LEGENDRY SANNINS MART

- MUHAIB SHAMSHER (CR-22029)
- MUNEEZA BADAR (CR-22022)
- MUHAMMAD HAMZA (CR-22048)

COURSE CODE: CT-260.

INSTRUCTOR: PROF.DR SHARIQ MAHMOOD KHAN

CYBERSECURITY

OBJECT ORIENTED PROGRAM

PROBLEM STATEMENT:

A system that allows customers to browse and purchase grocery items from the comfort of their homes.

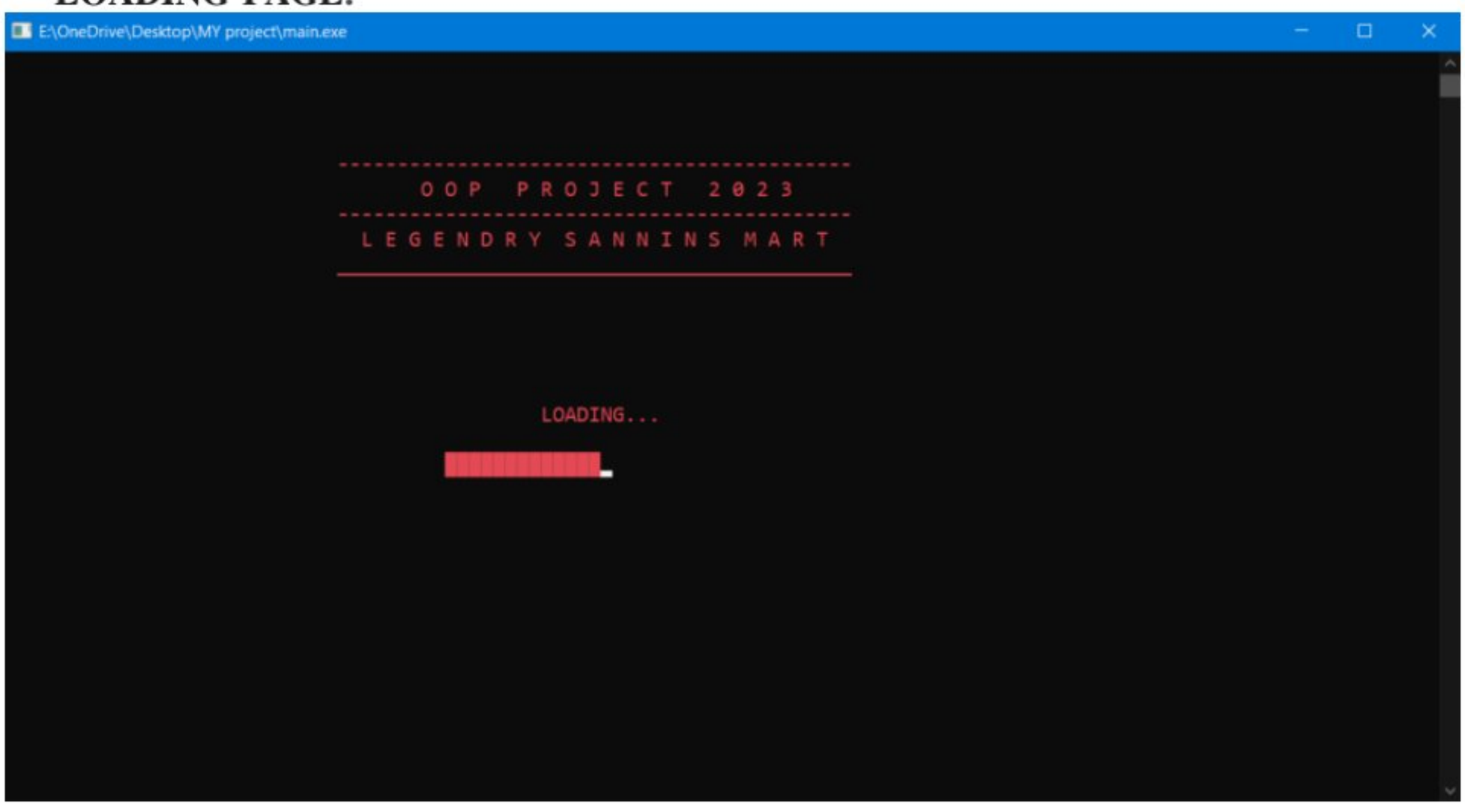
PROJECT DISCRIPTION:

The C++ OOP Online Grocery Mart project is a user-friendly and secure online platform for grocery shopping. The system incorporates key features such as a loading page, registration page, login page, menu page, and validation mechanisms. Upon launching the application, users are greeted with an engaging loading page. New users can register by providing their personal information through the registration page, which includes validation to ensure accurate data entry. Registered users can then securely log in through the login page using their username and password. After successful login, users are directed to the menu page where they can browse various grocery categories and view product details. Throughout the system, input validation is implemented to ensure data integrity and prevent errors. This project aims to provide a seamless and efficient online shopping experience for customers.

STEPS OF SOLUTION:

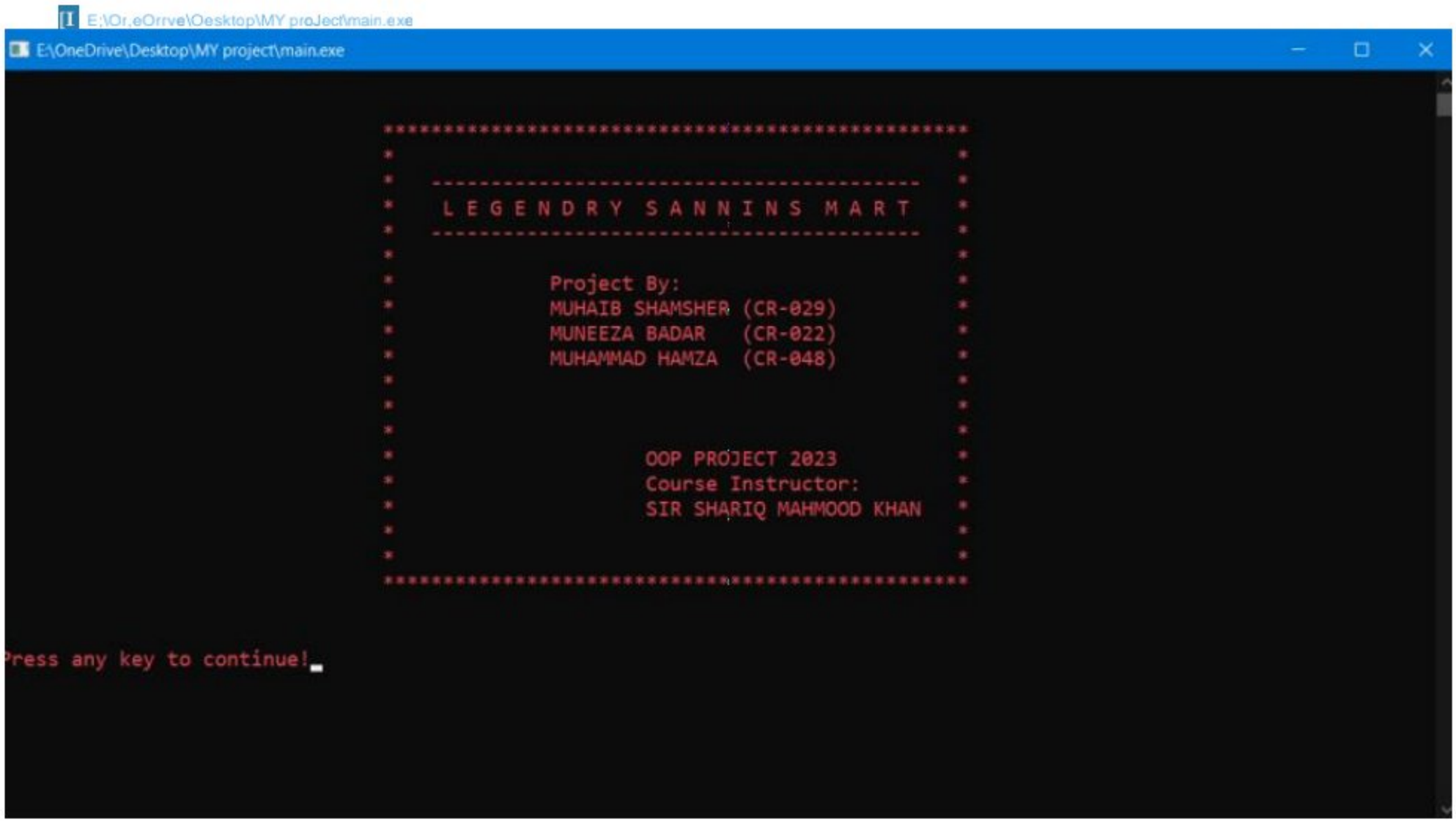
FEATURES OF PROJECT:

LOADING PAGE:



