



## C language program report: Amazon backend system

Mr. Supanut Sopha	ID : 67070503441
Mr. Supapanya Yathip	ID : 67070503443
Mr. Atip Infa-udom	ID : 67070503446
Mr. Pawarit Wongdaeng	ID : 67070503466
Mr. Piti Srisonkram	ID : 67070503467

This report is submitted as part of the coursework for CPE 100: Computer Programming for Engineering

Master of Engineering in Computer Engineering

Faculty of Engineering

King Mongkut's University of Technology Thonburi

Academic Year 2024

## Chapter 1 Introduction

### 1.1. Topic

Backend system for large e-commerce companies Amazon. This project focuses on developing a comprehensive backend system for a large e-commerce company, like Amazon. The system manages complex inventory management processes with multiple warehouses, tracks product movement between warehouses to maintain proper stock levels, and has a product search system with filters for writing such as quantity, price, or product name. In the customer-facing part, the system supports product sales to ensure that there is enough stock to fulfill customer orders. When a customer places an order, the system identifies the appropriate warehouse to ship the product.

### 1.2. Objectives

Objectives of the C Programming Project for the Backend System of an E-Commerce Company like Amazon:

- 1.2.1. To practice C programming and gain a deeper understanding of the language through project-based learning.
- 1.2.2. To develop an inventory management system that can track stock in multiple warehouses, with product quantities being updated whenever items are added, moved, or sold.
- 1.2.3. To enable customer-facing sales operations by creating a system that ensures product availability for customers making purchases.
- 1.2.4. To automate order fulfillment and shipping by coordinating the delivery of products from the most suitable warehouse to customers, reducing shipping time and cost by selecting the optimal warehouse for order dispatch.
- 1.2.5. To improve data management and file handling by reading and writing data from CSV files for products, orders, and users.
- 1.2.6. To support role-based access (admin and customer roles) by providing separate functionalities for admin users (product management) and customer users (browsing, searching, and purchasing products).

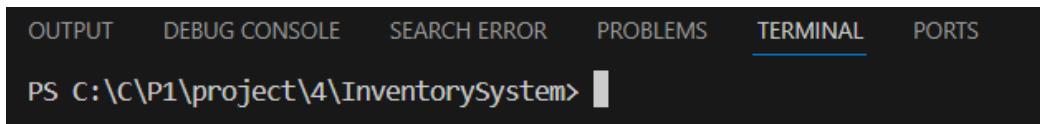
1.2.7. To promote real-time product availability to prevent customers from ordering out-of-stock items.

1.2.8. To provide a user-friendly interface and navigation by developing a simple and easy-to-use text-based interface for customers and admins to interact with the system.

### 1.3. How to make the program work

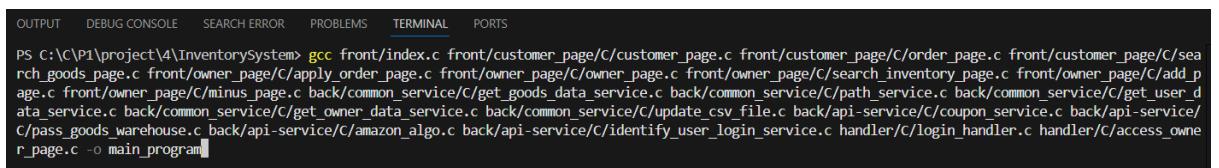
Link Repo Program: <https://github.com/kmood-Sakura/InventorySystem/>

1. After loading the file and opened file in the folder in vs code.
2. Open the terminal page and check if the location we are in is correct or not. Look at the folder in the outermost location of all files. If it is not at the outermost position, use the command “cd..” to exit to the largest folder

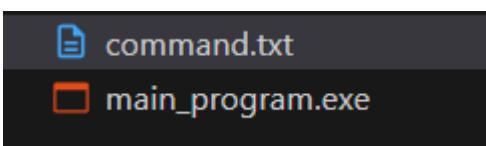


```
OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL PORTS  
PS C:\P1\project\4\InventorySystem> █
```

3. After that, use the command “gcc front/index.c front/customer\_page/C/customer\_page.c front/customer\_page/C/order\_page.c front/customer\_page/C/search\_goods\_page.c front/customer\_page/C/apply\_order\_page.c front/customer\_page/C/search\_inventory\_page.c front/customer\_page/C/add\_page.c front/customer\_page/C/minus\_page.c back/common\_service/C/get\_goods\_data\_service.c back/common\_service/C/path\_service.c back/common\_service/C/get\_user\_data\_service.c back/common\_service/C/get\_owner\_data\_service.c back/common\_service/C/update\_csv\_file.c back/api-service/C/coupon\_service.c back/api-service/C/pass\_goods\_warehouse.c back/api-service/C/identify\_user\_login\_service.c back/api-service/C/amazon\_algo.c back/api-service/C/login\_handler.c handler/C/access\_owner\_page.c -o main\_program” to merge all files together and make them work.



```
OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL PORTS  
PS C:\P1\project\4\InventorySystem> gcc front/index.c front/customer_page/C/customer_page.c front/customer_page/C/order_page.c front/customer_page/C/sea  
rch_goods_page.c front/customer_page/C/apply_order_page.c front/customer_page/C/owner_page.c front/customer_page/C/search_inventory_page.c front/customer_page/C/add_p  
age.c front/customer_page/C/minus_page.c back/common_service/C/get_goods_data_service.c back/common_service/C/path_service.c back/common_service/C/get_user_d  
ata_service.c back/common_service/C/get_owner_data_service.c back/common_service/C/update_csv_file.c back/api-service/C/coupon_service.c back/api-service/  
C/pass_goods_warehouse.c back/api-service/C/amazon_algo.c back/api-service/C/identify_user_login_service.c handler/C/login_handler.c handler/C/access_owne  
r_page.c -o main_program █
```



- Once we have created the file, use the command “./main\_program.exe” is used to command the program to open. After that, the program will be able to run and can be used. When you want to exit the program, click “Ctrl + C” to exit the program.

```
OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL PORTS
PS C:\P1\project\4\InventorySystem> ./main_program.exe
-----
Do you have an account?
y : Yes
n : No
e : Exit
Your choice: █
```

## Chapter 2 Data structure

The project will consist of 5 files used to store different data, which are divided based on the features as follows:

- 2.1 inventory.csv** - Stores all product information in the system, such as product ID, product name, description, price, and available quantity. This information is used for viewing products and performing other related operations. In this file, there will be 5 files in total, divided into folders that will contain the user's main, secondary and main data.
- 2.2 coupon.csv** - Stores information about coupons, such as coupon code, discount percentage, discount amount, and usage status. This data is used to apply discounts for customers when purchasing products.
- 2.3 orders.csv** - This file records customer order information, including the product ID, good ID, user ID, product price, and quantity of the product. All this data is stored as order records, which also serve as a purchase history for users.
- 2.4 user.csv** - This file stores user information such as user ID, name, email, phone number, and password, which are used for logging into the system. Users can browse and purchase products, and those with appropriate permissions can also access the backend system to manage inventory. (2 file)
- 2.5 warehouse.csv** - Records information about warehouses, such as warehouse ID, warehouse name, and the products stored in the warehouse. This data is used to display products specific to each warehouse, allowing for inventory management within individual warehouses.
- 2.6 owner.csv** – This file stores information such as ID, token, and name, which are used to access the backend system for managing products. It also allows users to view all products in the warehouse and is utilized for authentication during system login. (2 file)

