

**PSP [20ES104] COURSE PROJECT REPORT**

**On**

**“Grocery Store Management”**

Developed By:

H.T.NO STUDENT NAME

2203A51474 E.Manikanta

2203A51510 P.Dharma Reddy

2203A51514 P.Snehith Kumar

Under the Guidance of

Mrs. V. Shobha Rani, M.Tech.(Ph.D)

Assistant Professor

Submitted to

Department Computer Science and Artificial Intelligence SR University

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371 [www.sru.edu.in](http://www.sru.edu.in/)

**June 2023**

**Department of Computer Science and Artificial Intelligence**

**CERTIFICATE**

This is to certify that the PSP course project report entitled **“ Title of the project”** is a record of bonafide work carried out by the student(s) E.Manikanta, P.Dharma Reddy & P.Snehith Kumar bearing roll number(s) 2203A51474, 2203A51510 & 2203A51514 of Computer Science and Artificial Intelligence department during the academic year 2022-23.

**Supervisor**

(V. Shobha Rani)

**INDEX**

**Sl. No Title Page No.**

1. Problem statement 4
2. Module-wise description 6
3. Knowledge required to develop the project 8
4. Source code (.c file code followed by .h file code) 9
5. Results 17

# PROBLEM STATEMENT:

Develop a C Application to a Grocery Storage program. The program should allow users to perform various operations related to managing inventory and shopping cart. The program should maintain an inventory of products available in the grocerymart. Each product should have a unique ID, name, price, and quantity.

Provide the functionality for below mentioned:

1. Read ‘n’ items details dynamically.
2. Include the following details :
   * Add Product
   * Display Inventory
   * Search Product
   * Add to Cart
   * Remove Item from Cart
   * Display Cart
   * Generate Inventory Report
   * Checkout
3. Search and Add items details :
   * Add Product
   * Search Product
   * Add to Cart
   * Remove Item from Cart
   * Display Cart
   * Checkout
4. Print the total checkout bill.

# MODULES:

In this application all variables and structure are declared globally so that these variables and structure members can be accessed throughout the program at any function call. We can choose any function by using function calls which are declared in switch-case. This program is responsible for implementing a simple grocerystore management program. It allows users to perform various operations related to managing inventory and a shopping cart.

In this application five modules are used.

1. Add Product

In this module the application asks the person what product/item he/she is adding and its id, name, quantity and price.

1. Search

In this module the user can search if the item is present in Inventory by giving the product name

1. Add/Remove Item from Cart

The program allows users to add items to their shopping cart. The program allows users to remove items from their shopping cart. They can be able to specify the product ID of the item to be removed and the specified item will be removed from the cart.

1. Generate Inventory Report

The program provides an option to generate an inventory report. The report displays the ID, name, price, quantity, and value of each product in the inventory, as well as the total value of the inventory.

1. Checkout

The program allows users to checkout from their shopping cart. A checkout summary will be displayed, showing the details of each item in the cart, as well as the total amount to be paid.

**KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION**

* + Control Statements (if, if-else, switch)
  + Loop Statements (while/do while, for)
  + Arrays (1D/2D-arrays)
  + Strings (Strings and Table of strings) and its functions (strcpy, strcmp)
  + Functions (Any type of user defined functions)
  + Structure (structures and nested structures)
  + Pointers (pointer to strings and pointers to structures)

**SOURCE CODE [.C FILE]:**

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define MAX\_PRODUCTS 100

typedef struct {

int id;

char name[50];

double price;

int quantity;

} Product;

typedef struct {

Product products[MAX\_PRODUCTS];

int count;

} Inventory;

typedef struct {

int productId;

int quantity;

} CartItem;

typedef struct {

CartItem items[MAX\_PRODUCTS];

int count;

double totalAmount;

} ShoppingCart;

void initializeInventory(Inventory\* inventory) {

inventory->count = 0;

