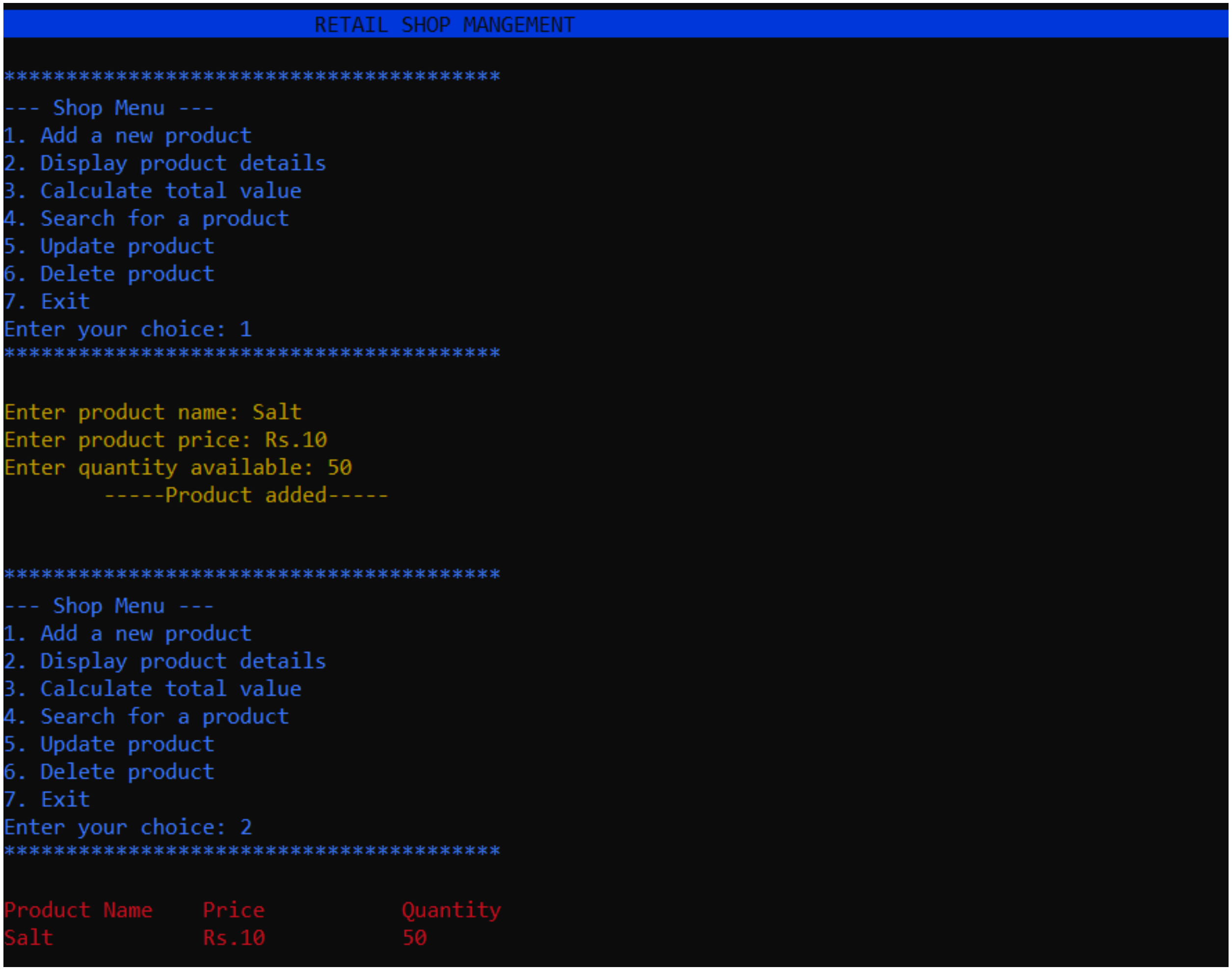
5. Conclusion:

The Retail Shop Management System provides basic inventory management functionalities for a retail store. While it serves its purpose, there is room for improvement in terms of error handling and user interface enhancements. This program can be a good starting point for further development and expansion of features in a retail management system.