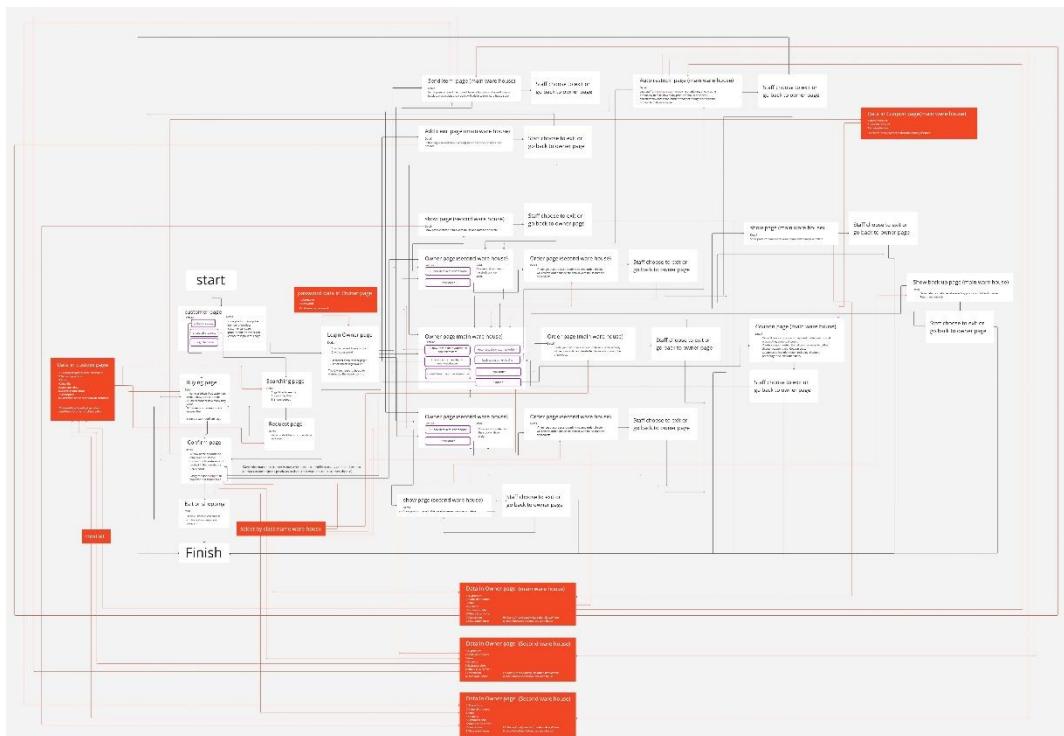
**2.7 remove-list.csv** – This file stores information such as the warehouse ID, warehouse name, product ID, and product quantity. It is used to track and facilitate the transfer of products from one warehouse to another.

**2.8 Transfer.csv** – This file stores the product ID and quantity of products to be moved to the selected warehouse. This information is used when the system automatically updates the inventory by reading the file. (2 file)

**2.9 Subwarehouse.csv** – This file will store the warehouse ID so that it can be retrieved when the warehouse ID is required.

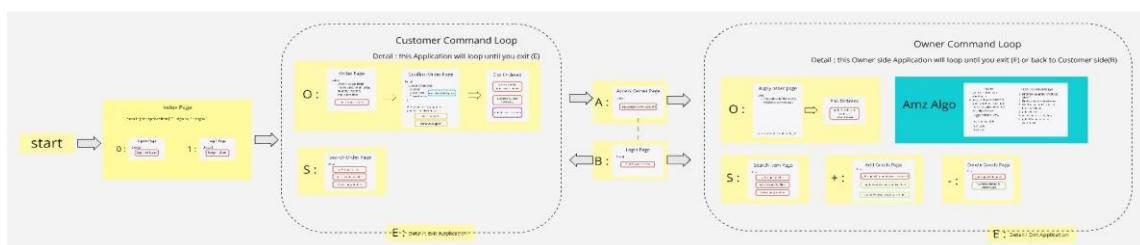
### Chapter 3 Flowchart and diagram

Flowchart C programming design that shows how the entire code works and how each thing works, as shown in the image below.



Link: [https://miro.com/app/board/uXjVL2415g8/?share\\_link\\_id=230614607013](https://miro.com/app/board/uXjVL2415g8/?share_link_id=230614607013)

Diagram C programming design:



Link: [https://miro.com/app/board/uXjVL2415g8/?share\\_link\\_id=230614607013](https://miro.com/app/board/uXjVL2415g8/?share_link_id=230614607013)

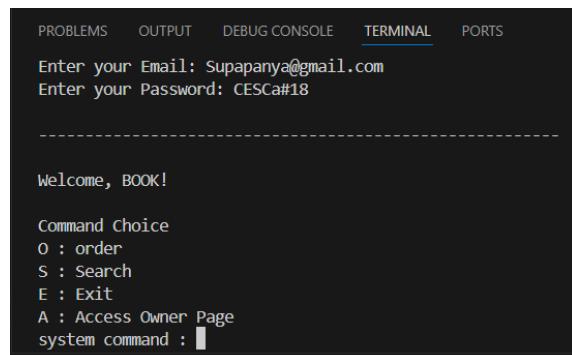
## Chapter 4 Feature explanation and responsibilities

### 4.1 Feature

In terms of coding or programming in C language, it is divided into 4 main parts. The programming features are as follows:

#### 4.1.1 Front-end customer

- I. The system selects the main functions of the first page, with options to order products, search for products, or exit the program and enter the back-end system.



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

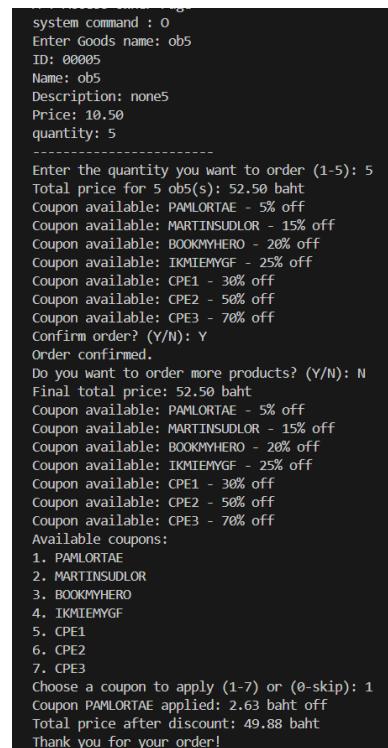
Enter your Email: Supapanya@gmail.com  
Enter your Password: CESCa#18

-----

Welcome, BOOK!

Command Choice  
O : order  
S : Search  
E : Exit  
A : Access Owner Page  
system command : █

- II. Coupon purchasing system and coupon system



```
system command : o
Enter Goods name: ob5
ID: 00005
Name: ob5
Description: none5
Price: 10.50
quantity: 5
-----
Enter the quantity you want to order (1-5): 5
Total price for 5 ob5(s): 52.50 baht
Coupon available: PAMLORTAE - 5% off
Coupon available: MARTINSUDLOR - 15% off
Coupon available: BOOKMYHERO - 20% off
Coupon available: IKMIEEMYGF - 25% off
Coupon available: CPE1 - 30% off
Coupon available: CPE2 - 50% off
Coupon available: CPE3 - 70% off
Confirm order? (Y/N): Y
Order confirmed.
Do you want to order more products? (Y/N): N
Final total price: 52.50 baht
Coupon available: PAMLORTAE - 5% off
Coupon available: MARTINSUDLOR - 15% off
Coupon available: BOOKMYHERO - 20% off
Coupon available: IKMIEEMYGF - 25% off
Coupon available: CPE1 - 30% off
Coupon available: CPE2 - 50% off
Coupon available: CPE3 - 70% off
Available coupons:
1. PAMLORTAE
2. MARTINSUDLOR
3. BOOKMYHERO
4. IKMIEEMYGF
5. CPE1
6. CPE2
7. CPE3
Choose a coupon to apply (1-7) or (0-skip): 1
Coupon PAMLORTAE applied: 2.63 baht off
Total price after discount: 49.88 baht
Thank you for your order!
```

III. Product search system divided into 3 functions

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

system command : s
Search for goods
Choose an option:
(1) Search by ID
(2) Search by Name
(3) View All Products
Enter 1, 2, or 3: 1
Enter goods ID to search: ■
```

IV. Product search system using product ID

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Enter goods ID to search: 00010
Searching for products matching ID: 00010
ID: 00010
Name: ob10
Description: none10
Price: 11.00
quantity: 10
-----
```