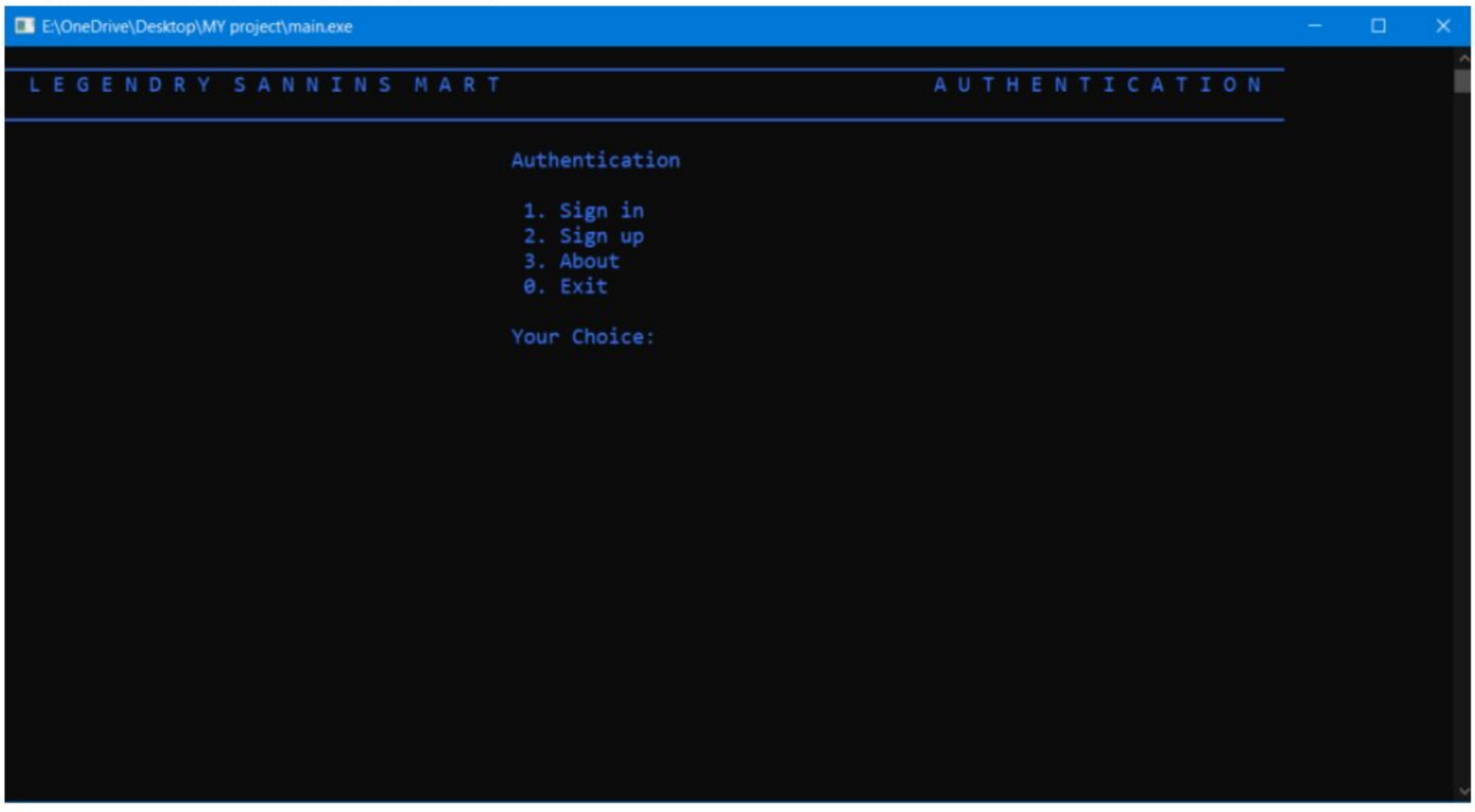
The system greets the user with a visually appealing loading page upon launching the application. It displays a progress indicator or animation to indicate that the system is initializing.

MAIN PAGE:



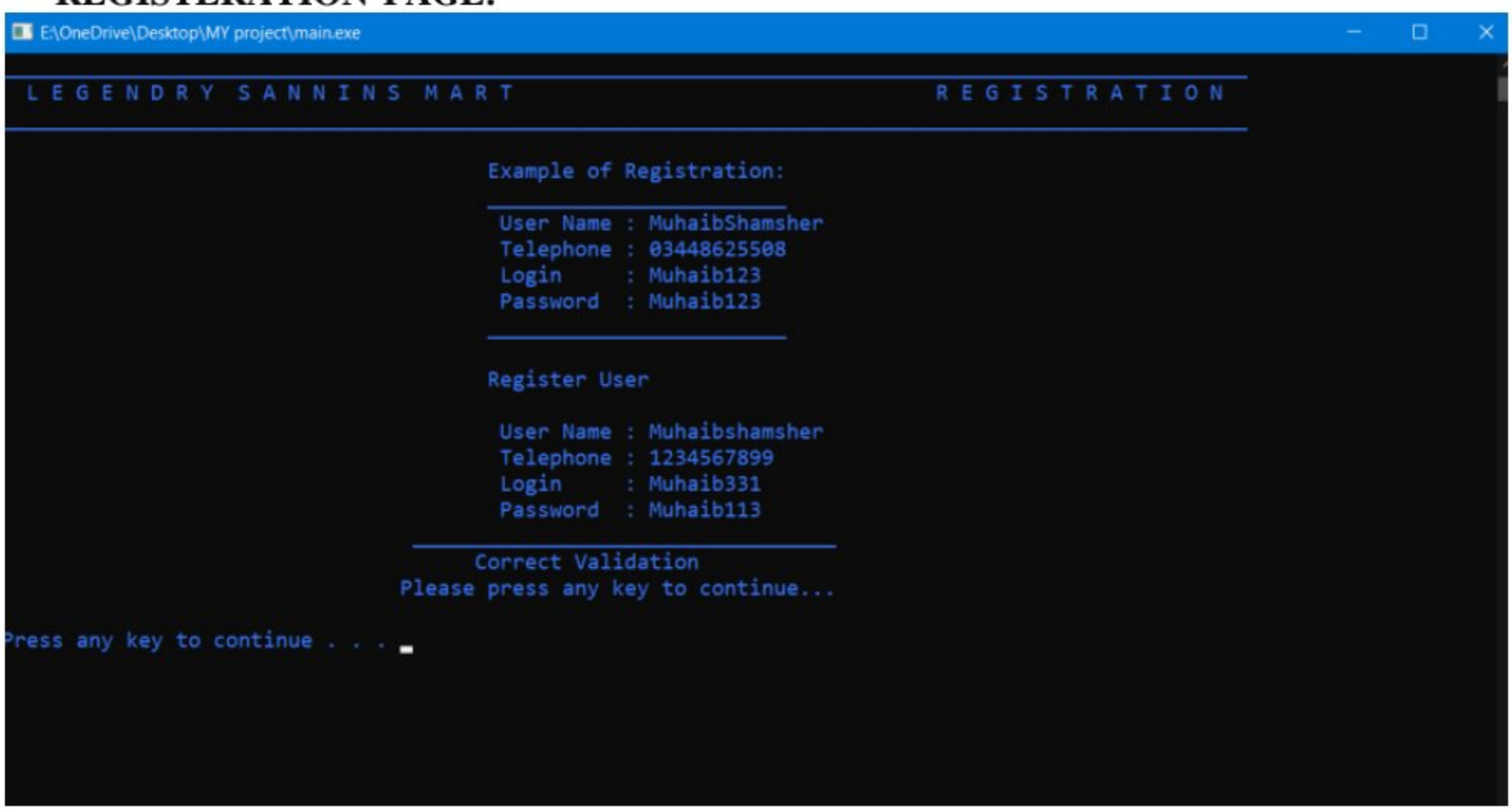
After loading it shows main page and ask user to press any key to continue.

AUTHENTICATION PAGE:



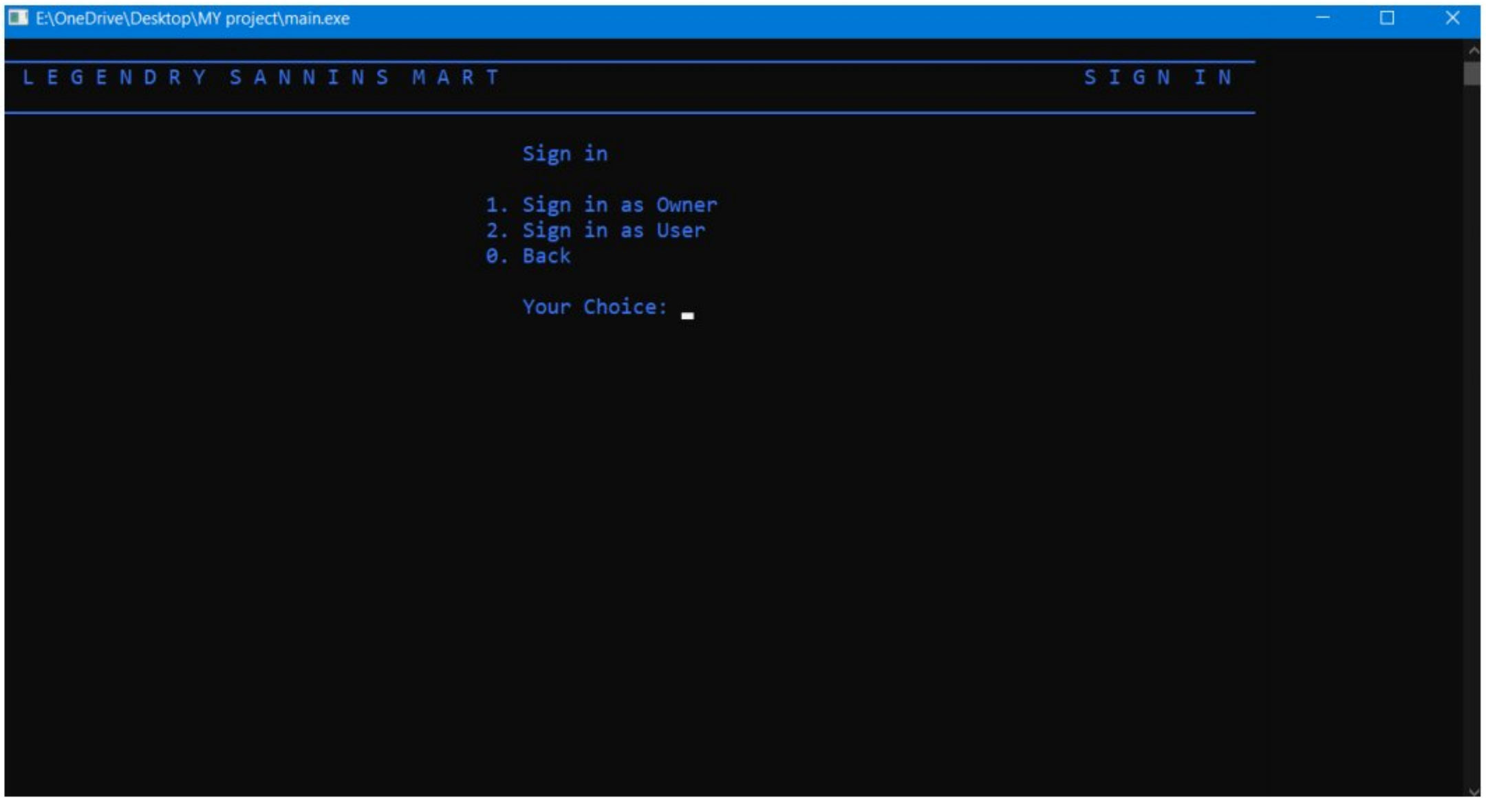
Authentication page ask user weather his account is authenticated or want to register himself. About section will open the main page.

