



"Programming in C Project"

EVALUATION REPORT

ON

"Inventory Management System"

Submitted By:

Vanshika Giri

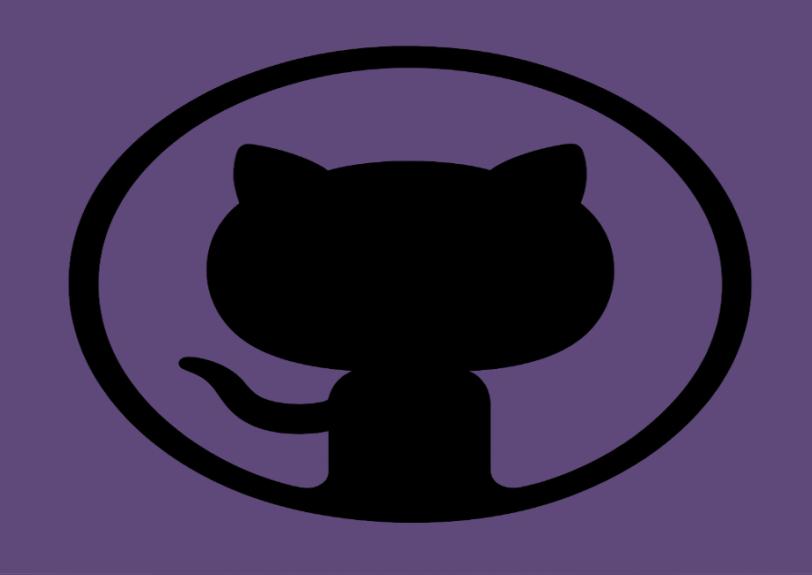
SAP ID: 590021900

Batch: B43

Submitted To:

Mr. Milton Kumar

GitHub



Repository Link

[“vanshika2621/InventoryManagementSystem-C-Project”](#)

INDEX

Contents:

Topic 1: Abstract

Topic 2: Problem Definition

Topic 3: System Design & Flow Chart

Topic 4: Algorithm (Simplified)

Topic 5: Implementation Details (C Code)

Topic 6: Testing & Results

Topic 7: Conclusion & Future Scope

Topic 8: References

Abstract

This project implements a simple Inventory Management System using the C programming language. It allows users to add products, view inventory, update stock, delete products, search for products and save the inventory data to a file. The project demonstrates the use of arrays, functions, loops, file handling, and string manipulation.