V. Product search system using product name

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(3) View All Products
Enter 1, 2, or 3: 2
Enter goods name to search: ob6
Searching for products matching name: ob6
ID: 00006
Name: ob6
Description: none6
Price: 10.60
quantity: 6
-----
system command : ■
```

VI. All product display system

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(3) View All Products
Enter 1, 2, or 3: 3
Displaying All Products:
ID      Name     Description    Price   Quantity
-----
000001  ob1      none1        10.10   1
000002  ob2      none2        10.20   2
000003  ob3      none3        10.30   0
000004  ob4      none4        10.40   4
000005  ob5      none5        10.50   4
000006  ob6      none6        10.60   6
000007  ob7      none7        10.70   7
000008  ob8      none8        10.80   8
000009  ob9      none9        10.90   9
000010  ob10     none10       11.00   10
000011  ob11     none11       11.10   11
000012  ob12     none12       11.20   12
000013  ob13     none13       11.30   13
system command : ■
```

#### 4.1.2 Front-end owner

- I. The system displays a list of all the administrator's products.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Access granted! Welcome, Owner.
Welcome to the Owner Management System.
Here are the available commands:
O: Apply an order. Navigate to the Order Page.
S: Search the warehouse inventory.
+: Add new items to the inventory.
 -: Remove items from the inventory.
E: Exit the system.
A: Access advanced owner settings (placeholder).
Enter your command below:
```

- II. The system displays the history of product orders, with the option to display all products or to sort products according to specified conditions, or to return to the main product management system.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

-: Remove items from the inventory.
E: Exit the system.
A: Access advanced owner settings (placeholder).
Enter your command below:

System command: O
Navigating to the Order Page...

Select an option:
1. Show all data
2. Sort data
3. Return
Enter your choice: [ ]
```

- III. The system displays all information about the previously selected topic.

```
OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL PORTS

Select an option:
1. Show all data
2. Sort data
3. Return
Enter your choice: 1

ID GoodsID UserID Price Quantity
-----
00001 00001 00001 10.1 1
00002 00002 00002 10.2 1
00003 00003 00003 10.3 1
00004 00004 00004 10.4 1
00005 00005 00005 10.5 1
00006 00006 00006 10.6 1

Select an option:
1. Show all data
2. Sort data
3. Return
Enter your choice: [ ]
```

- IV. Product display system with filter selection to display products according to the previously selected topic.

```

OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL ...
00006      00006      00006      10.6      1
Select an option:
1. Show all data
2. Sort data
3. Return
Enter your choice: 2

Select a sorting option:
1. Sort by Price (High to Low)
2. Sort by Price (Low to High)
3. Sort by Quantity (High to Low)
4. Sort by Quantity (Low to High)
Enter your choice: 1

ID      GoodsID     UserID    Price    Quantity
-----
00006    00006    00006    10.60    1
00005    00005    00005    10.50    1
00004    00004    00004    10.40    1
00003    00003    00003    10.30    1
00002    00002    00002    10.20    1
00001    00001    00001    10.10    1

Select an option:
1. Show all data
2. Sort data
3. Return
Enter your choice: 

```

- V. The system for adding products has 2 types: 1. adding by entering products manually and 2. Adding by reading a .csv file.

```

OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL ...
System command: +
Adding new items to the inventory...

Choose a sorting option:
1. Add manually
2. Add using .csv file

```

- VI. Warehouse trading system.

```

OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL ...
Enter your choice: 3

System command: s
Searching the warehouse inventory...
Searching the warehouse inventory...
Searching the warehouse inventory...

Searching the warehouse inventory...
Searching the warehouse inventory...

Choose an option:
1. Show all goods
2. Sort goods
3. Exit
Enter your choice: 

```

VII. Manual product adding system, can select warehouse ID

```
OUTPUT TERMINAL ...  
  
Choose a sorting option:  
1. Add manually  
2. Add using .csv file  
1  
Please enter the warehouse ID :1  
Please enter the goods ID :00010  
Please enter the quantity :1  
Goods transfer successfully
```

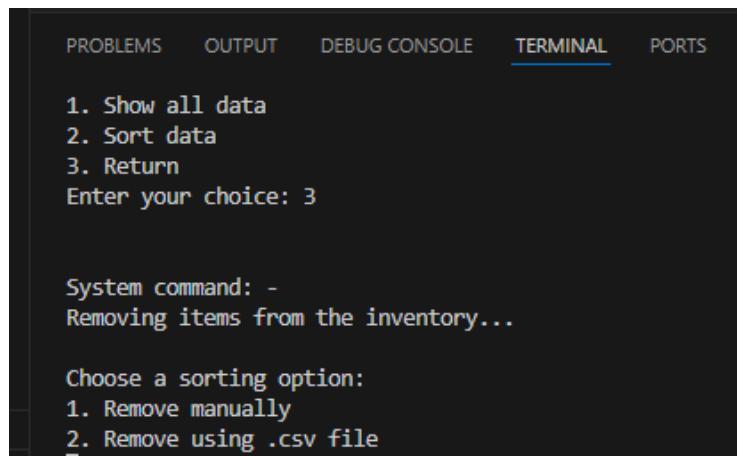
VIII. The system adds products through a .csv file to the secondary warehouse.

```
OUTPUT DEBUG CONSOLE SEARCH ERROR PROBLEMS TERMINAL PORTS  
  
System command: +  
Adding new items to the inventory...  
  
Choose a sorting option:  
1. Add manually  
2. Add using .csv file  
2  
path add file : back/warehouse/1/transfer.csv  
All items were successfully transferred to the warehouse.  
  
System command: []
```

File sample

```
command.txt inventory.csv transfer.csv X  
back > warehouse > 1 > transfer.csv > data  
1 goodsID,quantity  
2 00001,3  
3 00002,3  
4 00003,3
```

IX. Product deletion system



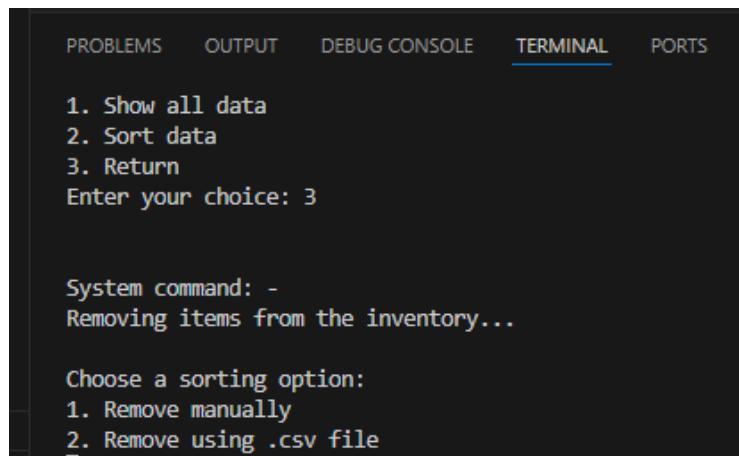
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

1. Show all data
2. Sort data
3. Return
Enter your choice: 3

System command: -
Removing items from the inventory...

Choose a sorting option:
1. Remove manually
2. Remove using .csv file
```

X. Manual or precise inventory removal system



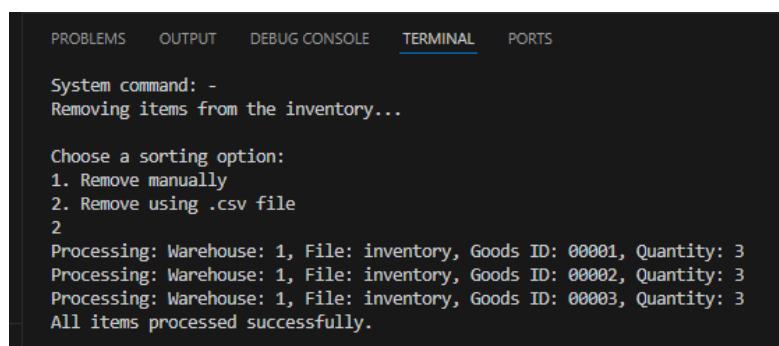
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

1. Show all data
2. Sort data
3. Return
Enter your choice: 3

System command: -
Removing items from the inventory...

Choose a sorting option:
1. Remove manually
2. Remove using .csv file
```

XI. Product deletion system by reading through .csv file

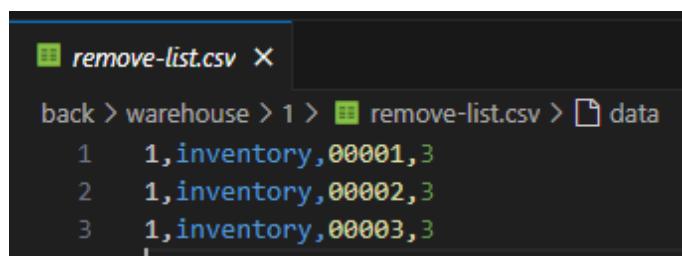


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

System command: -
Removing items from the inventory...