REGISTERATION PAGE:



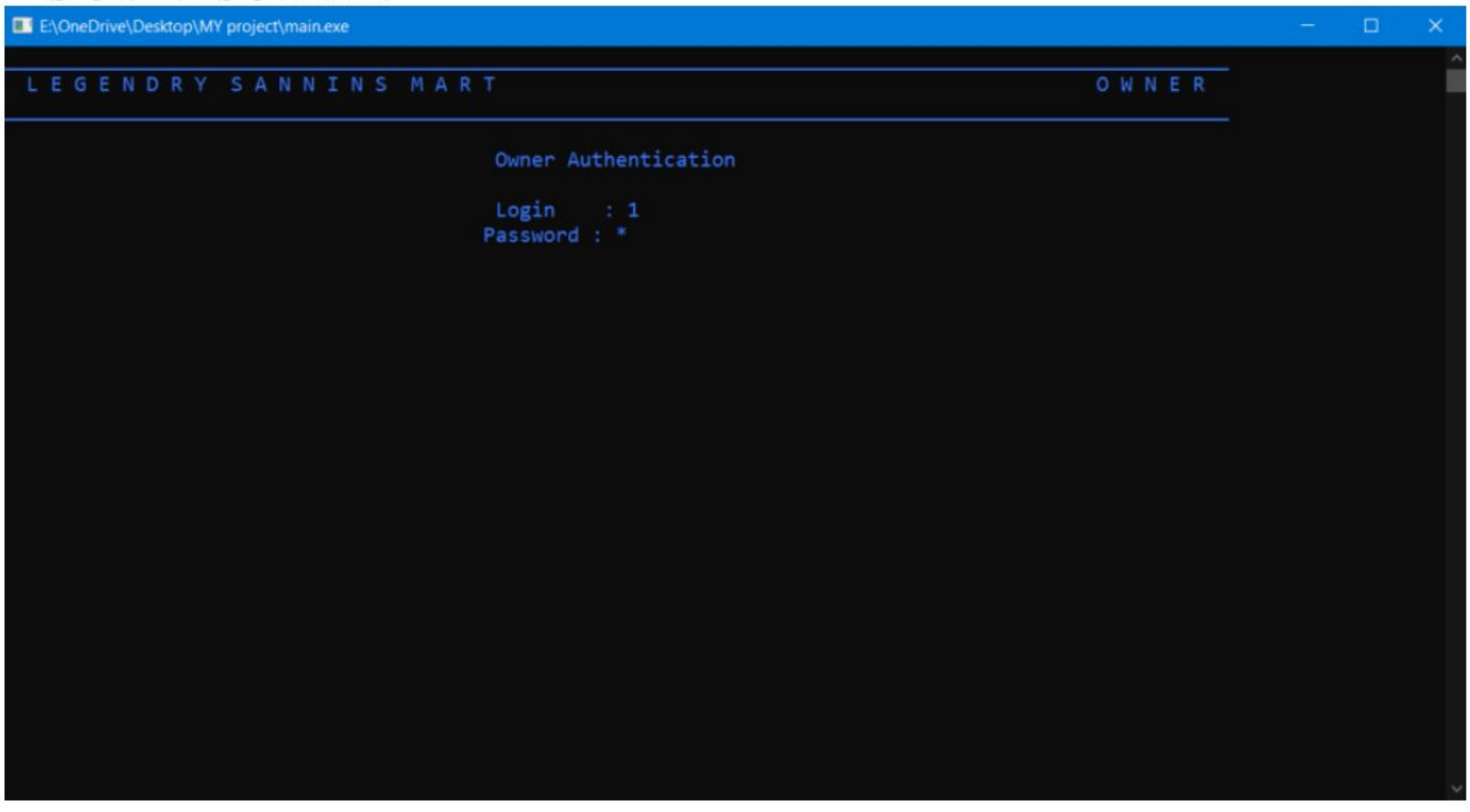
Customers can register for an account by providing their necessary details such as name, contact number, and password. The registration page should include validation checks to ensure the accuracy and completeness of user input.

SIGN IN PAGE:



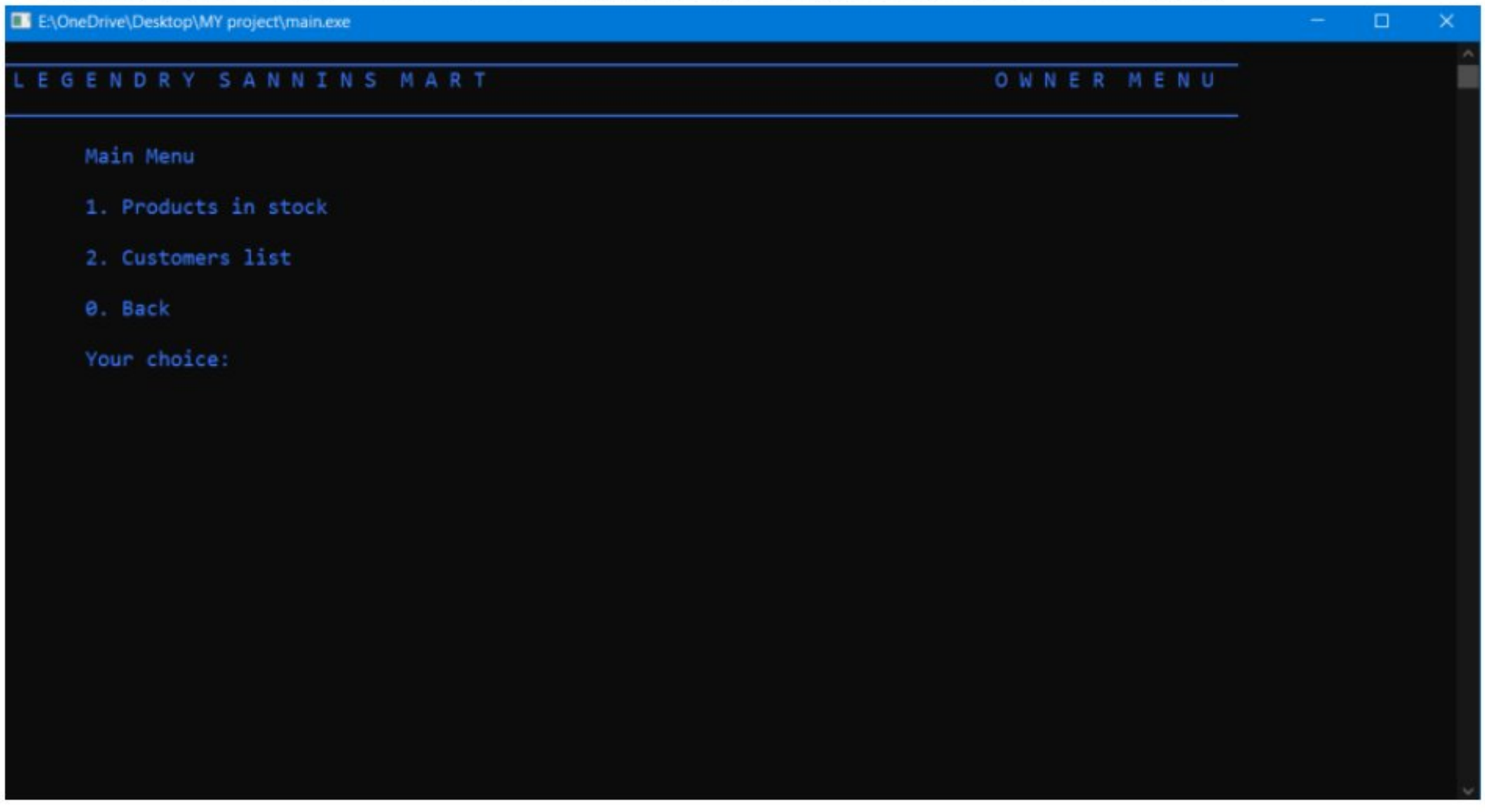
This page ask to sign in as owner or user and you also have a choice to go back.

