**INVENTORY ITEMS**

implement crud operations (create read update delete)for inventory items and optimize stock

Code

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_ITEMS 100

#define NAME\_LEN 50

typedef struct {

int id;

char name[NAME\_LEN];

int quantity;

} InventoryItem;

InventoryItem inventory[MAX\_ITEMS];

int itemCount = 0;

void createItem(int id, const char \*name, int quantity) {

if (itemCount < MAX\_ITEMS) {

inventory[itemCount].id = id;

strncpy(inventory[itemCount].name, name, NAME\_LEN);

inventory[itemCount].quantity = quantity;

itemCount++;

printf("Item created successfully.\n");

} else {

printf("Inventory is full.\n");

}

}

void readItem(int id) {

for (int i = 0; i < itemCount; i++) {

if (inventory[i].id == id) {

printf("ID: %d\nName: %s\nQuantity: %d\n",

inventory[i].id, inventory[i].name, inventory[i].quantity);

return;

}

}

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

}

// Function to update an inventory item by ID

void updateItem(int id, const char \*name, int quantity) {

for (int i = 0; i < itemCount; i++) {

if (inventory[i].id == id) {

if (name != NULL) strncpy(inventory[i].name, name, NAME\_LEN);

if (quantity >= 0) inventory[i].quantity = quantity;

printf("Item updated successfully.\n");

return;

}

}

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

}

void deleteItem(int id) {

for (int i = 0; i < itemCount; i++) {

if (inventory[i].id == id) {

for (int j = i; j < itemCount - 1; j++) {

inventory[j] = inventory[j + 1];

}

itemCount--;

printf("Item deleted successfully.\n");

return;

}

}

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

}

void checkStockLevels() {

for (int i = 0; i < itemCount; i++) {

if (inventory[i].quantity < 10) {

printf("Item %s is below the minimum stock level.\n", inventory[i].name);

}

}

}

int main() {

createItem(1, "Laptop", 50);

createItem(2, "Mouse", 5);

createItem(3, "Keyboard", 15);

createItem(4, "Monitor", 8);

createItem(5, "USB Drive", 30);

printf("\nReading items:\n");

readItem(1);

readItem(2);

readItem(3);

readItem(4);

readItem(5);

printf("\nUpdating Keyboard quantity to 20:\n");

updateItem(3, "Keyboard", 20);

readItem(3);

printf("\nChecking stock levels:\n");

checkStockLevels();

printf("\nDeleting Mouse:\n");

deleteItem(2);

readItem(2);

return 0;

}

Output

Item created successfully.

Item created successfully.

Item created successfully.

Item created successfully.

Item created successfully.

Reading items:

ID: 1

Name: Laptop

Quantity: 50

ID: 2

Name: Mouse

Quantity: 5

ID: 3

Name: Keyboard

Quantity: 15

ID: 4

Name: Monitor

Quantity: 8

ID: 5

Name: USB Drive

Quantity: 30

Updating Keyboard quantity to 20:

Item updated successfully.

ID: 3

Name: Keyboard

Quantity: 20

Checking stock levels:

Item Mouse is below the minimum stock level.

Item Monitor is below the minimum stock level.

Deleting Mouse:

Item deleted successfully.

Item not found.