Choose a sorting option:
1. Remove manually
2. Remove using .csv file
2
Processing: Warehouse: 1, File: inventory, Goods ID: 00001, Quantity: 3
Processing: Warehouse: 1, File: inventory, Goods ID: 00002, Quantity: 3
Processing: Warehouse: 1, File: inventory, Goods ID: 00003, Quantity: 3
All items processed successfully.
```

File sample



```
remove-list.csv X

back > warehouse > 1 > remove-list.csv > data
1    1,inventory,00001,3
2    1,inventory,00002,3
3    1,inventory,00003,3
```

- XII. Algorithmic system for adding products to the warehouse by calculating the products before writing the CSV file.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

System command: +
Adding new items to the inventory...

Choose a sorting option:
1. Add manually
2. Add using .csv file
3.Using Amazon Algorithm
3
count : 2
Updated stock file created: back/warehouse/1/transfer.csv
Updated stock file created: back/warehouse/2/transfer.csv

System command: █

```

#### 4.1.3 Back-end

##### A. Common service

###### I. Get goods data

It can read product data in a CSV file and send values to the code that can be called in the form of a map of struct goods and can display products according to the CSV file.

```

get_goods_data_service.c
back> cd ..> cd ..> cd get_goods_data_service> gedit get_goods_data_service.c
1 //include <stdio.h>
2 //include <string.h>
3 //include <get_goods_data_service.h>
4
5 // Function to parse one CSV line into a GOODS struct
6 int parseGOODSCSVLine(char *line, GOODS *item) {
7     char *token;
8     int field = 0;
9
10    // Skip empty lines
11    if (*strtol(line, 0, 10) <= 0) return 0;
12
13    token = strtok(line, ",");
14    // Remove trailing character if present
15    if (line[strlen(line) - 1] == '\n') {
16        line[strlen(line) - 1] = '\0';
17    }
18
19    // Get first token (ID)
20    token = strtok(line, ",");
21    while (token != NULL && field < 5) { // 5 fields in GOODS struct
22        if (token[0] == '\"') {
23            if (token[1] == '\"') {
24                token++;
25            }
26            token[strlen(token)-1] = '\0';
27        }
28        field++;
29    }
30
31    if (field < 5) {
32        item->id = 0;
33        item->name = "Unknown";
34        item->category = "Unknown";
35        item->price = 0.0;
36        item->stock = 0;
37    }
38
39    return 1;
40}
41
42// Main function
43int main() {
44    // Read CSV file
45    FILE *file = fopen("products.csv", "r");
46    if (file == NULL) {
47        perror("Error opening file");
48        exit(1);
49    }
50
51    // Parse CSV file
52    while (fscanf(file, "%d,%s,%s,%f,%d", &item->id, item->name,
53                  item->category, &item->price, &item->stock) == 5) {
54        printf("%d %s %s %.2f %d\n", item->id, item->name,
55               item->category, item->price, item->stock);
56    }
57
58    // Close file
59    fclose(file);
60
61    return 0;
62}

```

ID	Name	Category	Price	Stock
00001	000001	00001	30.5	1
00002	000002	00002	30.4	1
00003	000003	00003	30.5	1
00004	000004	00004	30.5	1

## II. Get user data

It can read user data in a csv file for login and send values to the callable code in the form of a map of struct user and can display user data according to the csv file

```
File Edit Selection View Go Run Terminal Help < > /usr/bin/python

... <--> C:\get_user_data_service.py

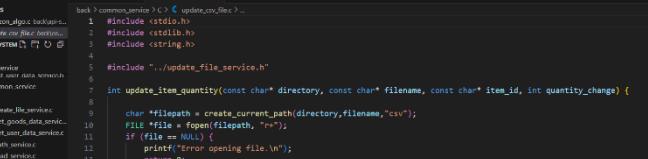
backslash_quote? > C:\get_user_data_service.py ...
1 #include <stdio.h>
2 #include <string.h>
3 #include <string>
4 #include <assert.h>
5 #include <errno.h>
6 #include <sys/types.h>
7 #include <sys/stat.h>
8 #include <fcntl.h>
9 #include <sys/types.h>
10 #include <sys/stat.h>
11 // Function to parse one CSV line into a USER struct
12 int parseCSVLine(char *line, USER *users) {
13     char *token;
14     int field = 0;
15
16     // Skip empty lines
17     if (strlen(line) <= 1) return 0;
18
19     token = strtok(line, ",");
20     while (token != NULL && field < 5) { // 5 fields in USER struct
21         // Remove quotes if present
22         if (*token == '\"') {
23             token++;
24             token[strlen(token)] = '\0';
25         }
26
27         // Get first token (ID)
28         token = strtok(NULL, ",");
29         if (token == NULL) break;
30         users->id = atoi(token);
31
32         // Get name
33         token = strtok(NULL, ",");
34         if (token == NULL) break;
35         users->name = strdup(token);
36
37         // Get age
38         token = strtok(NULL, ",");
39         if (token == NULL) break;
40         users->age = atoi(token);
41
42         // Get gender
43         token = strtok(NULL, ",");
44         if (token == NULL) break;
45         users->gender = strdup(token);
46
47         field++;
48     }
49
50     return field;
51 }
52
53 int main() {
54     FILE *fp;
55     fp = fopen("user_data.csv", "r");
56     if (fp == NULL) {
57         perror("Error opening file");
58         exit(1);
59     }
60
61     USER user;
62     int count = 0;
63
64     while (fscanf(fp, "%d,%s,%d,%s,%d", &user.id, user.name, &user.age, user.gender, &user.sex) == 5) {
65         count++;
66         printf("%d,%s,%d,%s,%d\n", user.id, user.name, user.age, user.gender, user.sex);
67     }
68
69     fclose(fp);
70
71     return 0;
72 }
```

### III. Path service

It is used as a system to reduce the time required for paths and to make it easier to create systems that require paths to access data.

#### IV. Update file service

It is used to update data in various csv files that are called by using the data ID to confirm the location where the data needs to be changed according to the call of this feature.



```
make
```

## B. API service

### I. Coupon service

It will manage the coupon system to check the expiration of the coupon or to edit the coupon usage status which will be called in the customer's ordering system.

```

#include <string>
#include <time.h>
#include "../coupon_service.h"

// Check if the coupon is expired
bool is_coupon_expired(const char* expiryDate) {
    struct tm couponTime = {0};
    time_t currentTime;
    time_t currentTime = time(&currentTime);

    // Parse expiry date string in the format YYYY-MM-DD
    sscanf(expiryDate, "%d-%d-%d", &couponTime.tm_year, &couponTime.tm_mon, &couponTime.tm_mday);
    couponTime.tm_year += 1900; // adjust year
    couponTime.tm_mon -= 1; // adjust month

    time_t couponExpiryTime = mktime(&couponTime);
    return difftime(currentTime, couponExpiryTime) > 0; // Expired if currentTime > couponExpiryTime
}

// Apply discount based on coupon
double apply_coupon_discount_csv(const char* couponCsvPath, const char* couponCode, double totalPrice) {
    FILE *file = fopen(couponCsvPath, "r");
    if (file == NULL) {
        perror("Error opening file");
        exit(1);
    }

    int lineNumber = 0;
    char line[100];
    while (fgets(line, 100, file)) {
        if (lineNumber == 0) {
            lineNumber++;
            continue;
        }

        if (strcmp(couponCode, line) == 0) {
            printf("Coupon applied! Total price: %f\n", totalPrice - (totalPrice * 0.1));
            break;
        }
    }
    fclose(file);
}