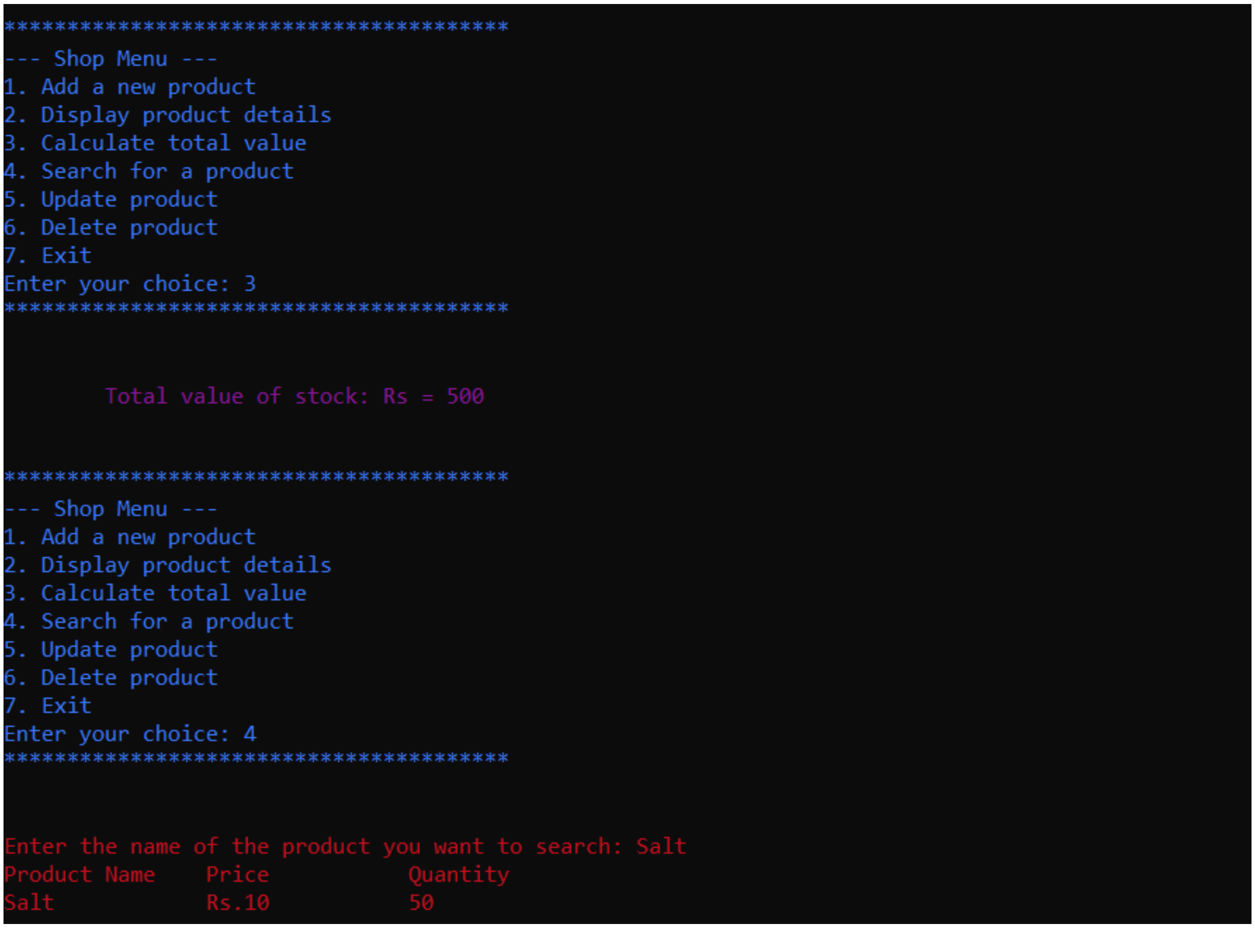
Code:

```
#include <iostream>
#include <windows.h>
using namespace std;

struct Product {
    string name;
    int price;
    int quantity;
};

void adding(Product record[], int& count)
{

    if (count < 100)
             {
                        HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,6);
        cout << "Enter product name: ";
        cin >> record[count].name;

        cout << "Enter product price: Rs.";
        cin >> record[count].price;

        cout << "Enter quantity available: ";
        cin >> record[count].quantity;

        count++;

        cout << "\t-----Product added-----\n";
    }
             else
              {
        cout << "Sorry, the range is full.\n";
    }
}

void displaying(const Product record[], int count)
 {
    if (count == 0)
            {
        cout << "No products available.\n";
    }
    HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,4);
    cout << "Product Name" << "\t" << "Price" << "\t\t" << "Quantity" << endl;
    for (int i = 0; i < count; ++i)
              {
        cout << record[i].name << "\t\t" << "Rs." << record[i].price << "\t\t" << record[i].quantity
<< endl;
    }


}

double TotalValue(const Product record[], int count)

  {
            HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,5);
    double total = 0.0;
    for (int i = 0; i < count; ++i)

            {

        total += record[i].price * record[i].quantity;
    }
    return total;
}

void search(const Product record[], int count)
  {
            HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,4);
    string searchName;
    cout << "\nEnter the name of the product you want to search: ";
    cin >> searchName;
    bool found = false;
    for (int i = 0; i < count; ++i)
            {