SIGN IN AS OWNER:



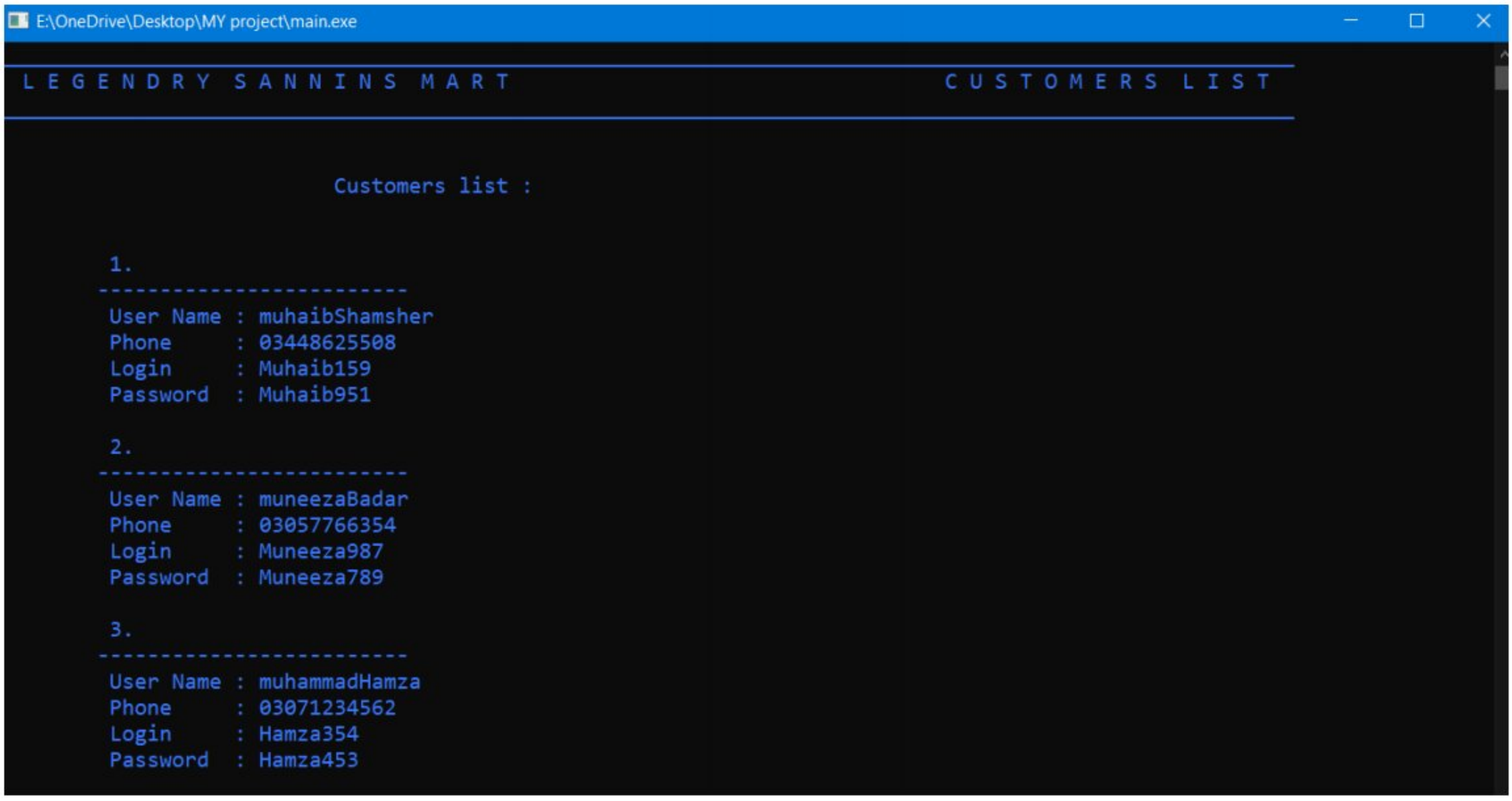
The project includes a separate login page for the owner/administrator of the online grocery mart to access the management features.

OWNER'S PAGE FOR CUSTOMER LIST AND PRODUCT STOCK:



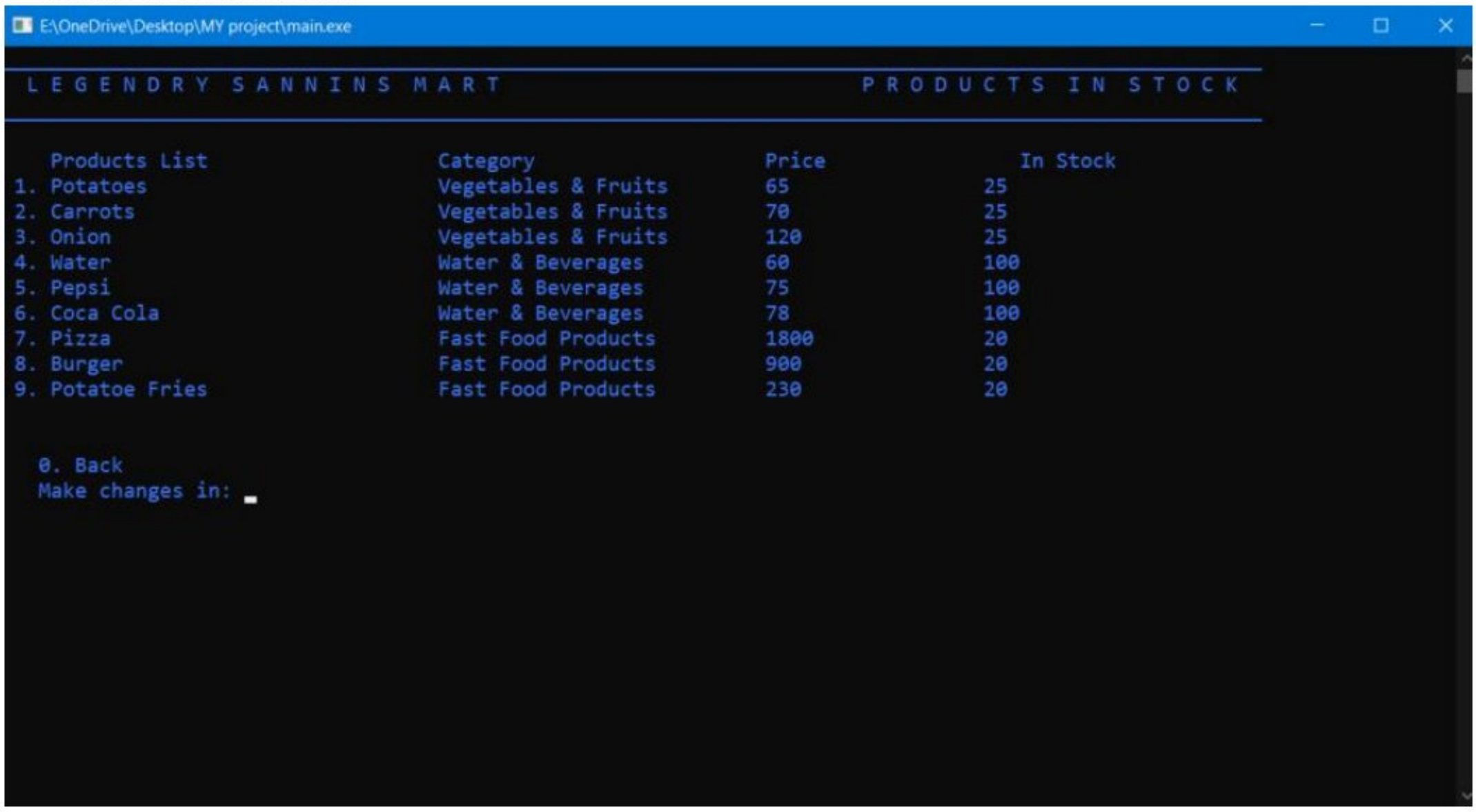
After successful owner login, a page is provided to display the list of registered customers and their relevant information.

CUSTOMER LIST:



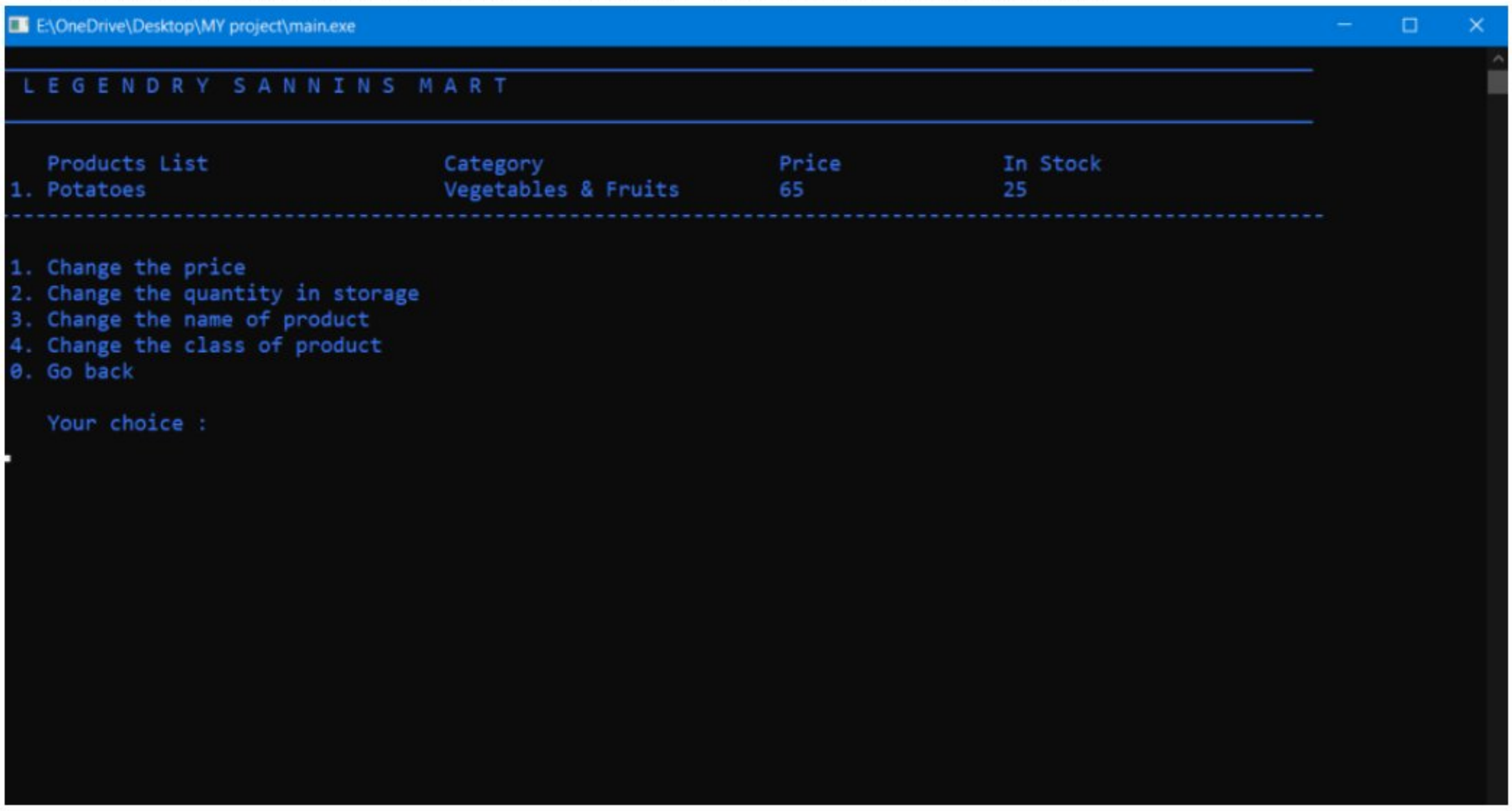
The owner can view and manage customer accounts, such as updating customer details or deactivating accounts.