```

### II. Pass goods warhouse

It will manage various data in the warehouse to send products to other warehouses or delete products from that warehouse.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../pass_goods_warehouse.h"
#define MAX_LINE_LENGTH 256

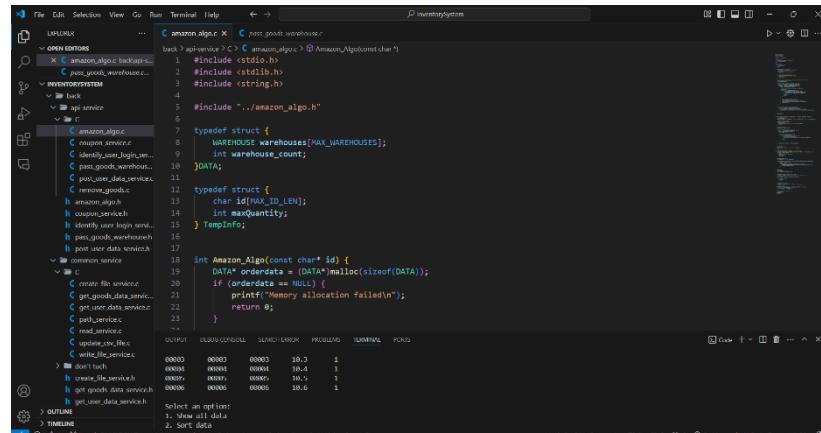
int addGoodsToWarehouse(const char* warehouse_name, const char* main_warehouse_name, const char* filename, const char* goodId) {
    char *mainDir = create_allocate_dir("back/warehouse/", main_warehouse_name);
    if(update_item_quantity(mainDir, filename, goodId, -quantity)){
        printf("Error: not enough item in inventory\n");
        free(mainDir);
        return 0;
    }
    return 1;
}

char *destinationDir = create_allocate_dir("back/warehouse/", warehouse_name);
if(update_item_quantity(destinationDir, filename, goodId, quantity)){
    printf("Error: Can't update to destination warehouse\n");
    free(destinationDir);
    free(mainDir);
    return 0;
}
return 1;
}