        if (record[i].name == searchName)
                        {
            found = true;
            cout << "Product Name" << "\t" << "Price" << "\t\t" << "Quantity" << endl;
            cout << record[i].name << "\t\t" << "Rs." << record[i].price << "\t\t" <<
record[i].quantity << endl;
            break;
        }
    }
    if (!found)
            {

        cout << "Product not found." << endl;
    }
}

void update(Product record[], int count)
{
            HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,7);
    string updateName;
    cout << "\nEnter the name of the product you want to update: ";
    cin >> updateName;
    bool found = false;
    for (int i = 0; i < count; ++i)
            {
        if (record[i].name == updateName)
                         {
            found = true;
            cout << "Enter new price for " << record[i].name << ": Rs.";
            cin >> record[i].price;
            cout << "Enter new quantity for " << record[i].name << ": ";
            cin >> record[i].quantity;
            cout << "Product details updated successfully.\n";
            break;
        }
    }
    if (!found)
            {
        cout << "Product not found." << endl;
    }
}

void remove(Product record[], int& count)
{
            HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,10);
    string deleteName;
    cout << "\nEnter the name of the product you want to delete: ";
    cin >> deleteName;
    bool found = false;
    for (int i = 0; i < count; ++i)
             {
        if (record[i].name == deleteName)
                         {
            found = true;
            for (int j = i; j < count - 1; ++j)
                                     {
                record[j] = record[j + 1];
            }
            count--;
            cout << "\n\tProduct deleted successfully.";