PRODUCT STOCK:



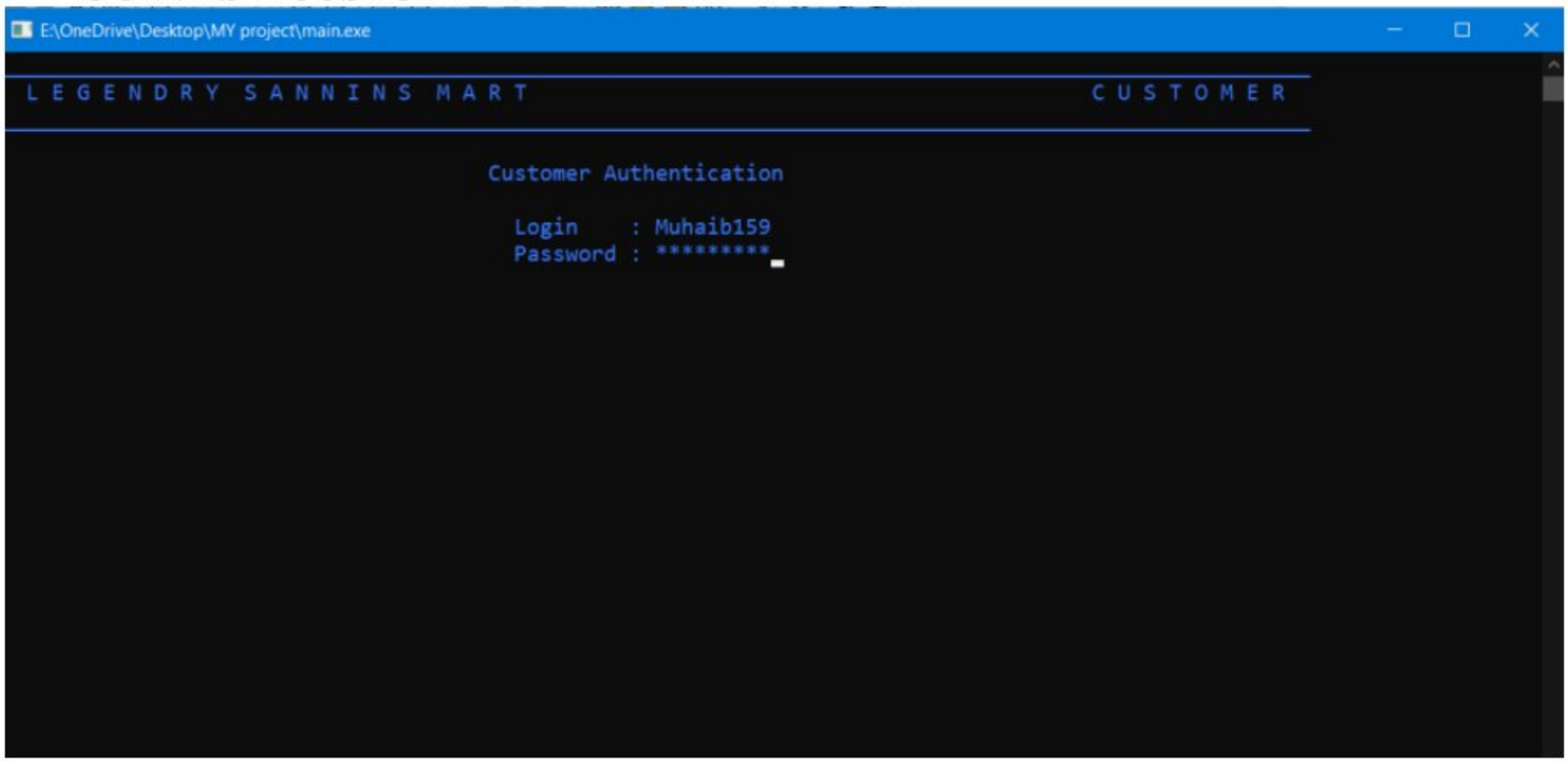
The page also presents the current stock of grocery items available for purchase.

OWNER'S PAGE TO MAKE CHANGES IN PRODUCT DETAILS:



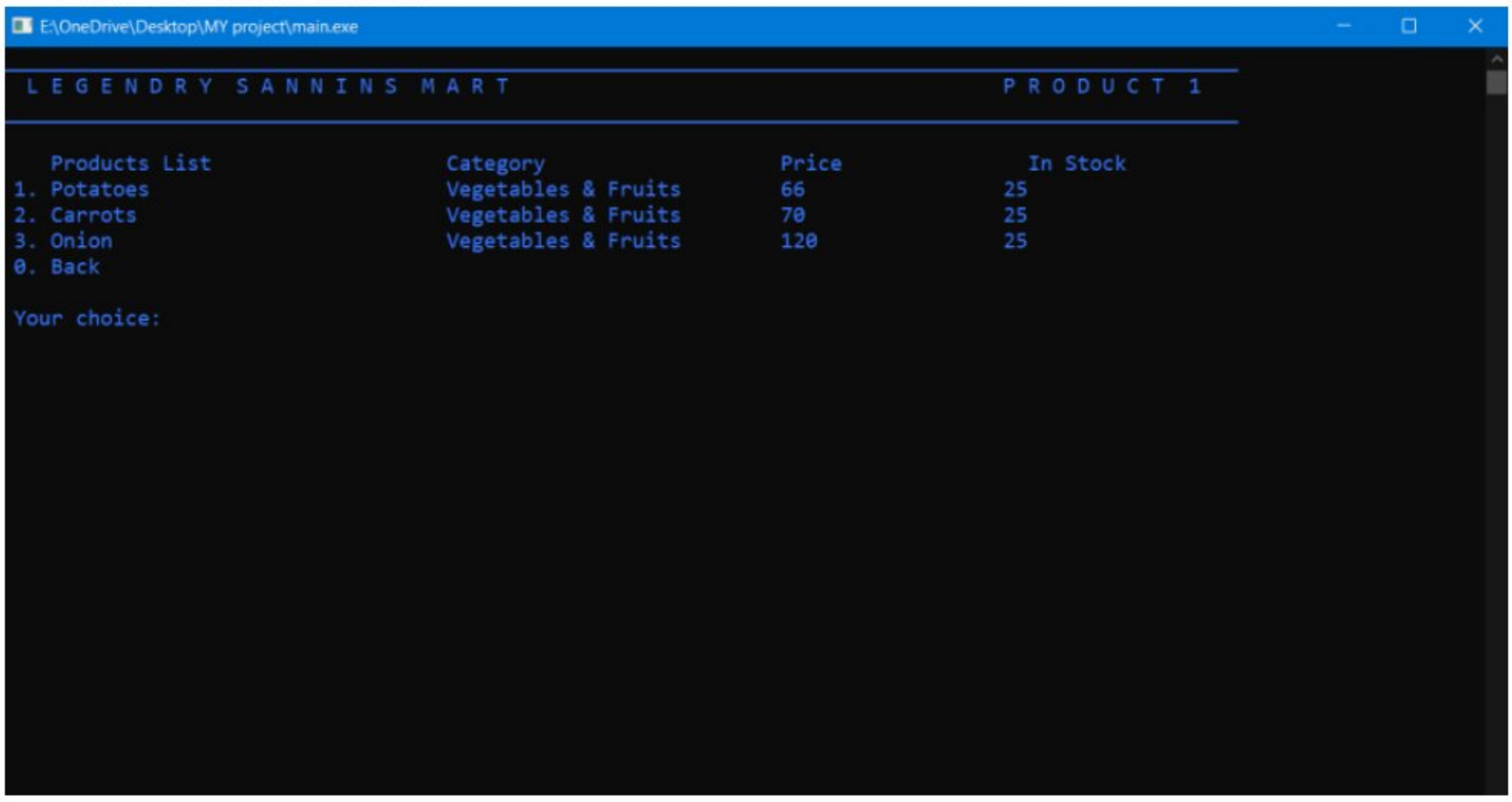
The owner has the authority to modify product information, such as changing the price, quantity, name, or category of grocery items.