int update_item_quantity(char* dir, const char* filename, const char* goodId, int quantity) {
    Select an option:
    1. Show all data
    2. Sort data
    3. Search
    4. Exit
    5. Add new data
    6. Update data
    7. Delete data
    8. Save changes
    9. Load changes
    10. Exit
    11. Help
    12. About
    13. Settings
    14. Preferences
    15. Help
    16. Exit
    17. Help
    18. Help
    19. Help
    20. Help
    21. Help
    22. Help
    23. Help
    24. Help
    25. Help
    26. Help
    27. Help
    28. Help
    29. Help
    30. Help
    31. Help
    32. Help
    33. Help
    34. Help
    35. Help
    36. Help
    37. Help
    38. Help
    39. Help
    40. Help
    41. Help
    42. Help
    43. Help
    44. Help
    45. Help
    46. Help
    47. Help
    48. Help
    49. Help
    50. Help
    51. Help
    52. Help
    53. Help
    54. Help
    55. Help
    56. Help
    57. Help
    58. Help
    59. Help
    60. Help
    61. Help
    62. Help
    63. Help
    64. Help
    65. Help
    66. Help
    67. Help
    68. Help
    69. Help
    70. Help
    71. Help
    72. Help
    73. Help
    74. Help
    75. Help
    76. Help
    77. Help
    78. Help
    79. Help
    80. Help
    81. Help
    82. Help
    83. Help
    84. Help
    85. Help
    86. Help
    87. Help
    88. Help
    89. Help
    90. Help
    91. Help
    92. Help
    93. Help
    94. Help
    95. Help
    96. Help
    97. Help
    98. Help
    99. Help
    100. Help
    101. Help
    102. Help
    103. Help
    104. Help
    105. Help
    106. Help
    107. Help
    108. Help
    109. Help
    110. Help
    111. Help
    112. Help
    113. Help
    114. Help
    115. Help
    116. Help
    117. Help
    118. Help
    119. Help
    120. Help
    121. Help
    122. Help
    123. Help
    124. Help
    125. Help
    126. Help
    127. Help
    128. Help
    129. Help
    130. Help
    131. Help
    132. Help
    133. Help
    134. Help
    135. Help
    136. Help
    137. Help
    138. Help
    139. Help
    140. Help
    141. Help
    142. Help
    143. Help
    144. Help
    145. Help
    146. Help
    147. Help
    148. Help
    149. Help
    150. Help
    151. Help
    152. Help
    153. Help
    154. Help
    155. Help
    156. Help
    157. Help
    158. Help
    159. Help
    160. Help
    161. Help
    162. Help
    163. Help
    164. Help
    165. Help
    166. Help
    167. Help
    168. Help
    169. Help
    170. Help
    171. Help
    172. Help
    173. Help
    174. Help
    175. Help
    176. Help
    177. Help
    178. Help
    179. Help
    180. Help
    181. Help
    182. Help
    183. Help
    184. Help
    185. Help
    186. Help
    187. Help
    188. Help
    189. Help
    190. Help
    191. Help
    192. Help
    193. Help
    194. Help
    195. Help
    196. Help
    197. Help
    198. Help
    199. Help
    200. Help
    201. Help
    202. Help
    203. Help
    204. Help
    205. Help
    206. Help
    207. Help
    208. Help
    209. Help
    210. Help
    211. Help
    212. Help
    213. Help
    214. Help
    215. Help
    216. Help
    217. Help
    218. Help
    219. Help
    220. Help
    221. Help
    222. Help
    223. Help
    224. Help
    225. Help
    226. Help
    227. Help
    228. Help
    229. Help
    230. Help
    231. Help
    232. Help
    233. Help
    234. Help
    235. Help
    236. Help
    237. Help
    238. Help
    239. Help
    240. Help
    241. Help
    242. Help
    243. Help
    244. Help
    245. Help
    246. Help
    247. Help
    248. Help
    249. Help
    250. Help
    251. Help
    252. Help
    253. Help
    254. Help
    255. Help
    256. Help
    257. Help
    258. Help
    259. Help
    260. Help
    261. Help
    262. Help
    263. Help
    264. Help
    265. Help
    266. Help
    267. Help
    268. Help
    269. Help
    270. Help
    271. Help
    272. Help
    273. Help
    274. Help
    275. Help
    276. Help
    277. Help
    278. Help
    279. Help
    280. Help
    281. Help
    282. Help
    283. Help
    284. Help
    285. Help
    286. Help
    287. Help
    288. Help
    289. Help
    290. Help
    291. Help
    292. Help
    293. Help
    294. Help
    295. Help
    296. Help
    297. Help
    298. Help
    299. Help
    300. Help
    301. Help
    302. Help
    303. Help
    304. Help
    305. Help
    306. Help
    307. Help
    308. Help
    309. Help
    310. Help
    311. Help
    312. Help
    313. Help
    314. Help
    315. Help
    316. Help
    317. Help
    318. Help
    319. Help
    320. Help
    321. Help
    322. Help
    323. Help
    324. Help
    325. Help
    326. Help
    327. Help
    328. Help
    329. Help
    330. Help
    331. Help
    332. Help
    333. Help
    334. Help
    335. Help
    336. Help
    337. Help
    338. Help
    339. Help
    340. Help
    341. Help
    342. Help
    343. Help
    344. Help
    345. Help
    346. Help
    347. Help
    348. Help
    349. Help
    350. Help
    351. Help
    352. Help
    353. Help
    354. Help
    355. Help
    356. Help
    357. Help
    358. Help
    359. Help
    360. Help
    361. Help
    362. Help
    363. Help
    364. Help
    365. Help
    366. Help
    367. Help
    368. Help
    369. Help
    370. Help
    371. Help
    372. Help
    373. Help
    374. Help
    375. Help
    376. Help
    377. Help
    378. Help
    379. Help
    380. Help
    381. Help
    382. Help
    383. Help
    384. Help
    385. Help
    386. Help
    387. Help
    388. Help
    389. Help
    390. Help
    391. Help
    392. Help
    393. Help
    394. Help
    395. Help
    396. Help
    397. Help
    398. Help
    399. Help
    400. Help
    401. Help
    402. Help
    403. Help
    404. Help
    405. Help
    406. Help
    407. Help
    408. Help
    409. Help
    410. Help
    411. Help
    412. Help
    413. Help
    414. Help
    415. Help
    416. Help
    417. Help
    418. Help
    419. Help
    420. Help
    421. Help
    422. Help
    423. Help
    424. Help
    425. Help
    426. Help
    427. Help
    428. Help
    429. Help
    430. Help
    431. Help
    432. Help
    433. Help
    434. Help
    435. Help
    436. Help
    437. Help
    438. Help
    439. Help
    440. Help
    441. Help
    442. Help
    443. Help
    444. Help
    445. Help
    446. Help
    447. Help
    448. Help
    449. Help
    450. Help
    451. Help
    452. Help
    453. Help
    454. Help
    455. Help
    456. Help
    457. Help
    458. Help
    459. Help
    460. Help
    461. Help
    462. Help
    463. Help
    464. Help
    465. Help
    466. Help
    467. Help
    468. Help
    469. Help
    470. Help
    471. Help
    472. Help
    473. Help
    474. Help
    475. Help
    476. Help
    477. Help
    478. Help
    479. Help
    480. Help
    481. Help
    482. Help
    483. Help
    484. Help
    485. Help
    486. Help
    487. Help
    488. Help
    489. Help
    490. Help
    491. Help
    492. Help
    493. Help
    494. Help
    495. Help
    496. Help
    497. Help
    498. Help
    499. Help
    500. Help
    501. Help
    502. Help
    503. Help
    504. Help
    505. Help
    506. Help
    507. Help
    508. Help
    509. Help
    510. Help
    511. Help
    512. Help
    513. Help
    514. Help
    515. Help
    516. Help
    517. Help
    518. Help
    519. Help
    520. Help
    521. Help
    522. Help
    523. Help
    524. Help
    525. Help
    526. Help
    527. Help
    528. Help
    529. Help
    530. Help
    531. Help
    532. Help
    533. Help
    534. Help
    535. Help
    536. Help
    537. Help
    538. Help
    539. Help
    540. Help
    541. Help
    542. Help
    543. Help
    544. Help
    545. Help
    546. Help
    547. Help
    548. Help
    549. Help
    550. Help
    551. Help
    552. Help
    553. Help
    554. Help
    555. Help
    556. Help
    557. Help
    558. Help
    559. Help
    560. Help
    561. Help
    562. Help
    563. Help
    564. Help
    565. Help
    566. Help
    567. Help
    568. Help
    569. Help
    570. Help
    571. Help
    572. Help
    573. Help
    574. Help
    575. Help
    576. Help
    577. Help
    578. Help
    579. Help
    580. Help
    581. Help
    582. Help
    583. Help
    584. Help
    585. Help
    586. Help
    587. Help
    588. Help
    589. Help
    590. Help
    591. Help
    592. Help
    593. Help
    594. Help
    595. Help
    596. Help
    597. Help
    598. Help
    599. Help
    600. Help
    601. Help
    602. Help
    603. Help
    604. Help
    605. Help
    606. Help
    607. Help
    608. Help
    609. Help
    610. Help
    611. Help
    612. Help
    613. Help
    614. Help
    615. Help
    616. Help
    617. Help
    618. Help
    619. Help
    620. Help
    621. Help
    622. Help
    623. Help
    624. Help
    625. Help
    626. Help
    627. Help
    628. Help
    629. Help
    630. Help
    631. Help
    632. Help
    633. Help
    634. Help
    635. Help
    636. Help
    637. Help
    638. Help
    639. Help
    640. Help
    641. Help
    642. Help
    643. Help
    644. Help
    645. Help
    646. Help
    647. Help
    648. Help
    649. Help
    650. Help
    651. Help
    652. Help
    653. Help
    654. Help
    655. Help
    656. Help
    657. Help
    658. Help
    659. Help
    660. Help
    661. Help
    662. Help
    663. Help
    664. Help
    665. Help
    666. Help
    667. Help
    668. Help
    669. Help
    670. Help
    671. Help
    672. Help
    673. Help
    674. Help
    675. Help
    676. Help
    677. Help
    678. Help
    679. Help
    680. Help
    681. Help