            break;
        }

    }
    if (!found) {
        cout << "Product not found." << endl;
    }
}

int main()
{
    Product record[100];
    int count = 0;
    int choice;
    HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,16);
    cout<<"\n\t\t\t RETAIL SHOP MANGEMENT ";
    while (true ) {
            HANDLE h=GetStdHandle(STD_OUTPUT_HANDLE);
            SetConsoleTextAttribute(h,9);

        cout << "\n\n****************************************\n";
        cout << "--- Shop Menu ---\n";
        cout << "1. Add a new product\n";
        cout << "2. Display product details\n";
        cout << "3. Calculate total value\n";
        cout << "4. Search for a product\n";
        cout << "5. Update product\n";
        cout << "6. Delete product\n";
        cout << "7. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        cout<<"****************************************"<<endl<<endl;

        switch (choice)
                {
            case 1:
                adding(record, count);
                break;
            case 2:
                displaying(record, count);
                break;
            case 3:
                cout << "\n\tTotal value of stock: Rs = " << TotalValue(record, count) << endl;
                break;
            case 4:
                search(record, count);
                break;
            case 5:
                update(record, count);
                break;
            case 6:
                remove(record, count);
                break;
            case 7:
                    cout<<"\n\t\t [ THANK YOU ] "<<endl;
                    exit(0);

                break;
            default:
                cout << "Invalid choice. Please try again.\n";
                break;
        }

    }

    return 0;
}
```