LOGIN AS A CUSTOMER:



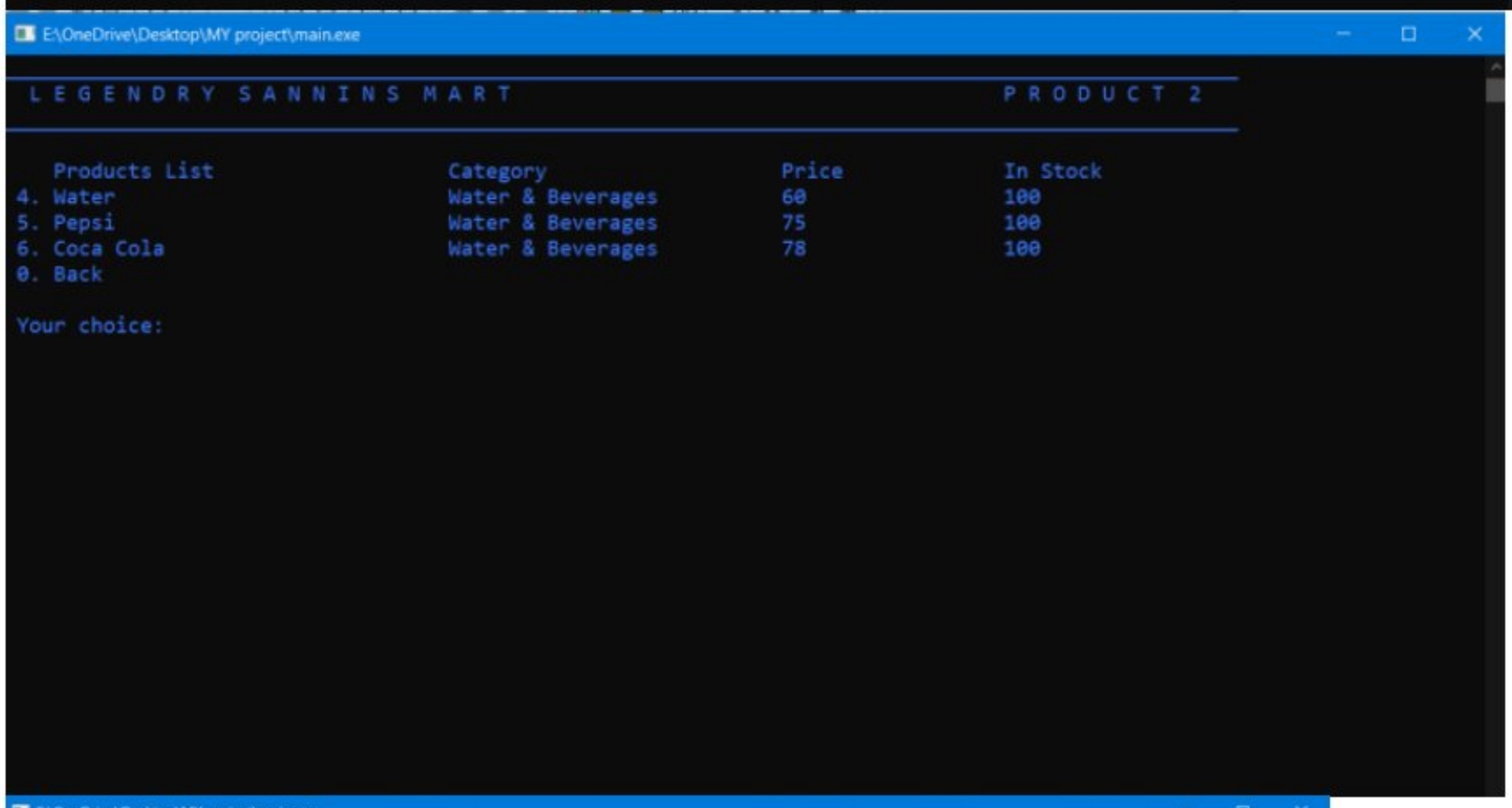
Registered customers can log in using their username and password. The login page includes appropriate validation and error handling to ensure a secure login process.

CATEGORIES PAGE:



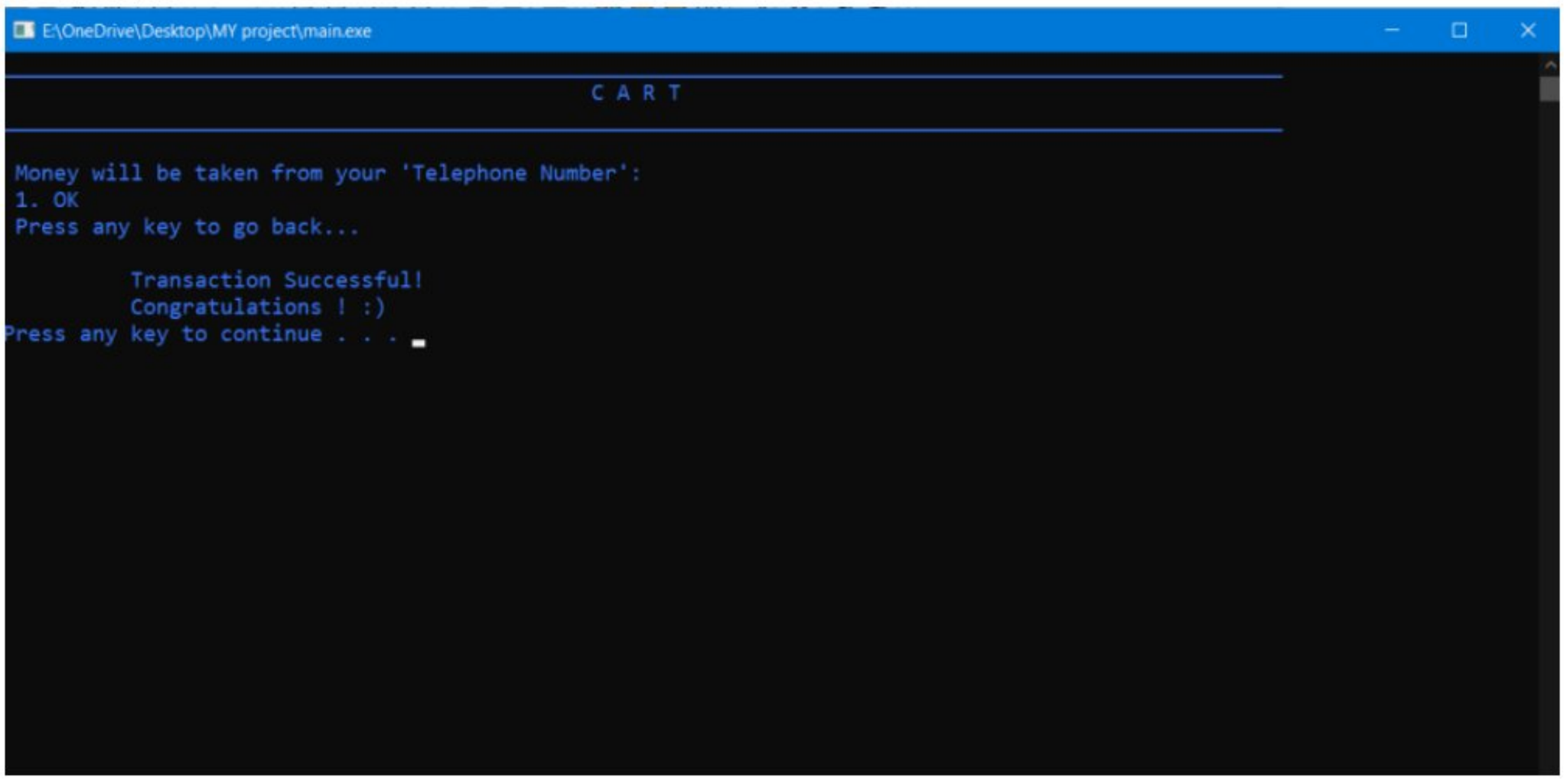
This page allows customer to choose their categories for shopping.

MENU PAGE:



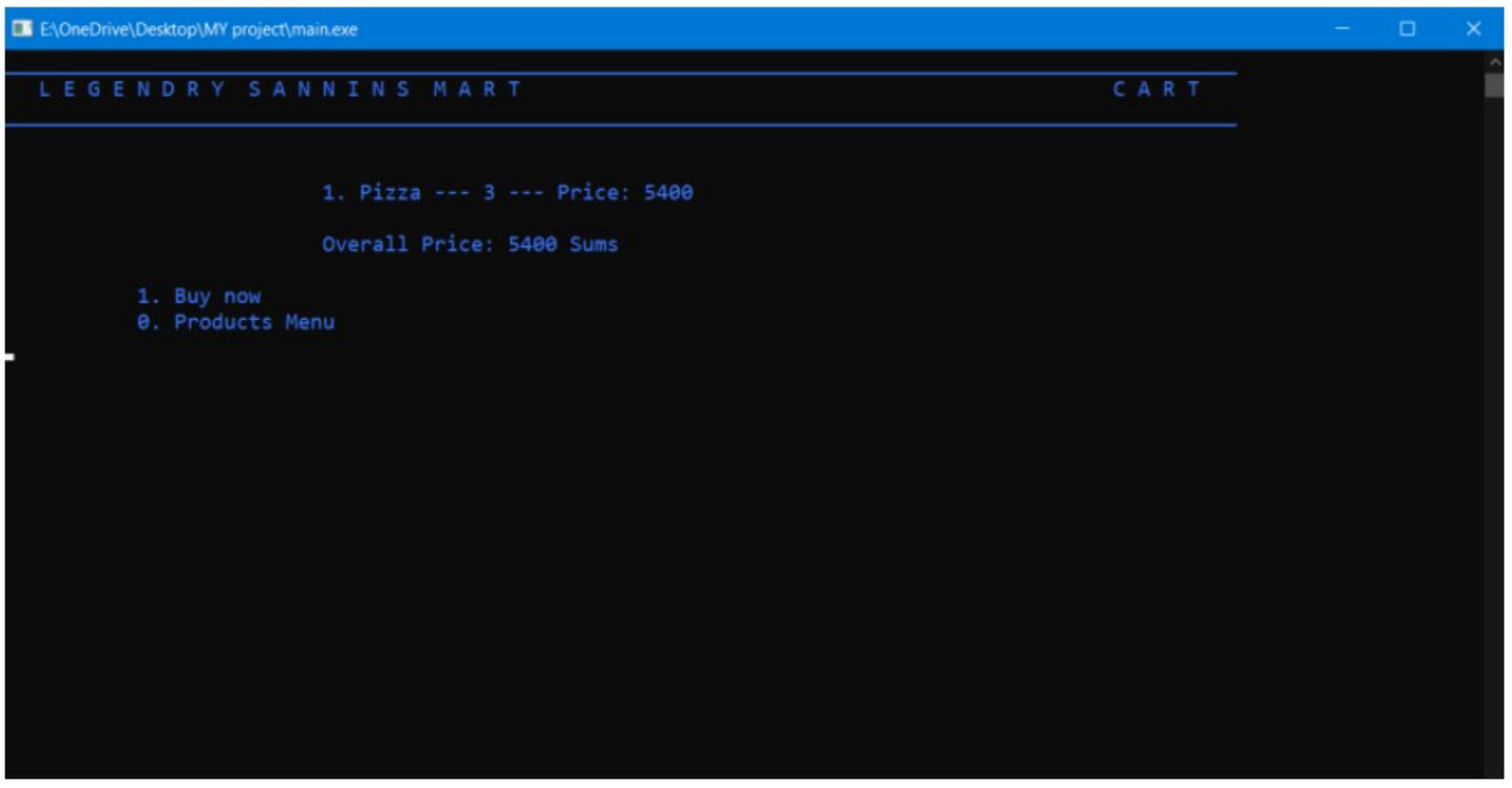
Upon successful login, customers are presented with a menu page displaying various grocery item categories such as fast food, vegetables, water & beverages. Customers can browse through different categories and select items to view their details and add them to the shopping cart.