    682. Help
    683. Help
    684. Help
    685. Help
    686. Help
    687. Help
    688. Help
    689. Help
    690. Help
    691. Help
    692. Help
    693. Help
    694. Help
    695. Help
    696. Help
    697. Help
    698. Help
    699. Help
    700. Help
    701. Help
    702. Help
    703. Help
    704. Help
    705. Help
    706. Help
    707. Help
    708. Help
    709. Help
    710. Help
    711. Help
    712. Help
    713. Help
    714. Help
    715. Help
    716. Help
    717. Help
    718. Help
    719. Help
    720. Help
    721. Help
    722. Help
    723. Help
    724. Help
    725. Help
    726. Help
    727. Help
    728. Help
    729. Help
    730. Help
    731. Help
    732. Help
    733. Help
    734. Help
    735. Help
    736. Help
    737. Help
    738. Help
    739. Help
    740. Help
    741. Help
    742. Help
    743. Help
    744. Help
    745. Help
    746. Help
    747. Help
    748. Help
    749. Help
    750. Help
    751. Help
    752. Help
    753. Help
    754. Help
    755. Help
    756. Help
    757. Help
    758. Help
    759. Help
    760. Help
    761. Help
    762. Help
    763. Help
    764. Help
    765. Help
    766. Help
    767. Help
    768. Help
    769. Help
    770. Help
    771. Help
    772. Help
    773. Help
    774. Help
    775. Help
    776. Help
    777. Help
    778. Help
    779. Help
    780. Help
    781. Help
    782. Help
    783. Help
    784. Help
    785. Help
    786. Help
    787. Help
    788. Help
    789. Help
    790. Help
    791. Help
    792. Help
    793. Help
    794. Help
    795. Help
    796. Help
    797. Help
    798. Help
    799. Help
    800. Help
    801. Help
    802. Help
    803. Help
    804. Help
    805. Help
    806. Help
    807. Help
    808. Help
    809. Help
    810. Help
    811. Help
    812. Help
    813. Help
    814. Help
    815. Help
    816. Help
    817. Help
    818. Help
    819. Help
    820. Help
    821. Help
    822. Help
    823. Help
    824. Help
    825. Help
    826. Help
    827. Help
    828. Help
    829. Help
    830. Help
    831. Help
    832. Help
    833. Help
    834. Help
    835. Help
    836. Help
    837. Help
    838. Help
    839. Help
    840. Help
    841. Help
    842. Help
    843. Help
    844. Help
    845. Help
    846. Help
    847. Help
    848. Help
    849. Help
    850. Help
    851. Help
    852. Help
    853. Help
    854. Help
    855. Help
    856. Help
    857. Help
    858. Help
    859. Help
    860. Help
    861. Help
    862. Help
    863. Help
    864. Help
    865. Help
    866. Help
    867. Help
    868. Help
    869. Help
    870. Help
    871. Help
    872. Help
    873. Help
    874. Help
    875. Help
    876. Help
    877. Help
    878. Help
    879. Help
    880. Help
    881. Help
    882. Help
    883. Help
    884. Help
    885. Help
    886. Help
    887. Help
    888. Help
    889. Help
    890. Help
    891. Help
    892. Help
    893. Help
    894. Help
    895. Help
    896. Help
    897. Help
    898. Help
    899. Help
    900. Help
    901. Help
    902. Help
    903. Help
    904. Help
    905. Help
    906. Help
    907. Help
    908. Help
    909. Help
    910. Help
    911. Help
    912. Help
    913. Help
    914. Help
    915. Help
    916. Help
    917. Help
    918. Help
    919. Help
    920. Help
    921. Help
    922. Help
    923. Help
    924. Help
    925. Help
    926. Help
    927. Help
    928. Help
    929. Help
    930. Help
    931. Help
    932. Help
    933. Help
    934. Help
    935. Help
    936. Help
    937. Help
    938. Help
    939. Help
    940. Help
    941. Help
    942. Help
    943. Help
    944. Help
    945. Help
    946. Help
    947. Help
    948. Help
    949. Help
    950. Help
    951. Help
    952. Help
    953. Help
    954. Help
    955. Help
    956. Help
    957. Help
    958. Help
    959. Help
    960. Help
    961. Help
    962. Help
    963. Help
    964. Help
    965. Help
    966. Help
    967. Help
    968. Help
    969. Help
    970. Help
    971. Help
    972. Help
    973. Help
    974. Help
    975. Help
    976. Help
    977. Help
    978. Help
    979. Help
    980. Help
    981. Help
    982. Help
    983. Help
    984. Help
    985. Help
    986. Help
    987. Help
    988. Help
    989. Help
    990. Help
    991. Help
    992. Help
    993. Help
    994. Help
    995. Help
    996. Help
    997. Help
    998. Help
    999. Help
    1000. Help
    1001. Help
    1002. Help
    1003. Help
    1004. Help
    1005. Help
    1006. Help
    1007. Help
    1008. Help
    1009. Help
    1010. Help
    1011. Help
    1012. Help
    1013. Help
    1014. Help
    1015. Help
    1016. Help
    1017. Help
    1018. Help
    1019. Help
    1020. Help
    1021. Help
    1022. Help
    1023. Help
    1024. Help
    1025. Help
    1026. Help
    1027. Help
    1028. Help
    1029. Help
    1030. Help
    1031. Help
    1032. Help
    1033. Help
    1034. Help
    1035. Help
    1036. Help
    1037. Help
    1038. Help
    1039. Help
    1040. Help
    1041. Help
    1042. Help
    1043. Help
    1044. Help
    1045. Help
    1046. Help
    1047. Help
    1048. Help
    1049. Help
    1050. Help
    1051. Help
    1052. Help
    1053. Help
    1054. Help
    1055. Help
    1056. Help
    1057. Help
    1058. Help
    1059. Help
    1060. Help
    1061. Help
    1062. Help
    1063. Help
    1064. Help
    1065. Help
    1066. Help
    1067. Help
    1068. Help
    1069. Help
    1070. Help
    1071. Help
    1072. Help
    1073. Help
    1074. Help
    1075. Help
    1076. Help
    1077. Help
    1078. Help
    1079. Help
    1080. Help
    1081. Help
    1082. Help
    1083. Help
    1084. Help
    1085. Help
    1086. Help
    1087. Help
    1088. Help
    1089. Help
    1090. Help
    1091. Help
    1092. Help
    1093. Help
    1094. Help
    1095. Help
    1096. Help
    1097. Help
    1098. Help
    1099. Help
    1100. Help
    1101. Help
    1102. Help
    1103. Help
    1104. Help
    1105. Help
    1106. Help
    1107. Help
    1108. Help
    1109. Help
    1110. Help
    1111. Help
    1112. Help
    1113. Help
    1114. Help
    1115. Help
    1116. Help
    1117. Help
    1118. Help
    1119. Help
    1120. Help
    1121. Help
    1122. Help
    1123. Help
    1124. Help
    1125. Help
    1126. Help
    1127. Help
    1128. Help
    1129. Help
    1130. Help
    1131. Help
    1132. Help
    1133. Help
    1134. Help
    1135. Help
    1136. Help
    1137. Help
    1138. Help
    1139. Help
    1140. Help
    1141. Help
    1142. Help
    1143. Help
    1144. Help
    1145. Help
    1146. Help
    1147. Help
    1148. Help
    1149. Help
    1150. Help
    1151. Help
    1152. Help
    1153. Help
    1154. Help
    1155. Help
    1156. Help
    1157. Help
    1158. Help
    1159. Help
    1160. Help
    1161. Help
    1162. Help
    1163. Help
    1164. Help
    1165. Help
    1166. Help
    1167. Help
    1168. Help
    1169. Help
    1170. Help
    1171. Help
    1172. Help
    1173. Help
    1174. Help
    1175. Help
    1176. Help
    1177. Help
    1178. Help
    1179. Help
    1180. Help
    1181. Help
    1182. Help
    1183. Help
    1184. Help
    1185. Help
    1186. Help
    1187. Help
    1188. Help
    1189. Help
    1190. Help
    1191. Help
    1192. Help
    1193. Help
    1194. Help
    1195. Help
    1196. Help
    1197. Help
    1198. Help
    1199. Help
    1200. Help
    1201. Help
    1202. Help
    1203. Help
    1204. Help
    1205. Help
    1206. Help
    1207. Help
    1208. Help
    1209. Help
    1210. Help
    1211. Help
    1212. Help
    1213. Help
    1214. Help
    1215. Help
    1216. Help
    1217. Help
    1218. Help
    1219. Help
    1220. Help
    1221. Help
    1222. Help
    1223. Help
    1224. Help
    1225. Help
    1226. Help
    1227. Help
    1228. Help
    1229. Help
    1230. Help
    1231. Help
    1232. Help
    1233. Help
    1234. Help
    1235. Help
    1236. Help
    1237. Help
    1238. Help
    1239. Help
    1240. Help
    1241. Help
    1242. Help
    1243. Help
    1244. Help
    1245. Help
    1246. Help
    1247. Help
    1248. Help
    1249. Help
    1250. Help
    1251. Help
    1252. Help
    1253. Help
    1254. Help
    1255. Help
    1256. Help
    12
```

### III. Amazon algo

It will be an automatic product replenishment system by calculating the products before shipping or replenishing the products. The calculation method will use the order quantity of every warehouse that has come in to find the ratio of product demand to be sent to the specified warehouses.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "../amazon_algo.h"

typedef struct {
    WAREHOUSE warehouses[MAX_WAREHOUSES];
    int warehouse_count;
} DATA;

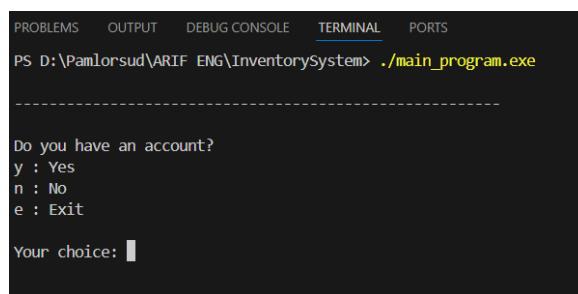
typedef struct {
    char id[MAX_ID_LEN];
    int maxQuantity;
} TempInfo;

int Amazon_Algo(const char* id) {
    DATA* orderdata = (DATA*)malloc(sizeof(DATA));
    if (orderdata == NULL) {
        printf("Memory allocation failed\n");
        return 0;
    }
    ...
}

int main() {
    orderdata->warehouse_count = 5;
    orderdata->warehouses[0].id = "W1";
    orderdata->warehouses[0].maxQuantity = 100;
    orderdata->warehouses[1].id = "W2";
    orderdata->warehouses[1].maxQuantity = 200;
    orderdata->warehouses[2].id = "W3";
    orderdata->warehouses[2].maxQuantity = 300;
    orderdata->warehouses[3].id = "W4";
    orderdata->warehouses[3].maxQuantity = 400;
    orderdata->warehouses[4].id = "W5";
    orderdata->warehouses[4].maxQuantity = 500;
}
```