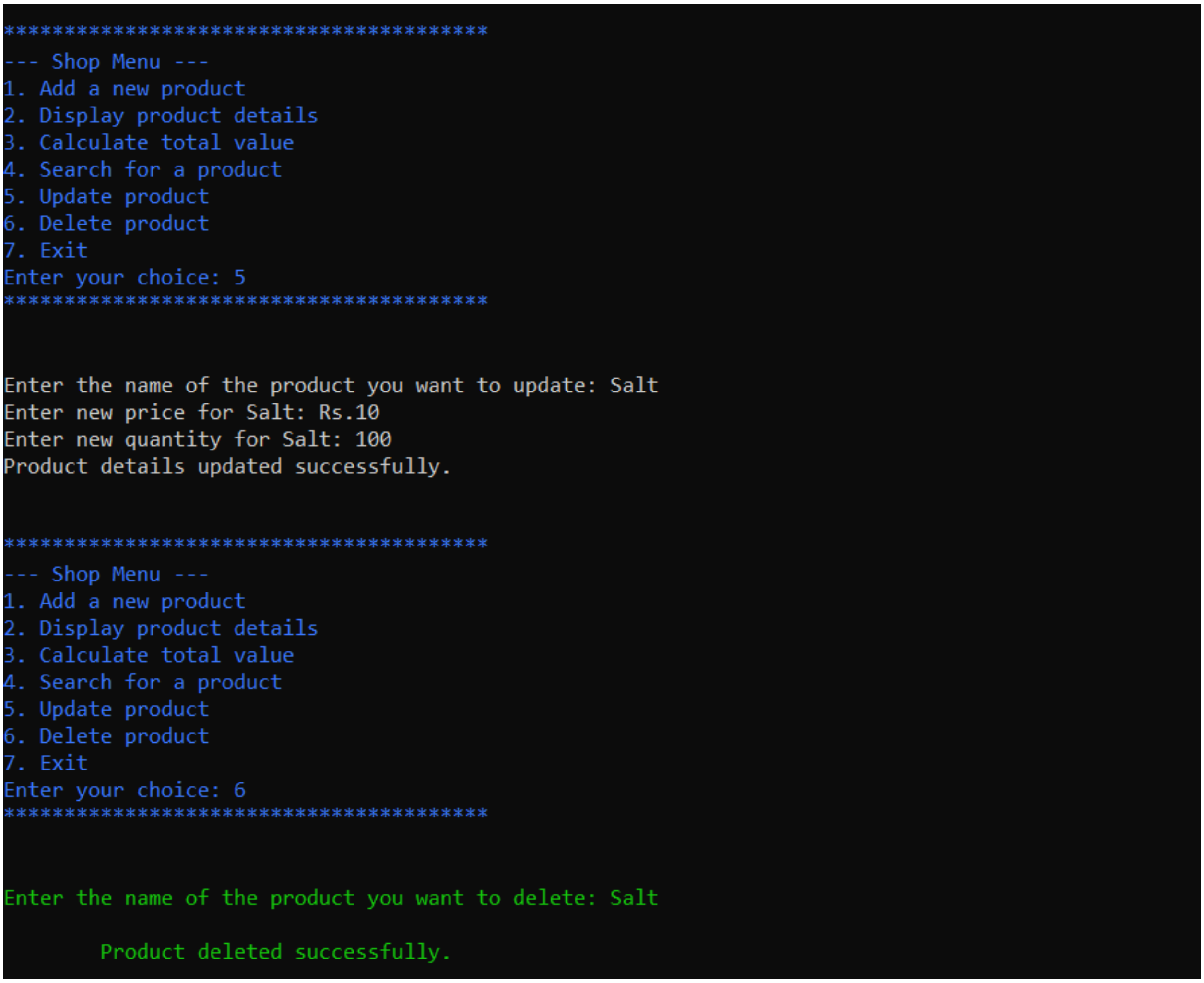
Output:



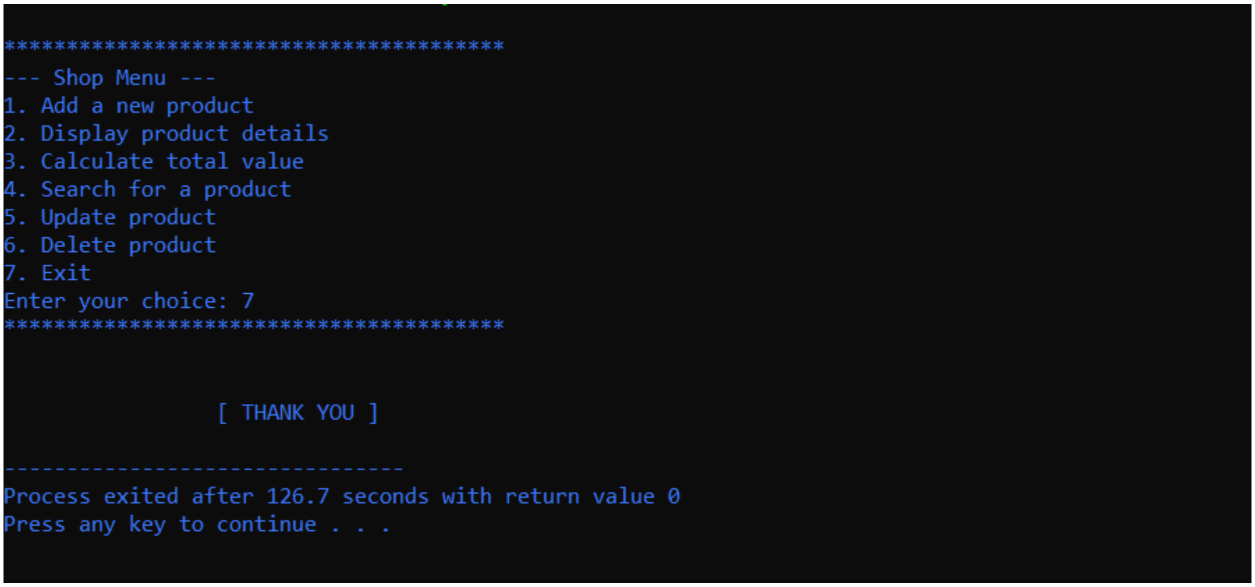
```
                         RETAIL SHOP MANGEMENT

****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 1
****************************************

Enter product name: Salt
Enter product price: Rs.10
Enter quantity available: 50
        -----Product added-----


****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 2
****************************************

Product Name    Price           Quantity
Salt            Rs.10           50
```



```
****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 3
****************************************


        Total value of stock: Rs = 500


****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 4
****************************************


Enter the name of the product you want to search: Salt
Product Name    Price           Quantity
Salt            Rs.10           50
```



```
****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 5
****************************************


Enter the name of the product you want to update: Salt
Enter new price for Salt: Rs.10
Enter new quantity for Salt: 100
Product details updated successfully.


****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 6
****************************************


Enter the name of the product you want to delete: Salt

        Product deleted successfully.
```



```
****************************************
--- Shop Menu ---
1. Add a new product
2. Display product details
3. Calculate total value
4. Search for a product
5. Update product
6. Delete product
7. Exit
Enter your choice: 7
****************************************


                 [ THANK YOU ]

--------------------------------
Process exited after 126.7 seconds with return value 0
Press any key to continue . . .
```