

A Micro Project Report

on

Problem Solving using C Language

Submitted by

MAGAM VENKATA ANILKUMAR(23471A05HU)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-1

**NIRF rank in the band of 201-300 and is an ISO 9001:2015 certified Approved by
AICTE, New Delhi, Permanently affiliated to JNTU Kakinada, Approved by AICTE,
Accredited by NBA and accredited 'A+' grade by NAAC Narasaraopet-522601,
Palnadu(Dt.), Andhra Pradesh, India**

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that **Magam Venkata Anilkumar**, Roll No: **23471A05HU**, a Second Year Student of the Department of Computer Science and Engineering, has completed the Micro Project Satisfactorily in "Problem Solving using C Language" for the Academic Year 2024-2025..

Project Co-Ordinator
Mr. Shaik Rafi, M.Tech., (Ph.D).

Asst. Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.

Professor

INDEX

SI.NO	DESCRIPTION
1.	Develop project for super market billing system.

SUPER MARKET BILLING SYSTEM

AIM:

Develop project for super market billing system.

Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Product
{
    int id;
    char name[50];
    float price;
    int quantity;
};
void addProduct()
{
    FILE *file = fopen("product.txt", "a");
    if (!file) {
        printf("Error opening file for writing.\n");
        return;
    }
    struct Product product;
    printf("Enter product ID: ");
    scanf("%d", &product.id);
    printf("Enter product name: ");
    scanf("%s", product.name);
    printf("Enter product price: ");
    scanf("%f", &product.price);
    printf("Enter product quantity: ");
    scanf("%d", &product.quantity);
    fprintf(file, "%d %s %.2f %d\n", product.id, product.name, product.price,
product.quantity);
    printf("Product data added successfully!\n");
    fclose(file);
}
void displayProducts()
```

```

{
    FILE *file = fopen("product.txt", "r");
    if (!file)
    {
        printf("Error opening file for reading.\n");
        return;
    }
    struct Product product;
    printf("%-10s %-20s %-10s %-10s\n", "ID", "PRODUCT NAME", "PRICE",
"QUANTITY");
    printf("-----\n");
    while (fscanf(file, "%d %s %f %d", &product.id, product.name,
&product.price, &product.quantity) != EOF)
    {
        printf("%-10d %-20s %-10.2f %-10d\n", product.id, product.name,
product.price, product.quantity);
    }
    fclose(file);
}

void loadProducts(struct Product products[], int *count)
{
    FILE *file = fopen("product.txt", "r");
    if (!file)
    {
        printf("Error opening file for reading.\n");
        return;
    }
    *count = 0;
    while (fscanf(file, "%d %s %f %d", &products[*count].id,
products[*count].name, &products[*count].price,
&products[*count].quantity) != EOF)
    {
        (*count)++;
    }
    fclose(file);
}

void saveProducts(struct Product products[], int count)
{
    FILE *file = fopen("product.txt", "w");

```

```

    if (!file)
    {
        printf("Error opening file for writing.\n");
        return;
    }
    for (int i = 0; i < count; i++)
    {
        fprintf(file, "%d %s %.2f %d\n", products[i].id, products[i].name,
products[i].price, products[i].quantity);
    }
    fclose(file);
}

void generateBill()
{
    struct Product products[100];
    int count = 0;
    loadProducts(products, &count);
    int id, quantity;
    double totalBill = 0.0;
    printf("Enter Product ID and Quantity (0 to finish):\n");
    while (1)
    {
        printf("Product ID: ");
        scanf("%d", &id);
        if (id == 0) break;
        printf("Quantity: ");
        scanf("%d", &quantity);
        int found = 0;
        for (int i = 0; i < count; i++)
        {
            if (products[i].id == id)
            {
                found = 1;
                if (products[i].quantity >= quantity)
                {
                    double itemTotal = products[i].price * quantity;
                    totalBill += itemTotal;
                    products[i].quantity -= quantity;
                    printf("Added %s - Quantity: %d, Item Total: $%.2f\n",

```

```

products[i].name, quantity, itemTotal);
    }
    else
    {
        printf("Insufficient stock for %s.\n", products[i].name);
    }
    break;
}
}
if (!found)
{
    printf("Product ID %d not found.\n", id);
}
}
printf("-----\n");
printf("Total Bill: $%.2f\n", totalBill);
saveProducts(products, count);
}
int main()
{
    int choice;
    do
    {
        printf("\n1. Add Product\n2. Display Products\n3. Generate Bill\n4.
Exit\nChoose an option: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                addProduct();
                break;
            case 2:
                displayProducts();
                break;
            case 3:
                generateBill();
                break;
            case 4:
                printf("Exiting program.\n");

```

```
        break;
    default:
        printf("Invalid choice. Please try again.\n");
        break;
    }
} while (choice != 4);
return 0;
}
```


OUTPUT:

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 1

Enter product ID: 101

Enter product name: Apple

Enter product price: 1.50

Enter product quantity: 100

Product data added successfully!

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 1

Enter product ID: 102

Enter product name: Orange

Enter product price: 0.75

Enter product quantity: 50

Product data added successfully!

DISPLAYING PRODUCTS

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 2

ID	PRODUCT NAME	PRICE	QUANTITY
----	--------------	-------	----------

101	Apple	1.50	100
102	Orange	0.75	50

GENERATING A BILL

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 3

Enter Product ID and Quantity (0 to finish):

Product ID: 101

Quantity: 10

Added Apple - Quantity: 10, Item Total: \$15.00

Product ID: 102

Quantity: 20

Added Orange - Quantity: 20, Item Total: \$15.00

Product ID: 0

Total Bill: \$30.00

DISPLAYING PRODUCTS AFTER BILLING

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 2

ID	PRODUCT NAME	PRICE	QUANTITY
----	--------------	-------	----------

101	Apple	1.50	90
-----	-------	------	----

102	Orange	0.75	30
-----	--------	------	----

EXITING THE PROGRAM

1. Add Product
2. Display Products
3. Generate Bill
4. Exit

Choose an option: 4

Exiting program.