}

void addProduct(Inventory\* inventory, int id, const char\* name, double price, int quantity) {

if (inventory->count >= MAX\_PRODUCTS) {

printf("Inventory is full. Cannot add more products.\n");

return;

}

Product newProduct;

newProduct.id = id;

snprintf(newProduct.name, sizeof(newProduct.name), "%s", name);

newProduct.price = price;

newProduct.quantity = quantity;

inventory->products[inventory->count] = newProduct;

inventory->count++;

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

}

void displayInventory(const Inventory\* inventory) {

printf("Inventory:\n");

int i;

for ( i = 0; i < inventory->count; i++) {

Product product = inventory->products[i];

printf("ID: %d, Name: %s, Price: %.2f, Quantity: %d\n", product.id, product.name, product.price, product.quantity);

}

}

bool searchProduct(const Inventory\* inventory, int id, Product\* product) {

int i;

for ( i = 0; i < inventory->count; i++) {

if (inventory->products[i].id == id) {

\*product = inventory->products[i];

return true;

}

}

return false;

}

void initializeCart(ShoppingCart\* cart) {

cart->count = 0;

cart->totalAmount = 0.0;

}

void addToCart(ShoppingCart\* cart, const Inventory\* inventory, int productId, int quantity) {

if (cart->count >= MAX\_PRODUCTS) {

printf("Cart is full. Cannot add more items.\n");

return;

}

Product product;

if (!searchProduct(inventory, productId, &product)) {

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

return;

}

if (product.quantity < quantity) {

printf("Insufficient quantity available in the inventory.\n");

return;

}

CartItem newItem;

newItem.productId = productId;

newItem.quantity = quantity;

cart->items[cart->count] = newItem;

cart->count++;

cart->totalAmount += (product.price \* quantity);

printf("Item added to cart successfully.\n");

}

void removeItemFromCart(ShoppingCart\* cart, const Inventory\* inventory, int productId) {

int foundIndex = -1;

double removedAmount = 0.0;

int i;

for ( i = 0; i < cart->count; i++) {

if (cart->items[i].productId == productId) {

foundIndex = i;

break;

}

}

if (foundIndex == -1) {

printf("Item not found in the cart.\n");

return;

}

CartItem removedItem = cart->items[foundIndex];

Product product;

if (searchProduct(inventory, productId, &product)) {

removedAmount = (product.price \* removedItem.quantity);

product.quantity += removedItem.quantity;

}

for ( i = foundIndex; i < cart->count - 1; i++) {

cart->items[i] = cart->items[i + 1];

}

cart->count--;

cart->totalAmount -= removedAmount;

printf("Item removed from cart successfully.\n");

}

void displayCart(const ShoppingCart\* cart, const Inventory\* inventory) {

printf("Shopping Cart:\n");

int i;

for ( i = 0; i < cart->count; i++) {

CartItem item = cart->items[i];

Product product;

if (searchProduct(inventory, item.productId, &product)) {

printf("Product: %s, Price: %.2f, Quantity: %d, Total: %.2f\n",

product.name, product.price, item.quantity, product.price \* item.quantity);

}

}

printf("Total Amount: %.2f\n", cart->totalAmount);

}

void checkout(ShoppingCart\* cart, Inventory\* inventory) {

printf("Checkout Summary:\n");

displayCart(cart, inventory);

if (cart->totalAmount <= 0.0) {

printf("No items in the cart. Nothing to checkout.\n");

return;

}

char confirmation;

printf("Confirm checkout (y/n): ");

scanf(" %c", &confirmation);

if (confirmation == 'y' || confirmation == 'Y') {

printf("Checkout completed. Total amount to pay: %.2f\n", cart->totalAmount);

int i;

for ( i = 0; i < cart->count; i++) {

CartItem item = cart->items[i];

Product\* product;

if (searchProduct(inventory, item.productId, &product)) {

product->quantity -= item.quantity;

}

}

initializeCart(cart);