CART:



A shopping cart feature that allows customers to add items, update quantities, and remove products before proceeding to checkout. The cart will store selected items and calculate the total amount to be paid.

TRANSACTION PAGE:



A secure transaction page that supports telephone number payment method, and ask customer that he is ok with the method or not, customer can go back to the cart .If the customer allow the transaction method then the screen will prompt "transaction successful" message.

CODE SCREENSHOT:

HEADER FILES:

LOADING PAGE:

```cpp
#include<iostream>
#include <ctime>      // Time
#include <windows.h> // Loading
#pragma once

using namespace std;

void gotoXY(int m, int n) {
    //'COORD' is a built in function for positioning the objects
    COORD d;
    d.X = m;
    d.Y = n;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), d);
}

void Loading()
{
    cout << "\n\n\n\n";
    cout << "\t\t\t    ---------------------------------------- \n";
    cout << "\t\t\t          O O P  P R O J E C T  2 0 2 3      \n";
    cout << "\t\t\t    ----------------------------------------  \n";
    cout << "\t\t\t     L E G E N D R Y  S A N N I N S  M A R T  \n";
    cout << "\t\t\t    _____________________________________\n";

    char a = 219;
    gotoXY(45, 14);

    cout << "LOADING... " << endl;

    gotoXY(37, 16);
    for (int r = 1; r <= 26; r++)
    {
        //for speed
        for (int speed = 0; speed <= 99000000; speed++);
            cout << a;
    }
    cout << endl;
}
```

PASSWORD:

```
#pragma once
#include <iostream>
#include <string>
#include <bits/stdc++.h>
#include <conio.h>

using namespace std;

// Global variables


string Function_Password_Val()
{
    string Password_Val = "";

    cout << "\t\t\t\t\tPassword : ";
    char ch;
    while ((ch = getch()) != 13)
    {
        if(ch!=8){
            Password_Val += ch;
            cout<<"*";
        }
        else if(ch == 8 ){
            putchar('\b');
            putchar(' ');
            putchar('\b');
            Password_Val = Password_Val.substr(0,Password_Val.length()-1);
        }
    }

    return Password_Val;
}
```

PERSON:

```cpp
#pragma once
#include <iostream>
using namespace std;

class Person {
protected:
    string name;
    string tell;

public:
    Person(string name, string tell) {
        this->name = name;
        this->tell = tell;
    }

    virtual void Display() = 0;
};

class Security : public Person {
protected:
    string login;
    string password;

public:
    Security(string name, string tell, string login, string password) : Person(name, tell) {
        this->login = login;
        this->password = password;
    }
};

class User : public Security {
public:
    // Product brought by the
    int Product1_1_User = 0, Product1_2_User = 0, Product1_3_User = 0;
    int Product2_1_User = 0, Product2_2_User = 0, Product2_3_User = 0;
    int Product3_1_User = 0, Product3_2_User = 0, Product3_3_User = 0;

    User(string name, string tell, string login, string password) : Security(name, tell, login, password) {}

    string getName() {
        return name;
    }

    string getTell() {
        return tell;
    }

    string getLogin() {
        return login;
    }

    string getPassword() {
        return password;
    }

    void Display() {
        cout << "\t\tName     : " << name << endl;
        cout << "\t\tTell     : " << tell << endl;
        cout << "\t\tLogin    : " << login << endl;
        cout << "\t\tPassword : " << password << endl;
    }
};
```

MENU:

```
#include<iostream>
#include<conio.h>
using namespace std;

#include "LoadingPage.h"

void welcome()
{
    cout<<endl<<endl;
    cout<<"\t\t\t\t**************************************************\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*   ---------------------------------------      *\n" ;
    cout<<"\t\t\t\t*    L E G E N D R Y  S A N N I N S  M A R T     *\n" ;
    cout<<"\t\t\t\t*   ---------------------------------------      *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*            Project By:                         *\n" ;
    cout<<"\t\t\t\t*            MUHAIB SHAMSHER (CR-029)            *\n" ;
    cout<<"\t\t\t\t*            MUNEEZA BADAR   (CR-022)            *\n" ;
    cout<<"\t\t\t\t*            MUHAMMAD HAMZA  (CR-048)            *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*                  OOP PROJECT 2023              *\n" ;
    cout<<"\t\t\t\t*                  Course Instructor:            *\n" ;
    cout<<"\t\t\t\t*                  SIR SHARIQ MAHMOOD KHAN       *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t*                                                *\n" ;
    cout<<"\t\t\t\t**************************************************\n\n\n" ;
}


void clear()
{
    cout << "Press any key to continue!";
    getch();
    system("CLS");
}


void start() {
    Loading();
    system("cls");
    welcome();
    clear();
}
```

PRODUCT:

```cpp
#pragma once
#include <iostream>
#include <string>
#include<iomanip>

using namespace std;

int position1 = 1; // for numbering the products

class Products {
    public:
        string name;
        string class_of_products;
        double price;

    public:
        Products(string name, string class_of_products, double price) {
            this->name = name;
            this->class_of_products = class_of_products;
            this->price = price;
        }

        virtual void Display() = 0;
};


class Class_Of_Products1 : public Products {
    public:
        int quantity;

    public:
        Class_Of_Products1(string name, string class_of_products, double price, int quantity):
        Products(name, class_of_products, price)
        {
            this->quantity = quantity;
        }

        string getName() {
            return name;
        }

        string getClassOfProducts() {
            return class_of_products;
        }

        double getPrice() {
            return price;
        }

        int getQuantity() {
            return quantity;
        }

        void Display() {
            cout << " " << position1 << ". " << left << setw(32)
                        << name              << left << setw(27)
                        << class_of_products << left << setw(18)
                        << price << quantity << endl;

            position1++;
        }
};

```

```
65
66  class Class_Of_Products2 : public Products {
67      public:
68          int quantity;
69
70      public:
71          Class_Of_Products2(string name, string class_of_products, double price, int quantity) : Products(name, class_of_products, price) {
72              this->quantity = quantity;
73          }
74
75          string getName() {
76              return name;
77          }
78
79          string getClassOfProducts() {
80              return class_of_products;
81          }
82
83          double getPrice() {
84              return price;
85          }
86
87          int getQuantity() {
88              return quantity;
89          }
90
91          void Display() {
92              cout << " " << position1 << ". " << left << setw(32) << name << left << setw(27) << class_of_products << left << setw(18) << price << quantity << endl;
93              position1++;
94          }
95  };
96
97
98  class Class_Of_Products3 : public Products {
99      public:
100         int quantity;
101
102     public:
103         Class_Of_Products3(string name, string class_of_products, double price, int quantity) : Products(name, class_of_products, price) {
104             this->quantity = quantity;
105         }
106
107         string getName() {
108             return name;
109         }
110
111         string getClassOfProducts() {
112             return class_of_products;
113         }
114
115         double getPrice() {
116             return price;
117         }
118
119         int getQuantity() {
120             return quantity;
121         }
122
123         void Display() {
124             cout << " " << position1 << ". " << left << setw(32) << name << left << setw(27) << class_of_products << left << setw(18) << price << quantity << endl;
125             position1++;
126         }
127 };
```

