ASSESSMENT 2

```
/*
Inventory Management System
Specifications:
Variables: Item ID, name, quantity, and price.
Static & Const: Static variable for total items; const for maximum inventory size.
Switch Case: Menu for adding, updating, and viewing inventory.
Looping Statements: Loop through items.
Pointers: Pointer for updating item quantity and price.
Functions: Functions for inventory operations.
Arrays: Store item details.
Structures: Structure for item details.
Nested Structures: Nested structures for item and supplier details.
Unions: Union for storing different measurement units.
Nested Unions: Nested union for unit conversion factors.
Output Expectations: Display inventory list with details.
Menu Example:
1. Add Item
2. Update Item
3. View Inventory
4. Exit

*/


#include <stdio.h>
#include<string.h>
#define MAX_INVENTORY 5

static int total_items=0;

struct supplier{
    char name[50];
    char contact[50];
};


struct item{
    int id;
    char name[50];
    int quantity;
```

```
    float price;
    struct supplier supplier;
};

union unit{
    int count;
    int weight;
    float volume;

};

union ConversionFactor{
    float tokg;
    float toLiters;
};

struct inventory{
    struct item items[MAX_INVENTORY];
    union unit unit[MAX_INVENTORY];
};

void addItem(struct inventory *inventory);
void updateItem(struct inventory *inventory);
void viewInventory(struct inventory *inventory);

int main(){
    struct inventory inventory;
    int choice;
    do{
        printf("\nMenu: \n1.Add Item\n2.Update Item\n3.View Inventory\n4.Exit\n");
        printf("\nEnter your choice\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:
            addItem(&inventory);
            break;
            case 2:
            updateItem(&inventory);
            break;
            case 3:
            viewInventory(&inventory);
            break;
            case 4:
            printf("\n Exiting...\n");
```

```
            break;
            default:
            printf("Invalid Choice\n");

        }
    }while(choice!=4);

}

void addItem(struct inventory *inventory){
    if (total_items>=MAX_INVENTORY){
        printf("Inventory is full");

    }
    struct item *newItem=&inventory-> items[total_items];
    printf("Enter the item id\n");
    scanf("%d", &newItem->id);
    printf("Enter the item name\n");
    scanf("%s", &newItem->name);
    printf("Enter the item quantity\n");
    scanf("%d", &newItem->quantity);
    printf("Enter the item price\n");
    scanf("%f", &newItem->price);
    printf("Enter the supplier name\n");
    scanf("%s", &newItem->supplier.name);
    printf("Enter the supplier contact\n");
    scanf("%s", &newItem->supplier.contact);


    total_items++;

}

void updateItem(struct inventory *inventory){
    int id, found=0;
    printf("Enter item id to update: ");
    scanf("%d",&id);
    for(int i=0;i<total_items;i++){
        if (inventory->items[i].id==id){
            found=1;
            int *quantity=&inventory->items[i].quantity;
            float *price=&inventory->items[i].price;
            printf("Enter new Quantity: ");
            scanf("%d", quantity);
```

```
            printf("Enter new Price: ");
            scanf("%f", price);
            break;

            }

            if(!found){
                printf("Item not found.\n");
            }
    }
}

void viewInventory(struct inventory *inventory){
    if(total_items==0){
        printf("Inventory is empty.\n");
        return;
    }
    printf("\nInventory List:\n");
    for(int i=0; i<total_items; i++){
        printf("Item ID: %d\n",inventory->items[i].id);
        printf("Item Name: %s\n",inventory->items[i].name);
        printf("Quantity: %d\n",inventory->items[i].quantity);
        printf("Price: %.2f\n",inventory->items[i].price);
        printf("Supplier Name: %s\n",inventory->items[i].supplier.name);
        printf("Supplier Contact: %s\n\n",inventory->items[i].supplier.contact);
    }
}
```

Output:

```
Menu:
1.Add Item
2.Update Item
3.View Inventory
4.Exit

Enter your choice
1
Enter the item id
1
Enter the item name
pen
Enter the item quantity
5
Enter the item price
10
Enter the supplier name
nithin
Enter the supplier contact
nithir@gmail.com

Menu:
1.Add Item
2.Update Item
3.View Inventory
4.Exit

Enter your choice
1
Enter the item id
2
Enter the item name
pencil
Enter the item quantity
6
Enter the item price
8
Enter the supplier name
emil
Enter the supplier contact
emil@gamil.com

Menu:
1.Add Item
2.Update Item
3.View Inventory
4.Exit

Enter your choice
3

Inventory List:
Item ID: 1
Item Name: pen
Quantity: 5
Price: 10.00
Supplier Name: nithin
Supplier Contact: nithir@gmail.com

Item ID: 2
Item Name: pencil
Quantity: 6
Price: 8.00
Supplier Name: emil
Supplier Contact: emil@gamil.com


Menu:
1.Add Item
2.Update Item
3.View Inventory
4.Exit

Enter your choice
4

 Exiting...
PS E:\C_tasks\assessment2>
```