} else {

printf("Checkout canceled.\n");

}

}

void generateReport(const Inventory\* inventory) {

printf("Inventory Report:\n");

double totalValue = 0.0;

int i;

for ( i = 0; i < inventory->count; i++) {

Product product = inventory->products[i];

double value = product.price \* product.quantity;

totalValue += value;

printf("ID: %d, Name: %s, Price: %.2f, Quantity: %d, Value: %.2f\n",

product.id, product.name, product.price, product.quantity, value);

}

printf("Total Inventory Value: %.2f\n", totalValue);

}

int main() {

Inventory inventory;

initializeInventory(&inventory);

ShoppingCart cart;

initializeCart(&cart);

int choice;

int id;

char name[50];

double price;

int quantity;

do {

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

printf("1. Add Product\n");

printf("2. Display Inventory\n");

printf("3. Search Product\n");

printf("4. Add to Cart\n");

printf("5. Remove Item from Cart\n");

printf("6. Display Cart\n");

printf("7. Checkout\n");

printf("8. Generate Inventory Report\n");

printf("9. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter product ID: ");

scanf("%d", &id);

printf("Enter product name: ");

scanf("%s", name);

printf("Enter product price: ");

scanf("%lf", &price);

printf("Enter product quantity: ");

scanf("%d", &quantity

);

addProduct(&inventory, id, name, price, quantity);

break;

case 2:

displayInventory(&inventory);

break;

case 3:

printf("Enter product ID: ");

scanf("%d", &id);

Product product;

if (searchProduct(&inventory, id, &product)) {

printf("Product found:\n");

printf("ID: %d, Name: %s, Price: %.2f, Quantity: %d\n",

product.id, product.name, product.price, product.quantity);

} else {

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

}

break;

case 4:

printf("Enter product ID: ");

scanf("%d", &id);

printf("Enter quantity: ");

scanf("%d", &quantity);

addToCart(&cart, &inventory, id, quantity);

break;

case 5:

printf("Enter product ID: ");

scanf("%d", &id);

removeItemFromCart(&cart, &inventory, id);

break;

case 6:

displayCart(&cart, &inventory);

break;

case 7:

checkout(&cart, &inventory);

break;

case 8:

generateReport(&inventory);

break;

case 9:

printf("Exiting program.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

break;

}

} while (choice != 9);

return 0;

}

**RESULTS:**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 1**

**Enter product ID: 1**

**Enter product name: bread**

**Enter product price: 35**

**Enter product quantity: 23**

**Product added successfully.**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 1**

**Enter product ID: 2**

**Enter product name: milk**

**Enter product price: 30**

**Enter product quantity: 25**

**Product added successfully.**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 2**

**Inventory:**

**ID: 1, Name: bread, Price: 35.00, Quantity: 23**

**ID: 2, Name: milk, Price: 30.00, Quantity: 25**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 3**

**Enter product ID: 2**

**Product found:**

**ID: 2, Name: milk, Price: 30.00, Quantity: 25**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 8**

**Inventory Report:**

**ID: 1, Name: bread, Price: 35.00, Quantity: 23, Value: 805.00**

**ID: 2, Name: milk, Price: 30.00, Quantity: 25, Value: 750.00**

**Total Inventory Value: 1555.00**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 4**

**Enter product ID: 2**

**Enter quantity: 3**

**Item added to cart successfully.**

**--Grocery Store--**

**1. Add Product**

**2. Display Inventory**

**3. Search Product**

**4. Add to Cart**

**5. Remove Item from Cart**

**6. Display Cart**

**7. Checkout**

**8. Generate Inventory Report**

**9. Exit**

**Enter your choice: 7**

**Checkout Summary:**

**Shopping Cart:**

**Product: milk, Price: 30.00, Quantity: 3, Total: 90.00**

**Total Amount: 90.00**

**Confirm checkout (y/n): y**

**Checkout completed. Total amount to pay: 90.00**