Problem Definition

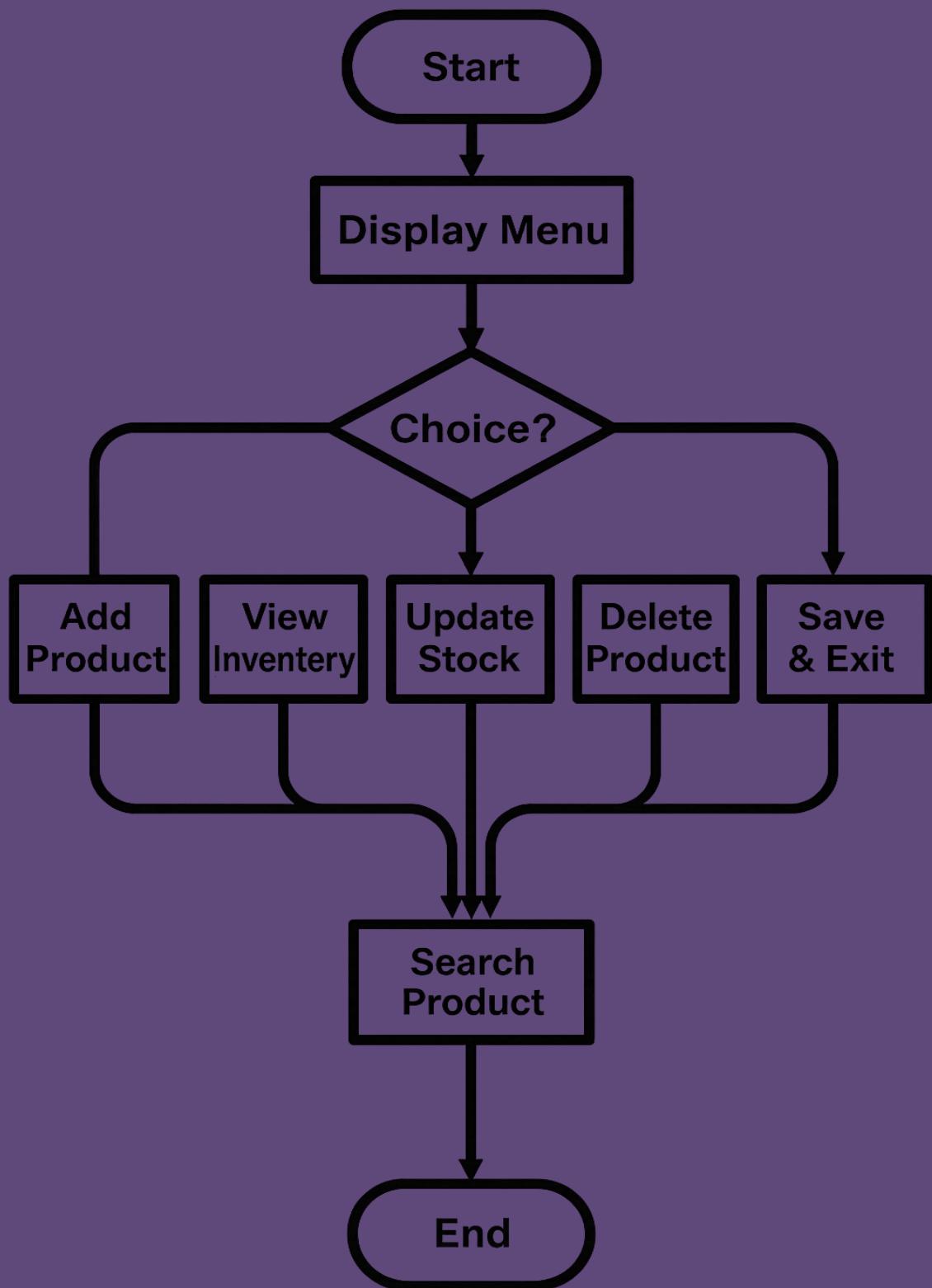
The goal of this project is to design and implement a console-based Inventory Management System. The system must allow users to manage product records efficiently with options to add, update, delete, search, and store inventory records permanently.

System Design

The system is divided into the following functional modules:

- Add Product
- View Inventory
- Update Stock
- Delete Product
- Search Product
- Save & Exit

Flow Chart



Algorithm (Simplified)

- 1. Display menu options.
- 2. Take user choice.
- 3. Perform selected operation.
- 4. Repeat until Save & Exit is chosen.