#### 4.1.4 Logging system

- I. The lock system has a username or if not, it will take you to the user creation page. If there is already one, you can continue using the program or choose to exit.

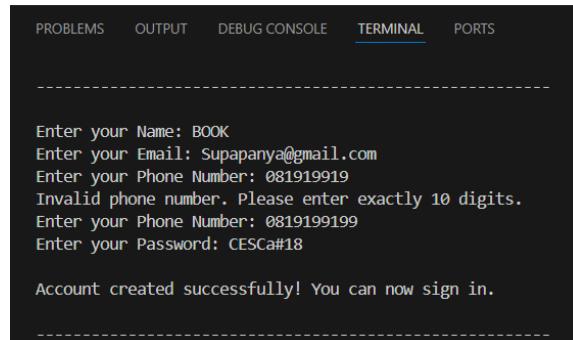


```
PS D:\Pamlorsud\ARIF ENG\InventorySystem> ./main_program.exe

-----
Do you have an account?
y : Yes
n : No
e : Exit

Your choice: [Y]
```

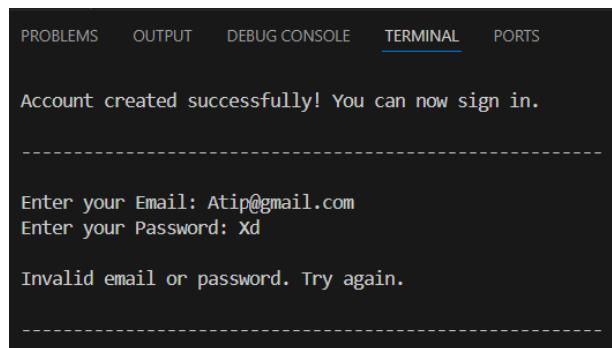
- II. The system generates a username by providing a name, email address, phone number, and password.



```
Enter your Name: BOOK
Enter your Email: Supapanya@gmail.com
Enter your Phone Number: 081919919
Invalid phone number. Please enter exactly 10 digits.
Enter your Phone Number: 0819199199
Enter your Password: CESCa#18

Account created successfully! You can now sign in.
```

- III. Login system using email and password



```

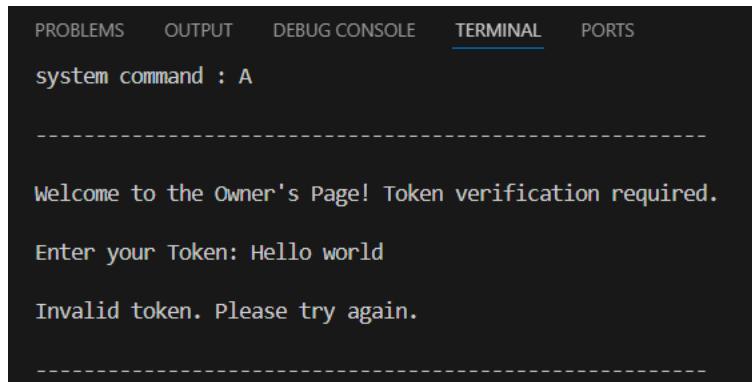
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Account created successfully! You can now sign in.

-----
Enter your Email: Atip@gmail.com
Enter your Password: Xd

Invalid email or password. Try again.
-----
```

- IV. Administrator login system that provides a token to log into the administrator system.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

system command : A

-----
Welcome to the Owner's Page! Token verification required.

Enter your Token: Hello world

Invalid token. Please try again.
-----
```

## 4.2 Responsibilities

The team responsibilities are assigned to each person to design different parts and then combined later, and the responsibilities of each person can be seen in this table.

Summary table of project responsibilities	
Name	Responsibilities
Supanut Sopha	<ul style="list-style-type: none"> <li>- System for moving goods between warehouses</li> <li>- System for managing goods in selected warehouses</li> <li>- System for searching warehouses</li> <li>- System for filtering goods by price or quantity</li> <li>- System for viewing goods</li> </ul>
Supapanya Yathip	<ul style="list-style-type: none"> <li>- Design Amazon Alchemy System</li> <li>- Design the overall flow of the program</li> <li>- Help organize various tasks within the team</li> </ul>
Atip Infa-udom	<ul style="list-style-type: none"> <li>- Purchase system</li> </ul>

	<ul style="list-style-type: none"> <li>- Product search system using product name and product ID</li> <li>- Coupon system</li> <li>- Advance order system using time</li> </ul>
Pawarit Wongdaeng	<ul style="list-style-type: none"> <li>- Login system for both users and owners to access the back office</li> <li>- User registration system</li> </ul>
Piti Srisongkram	<ul style="list-style-type: none"> <li>- Design data within the program to listen to various results</li> <li>- Create functions within the program</li> <li>- Help to improve the code when there are additional questions</li> </ul>

## Chapter 5 Challenges and solutions

During the development of this project, we encountered several challenges, all of which were successfully resolved. The issues as follows:

### 5.1. File Management Challenge

The first issue we encountered was the challenge of managing a large number of files that were not properly organized. This led to confusion, making it difficult to access the correct files. When searching for specific functions or needing to modify code, we had to go through files one by one, which was very time-consuming. The solution was to reorganize the files by creating more detailed folders, such as separating .c and .h files, and giving the files more specific and descriptive names. This made it easier to identify and access the files, ensuring that when changes were needed, we knew exactly where to go and fix it.

### 5.2. File Opening Challenge

The second issue was a continuation of the first. After organizing the files into various folders, we encountered a new problem the inability to locate certain files when attempting to access them. This occurred because some files were nested within subfolders, leading to errors when trying to reference them. The solution was to thoroughly review the folder structure before attempting to access any files. We also implemented a habit of verifying the file paths before use. If this issue arose, we would first check the file path to ensure its accuracy before investigating other possible causes.

## Chapter 6 File path configuration

The file path that is in the program will be as follows that will be used:

- D:\HINewLife\InventorySystem\back\database\inventory.csv
- D:\HI NEW LIFE\InventorySystem\back\database\owner.csv
- D:\HI NEW LIFE\InventorySystem\back\database\user.csv
- D:\HI NEW LIFE\InventorySystem\back\database\warehouse.csv
- D:\HI NEW LIFE\InventorySystem\back\user\1\coupon.csv
- D:\HI NEW LIFE\InventorySystem\back\user\1\inventory.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\1\inventory.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\1\order.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\1\remove-list.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\1\transfer.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\2\inventory.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\2\order.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\2\transfer.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\main\inventory.csv
- D:\HI NEW LIFE\InventorySystem\back\warehouse\main\subwarehouse.csv

## Improved from Feedback

The feature or fixed we have improved will come from feedback from seniors and teachers who want us to adjust and add more features to make the program complete.

- Fixed the issue of entering the wrong amount more than 2 times, causing the program to repeatedly ask the user whether or not they want to buy the product or not.

```
int count = 0;
// Step 4: Quantity Validation
do {
    printf("Enter the quantity you want to order (1-%d): ", foundProduct->quantity);
    if (scanf("%d", &quantity) != 1 || quantity <= 0 || quantity > foundProduct->quantity) {
        if (count >= 1){
            return 0;
        }else{
            count++;
        }
        printf("Invalid input. Try again.\n");
        while (getchar() != '\n'); // Clear input buffer
    }
}
```

- The next thing that was improved was the addition of features such as adding a file log system that will keep a history of all the work done in the program, such as logging in, purchasing products, adding products to keep for viewing history.

```
#include <stdio.h>

#include "../log_service.h"

void COMMON_log(const char* id, const char *type, const char * path_Data, int quantity){
    printf("-----\n");
    printf("object id : %s was %s to %s\n", id, type, path_Data);
    printf("-----\n");
}

void GOODS_log(GOODS *goods, const char *type, const char * path_Data, int quantity){
    printf("-----\n");
    for(int i=0;i<quantity;i++){
        printf("ID : %s\n", goods->id);
        printf("Name : %s\n", goods->name);
        printf("Price : %.2f\n", goods->price);
        printf("Quantity : %d\n", goods->quantity);
        printf("Description : %s\n", goods->description);
    }
    printf("-----was--%s--into--%s-----\n", type, path_Data);
}

void USER_log(USER *user, const char *type, const char * path_Data, int quantity){
    printf("-----\n");
    for(int i=0;i<quantity;i++){
        printf("ID : %s\n", user->id);
        printf("Name : %s\n", user->name);
        printf("Email : %s\n", user->email);
        printf("Phone : %s\n", user->phone);
        printf("Password : %s\n", user->password);
    }
    printf("-----was--%s--into--%s-----\n", type, path_Data);
}
```