Implementation Details (C Code)

```
#include <stdio.h>
#include <string.h>
#define MAX 100

// Global Arrays

char productNames[MAX][50];
int productQuantities[MAX];
float productPrices[MAX];

int count = 0; // Total number of products

// Function Declarations

void addProduct();
void viewInventory();
void updateStock();
void deleteProduct();
void searchProduct();
void saveAndExit();

int main()
{
    int choice;

    while (1)
    {
        printf("\n-----");
        printf("\n INVENTORY MANAGEMENT SYSTEM");
        printf("\n-----");
        printf("\n1. Add Product");
        printf("\n2. View Inventory");
        printf("\n3. Update Stock");
```

```
printf("\n4. Delete Product");

printf("\n5. Search Product");

printf("\n6. Save & Exit");

printf("\nEnter your choice: ");

scanf("%d", &choice);

if (choice == 1)

    addProduct();

else if (choice == 2)

    viewInventory();

else if (choice == 3)

    updateStock();

else if (choice == 4)

    deleteProduct();

else if (choice == 5)

    searchProduct();

else if (choice == 6)

{

    saveAndExit();

    break;

}

else

    printf("\nInvalid choice! Try again.");

}

return 0;

}

// Add Product

void addProduct()
```

```
{  
    char name[50];  
    int qty;  
    float price;  
    int found = 0;  
    printf("\nEnter Product Name: ");  
    scanf("%s", name);  
    printf("Enter Quantity: ");  
    scanf("%d", &qty);  
    printf("Enter Price: ");  
    scanf("%f", &price);  
    for (int i = 0; i < count; i++)  
    {  
        if (strcmp(productNames[i], name) == 0)  
        {  
            productQuantities[i] += qty;  
            productPrices[i] = price;  
            printf("\nProduct '%s' updated successfully!\n", name);  
            found = 1;  
            break;  
        }  
    }  
    if (!found)  
    {  
        strcpy(productNames[count], name);  
        productQuantities[count] = qty;  
        productPrices[count] = price;  
        count++;  
    }  
}
```

```
printf("\nProduct '%s' added successfully!\n", name);

}

}

// View Inventory

void viewInventory()

{

if (count == 0)

{

printf("\nInventory is empty.\n");

return;

}

printf("\n-----");

printf("\n%-15s %-10s %-10s", "Product", "Quantity", "Price");

printf("\n-----");

for (int i = 0; i < count; i++)

{

printf("\n%-15s %-10d %-10.2f",

productNames[i],  
productQuantities[i],  
productPrices[i]);  
}  
printf("\n-----\n");  
}

// Update Stock  
void updateStock()  
{  
char name[50];  
int newQty, found = 0;
```

```
printf("\nEnter Product Name to Update: ");
scanf("%s", name);
for (int i = 0; i < count; i++)
{
    if (strcmp(productNames[i], name) == 0)
    {
        printf("Enter New Quantity: ");
        scanf("%d", &newQty);
        productQuantities[i] = newQty;
        printf("\nStock updated successfully!\n");
        found = 1;
        break;
    }
}
if (!found)
    printf("\nProduct not found!\n");
}

// Delete Product

void deleteProduct()
{
    char name[50];
    int found = 0;
    printf("\nEnter Product Name to Delete: ");
    scanf("%s", name);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(productNames[i], name) == 0)
```

```
{  
    for (int j = i; j < count - 1; j++)  
  
    {  
        strcpy(productNames[j], productNames[j + 1]);  
        productQuantities[j] = productQuantities[j + 1];  
        productPrices[j] = productPrices[j + 1];  
    }  
    count--;  
    printf("\nProduct deleted successfully!\n");  
    found = 1;  
    break;  
}  
}  
if (!found)  
    printf("\nProduct not found!\n");  
}  
// Search Product  
void searchProduct()  
{  
    char name[50];  
    int found = 0;  
    printf("\nEnter Product Name to Search: ");  
    scanf("%s", name);  
    for (int i = 0; i < count; i++)  
    {  
        if (strcmp(productNames[i], name) == 0)  
        {  
            printf("\nProduct Found!");  
        }  
    }  
}
```

```
    printf("\nName: %s", productNames[i]);

    printf("\nQuantity: %d", productQuantities[i]);

    printf("\nPrice: %.2f\n", productPrices[i]);

    found = 1;

    break;

}

}

if (!found)

    printf("\nProduct not found!\n");

}

// Save & Exit

void saveAndExit()

{

FILE *fp;

fp = fopen("inventory.csv", "w");

if (fp == NULL)

{

    printf("\nError saving file!\n");

    return;

}

fprintf(fp, "Product,Quantity,Price\n");



for (int i = 0; i < count; i++)

{

    fprintf(fp, "%s,%d,%.2f\n",

            productNames[i],



            productQuantities[i],



            productPrices[i]);
```

```
}

fclose(fp);

printf("\nInventory saved to inventory.csv");

printf("\nExiting program...\n");

}
```

Testing & Results

The system was tested using multiple product entries. All operations such as add, update, delete, and search worked successfully. The inventory file was correctly saved in CSV format.

Conclusion & Future Scope

The Inventory Management System works efficiently for small businesses. Future improvements can include database connectivity, login authentication, graphical user interface, and barcode scanning.

References

1. Balagurusamy, E. (2017). *Programming in ANSI C* (8th ed.). McGraw Hill Education, India.
2. Forouzan, B. A., & Gilberg, R. F. (2018). *Computer Science: A Structured Programming Approach Using C*. Cengage Learning.
3. Studytonight. (2024). *Stock Management System Project using C Language*.
Available at: <https://www.studytonight.com>
4. Tutorialspoint. (2024). *C Program to Store Inventory Using Structures*.
Available at: <https://www.tutorialspoint.com>
5. Sanfoundry. (2023). *C Program for Inventory Management System*.
Available at: <https://www.sanfoundry.com>
6. GitHub Open Source Repository. (2024). *Inventory Management System in C*.
Available at: <https://www.github.com>