VALIDATION:

```cpp
#include <iostream>
#include <string>
#include <ctype.h>
using namespace std;

// Global Identifires for validation
int Num_Upper = 0, Num_Lower = 0, Num_Number = 0, validation = 0;

// Identifires for File Hendling
string Name_Memory, Login_Memory, Password_Memory, TellNum_Memory;


class Validation {
private:
    string Name, Login, Password, TellNum;

public:
    Validation(): Name(" "), Login(" "), Password(" "), TellNum(" ") {}

    void Display() {
        cout << "\t\t\tUser Information :" << endl << endl;
        cout << "\t\tUser Name  : " << Name << endl;
        cout << "\t\tTelephone  : " << TellNum << endl;
        cout << "\t\tLogin      : " << Login << endl;
        cout << "\t\tPassword   : " << Password << endl;
    }

    void SetUser() {
        cout << "\t\t\t\t       Register User " << endl << endl;
        cout << "\t\t\t\t        User Name : " ;    cin >> Name;
        cout << "\t\t\t\t        Telephone : " ;    cin >> TellNum;
        cout << "\t\t\t\t        Login     : " ;    cin >> Login;
        cout << "\t\t\t\t        Password  : " ;    cin >> Password;

        // Info sended to Memory
        Name_Memory = Name;
        Password_Memory = Password;
        Login_Memory = Login;
        TellNum_Memory = TellNum;
    }

    //Validation Check for Password.
    friend void ValidationPassword(Validation User);

    //Validation Check for Name
    friend void ValidationName(Validation  User);

    //Validation Check for TellNum
    friend void ValidationTellNum(Validation User);
};
```

```
51
52
53  //Validation Check for Password.
54  void ValidationPassword(Validation User)
55  {
56      if (User.Password.length() >= 6 && User.Password.length() <= 15)
57      {
58          for (int i = 0; i < User.Password.length(); i++) {
59              if (isupper(User.Password[i])) {
60                  Num_Upper += 1;
61              }
62              if (islower(User.Password[i])) {
63                  Num_Lower += 1;
64              }
65              if (isdigit(User.Password[i])) {
66                  Num_Number += 1;
67              }
68          }
69
70          if (Num_Upper >= 1 && Num_Upper <= 10 && Num_Lower >= 4 && Num_Lower <= 10 && Num_Number >= 2 && Num_Number <= 10) {
71              validation++;
72              Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
73          }
74
75          else {
76              Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
77          }
78      }
79
80      else {
81          Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
82      }
83  }
84
85  //Validation Check for Name
86  void ValidationName(Validation  User)
87  {
88      if (User.Name.length() >= 1 && User.Name.length() <= 15)
89      {
90          for (int i = 0; i < User.Name.length(); i++) {
91              if (isupper(User.Name[i])) {
92                  Num_Upper += 1;
93              }
94              if (islower(User.Name[i])) {
95                  Num_Lower += 1;
96              }
97              if (isdigit(User.Name[i])) {
98                  Num_Number += 1;
99              }
100         }
```

```
101
102         if (Num_Upper <= 1 && Num_Lower >= 1 && Num_Lower <= 14 && Num_Number == 0) {
103             validation++;
104             Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
105         }
106
107         else {
108             Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
109         }
110     }
111
112     else {
113         Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
114     }
115 }
116
117 //Validation Check for TellNum
118 void ValidationTellNum(Validation User)
119 {
120     if (User.TellNum.length() >= 9 && User.TellNum.length() <= 12) {
121         for (int i = 0; i < User.TellNum.length(); i++) {
122             if (isupper(User.TellNum[i])) {
123                 Num_Upper += 1;
124             }
125             if (islower(User.TellNum[i])) {
126                 Num_Lower += 1;
127             }
128             if (isdigit(User.TellNum[i])) {
129                 Num_Number += 1;
130             }
131         }
132
133         if (Num_Upper == 0 && Num_Lower == 0 && Num_Number >= 9 && Num_Number <= 12) {
134             validation++;
135             Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
136         }
137     }
138
139     else {
140         Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
141     }
142 }
```

MAIN CODE:

```cpp
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

// Header Files
#include "menu.h"
#include "Validation.h"
#include "Sign_Password.h"
#include "Products.h"
#include "person.h"

// Classes
Validation User_Validtaion;
User user(Name_Memory, TellNum_Memory, Login_Memory, Password_Memory);

Class_Of_Products1 Product1_1("Potatoes", "Vegetables & Fruits", 65.0, 25);
Class_Of_Products1 Product1_2("Carrots", "Vegetables & Fruits", 70.0, 25);
Class_Of_Products1 Product1_3("Onion", "Vegetables & Fruits", 120.0, 25);

Class_Of_Products2 Product2_1("Water", "Water & Beverages", 60.0, 100);
Class_Of_Products2 Product2_2("Pepsi", "Water & Beverages", 75.0, 100);
Class_Of_Products2 Product2_3("Coca Cola", "Water & Beverages", 78.0, 100);

Class_Of_Products3 Product3_1("Pizza", "Fast Food Products", 1800.0, 20);
Class_Of_Products3 Product3_2("Burger", "Fast Food Products", 900.0, 20);
Class_Of_Products3 Product3_3("Potatoe Fries", "Fast Food Products", 230.0, 20);


class LegendrySanninsMart {
    private:
        // Password and login for Owner
        string Owner_Login_Sign = "1";
        string Owner_Password_Sign = "1";

    protected:
        // Functions
        void Sign_in();
        void About();
        void Owner_Main_Menu();
        void Owner_Products_Stotage();
        void Modify_Info_Products(int take_product_location);
        void Owner_Customers_List();
        void User_Main_Menu();
        void Table_For_Increasing_And_Decreasing();
        void Class_Of_Products1_Menu();
        void Class_Of_Products2_Menu();
        void Class_Of_Products3_Menu();
        void Cart_Check();

    public:
        void General_Menu();
};

```

```
65
66  void LegendrySanninsMart :: General_Menu()
67  {
68      for (int i = 0; i < 1000; i++)
69      {
70          system("cls");
71          cout << "_____________________________________________________________________________ \n";
72          cout << "  L E G E N D R Y  S A N N I N S  M A R T                  A U T H E N T I C A T I O N   \n";
73          cout << "_____________________________________________________________________________ \n\n";
74          cout << "\t\t\t\t\t  Authentication \n"
75               << endl;
76          cout << "\t\t\t\t\t   1. Sign in" << endl;
77          cout << "\t\t\t\t\t   2. Sign up" << endl;
78          cout << "\t\t\t\t\t   3. About" << endl;
79          cout << "\t\t\t\t\t   0. Exit" << endl
80               << endl;
81          cout << "\t\t\t\t\t  Your Choice: ";
82
83          switch (getch())
84          {
85          case 49:
86              // Sign in
87              system("cls");
88              Sign_in();
89              system("pause");
90              break;
91
92          case 50:
93              // Register
94              // Set details with validation
95              for (int i = 0; i != 1;)
96              {
97                  system("cls");
98                  cout << "_____________________________________________________________________________ \n";
99                  cout << "  L E G E N D R Y  S A N N I N S  M A R T                  R E G I S T R A T I O N   \n";
100                 cout << "_____________________________________________________________________________\n\n";
101                 cout << "\t\t\t\t\t Example of Registration: " << endl;
102                 cout << "\t\t\t\t\t _____________________" << endl;
103                 cout << "\t\t\t\t\t  User Name : MuhaibShamsher " << endl;
104                 cout << "\t\t\t\t\t  Telephone : 03448625508 " << endl;
105                 cout << "\t\t\t\t\t  Login     : Muhaib123 " << endl;
106                 cout << "\t\t\t\t\t  Password  : Muhaib123 " << endl;
107                 cout << "\t\t\t\t\t _____________________" << endl
108                      << endl;
109
110                 User_Validtaion.SetUser();
111                 ValidationPassword(User_Validtaion);
112                 ValidationName(User_Validtaion);
113                 ValidationTellNum(User_Validtaion);
```

```
114
115             if (validation == 3)
116             {
117                 // File Handling For User Info
118                 ofstream User_Info;
119                 User_Info.open("User_Info.txt", ios::app); // Opening File
120
121                 // Setting User Info in Memory
122                 User_Info << Name_Memory << endl;
123                 User_Info << TellNum_Memory << endl;
124                 User_Info << Login_Memory << endl;
125                 User_Info << Password_Memory << endl;
126                 User_Info.close();
127                 // End of the File Handling
128
129                 cout << "\t\t\t\t ____________________________________" << endl;
130                 cout << "\t\t\t\t      Correct Validation" << endl;
131                 i++;
132                 cout << "\t\t\t\tPlease press any key to continue..." << endl
133                     << endl;
134                 system("Pause");
135                 validation = 0;
136             }
137
138             else
139             {
140                 cout << "\t\t\t\t ____________________________________" << endl;
141                 cout << "\t\t\t\t       Invalid Validation" << endl;
142                 cout << "\t\t  Please press '1' to rewrite || Press any key to go back to menu" << endl
143                     << endl;
144                 validation = 0;
145
146                 switch (getch())
147                 {
148                 case 49:
149                     // Rewriting i=0;
150                     break;
151
152                 default:
153                     General_Menu();
154                 }
155             }
156             }
157             break;
158
159         case 51:
160             // About
161             system("cls");
162             About();
163             break;
164
165         case 48:
166             // Exist
167             system("cls");
168             cout << "\n\n\t\t\t Thank you for your attention! \n\n\n\n";
169             system("pause");
170             exit(0);
171             break;
172
173         default:
174             cout << "\n\n\t\t\t Your choice is not available in Menu." << endl;
175             cout << "\t\t\t Please enter correct keys." << endl;
176             system("pause");
177             break;
178         }
179     }
180 }
```

```cpp
void LegendrySanninsMart :: About()
{
    cout << endl
         << endl;
    cout << "\t\t\t\t**************************************************\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*   ------------------------------------         *\n";
    cout << "\t\t\t\t*    L E G E N D R Y  S A N N I N S  M A R T     *\n";
    cout << "\t\t\t\t*   ------------------------------------         *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*            Project By:                         *\n";
    cout << "\t\t\t\t*            MUHAIB SHAMSHER (CR-029)            *\n";
    cout << "\t\t\t\t*            MUNEEZA BADAR   (CR-022)            *\n";
    cout << "\t\t\t\t*            MUHAMMAD HAMZA  (CR-048)            *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*                    OOP PROJECT 2023            *\n";
    cout << "\t\t\t\t*                    Course Instructor:          *\n";
    cout << "\t\t\t\t*                    SIR SHARIQ MAHMOOD KHAN     *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t*                                                *\n";
    cout << "\t\t\t\t**************************************************\n\n\n";

    cout << "\n\t\t\t\tPress any key to go back to Menu \n\n";
    system("pause");
}
// DONE

void LegendrySanninsMart :: Sign_in()
{
    for (int i = 0; i < 1000; i++)
    {
        string Login_Sign, Password_Sign;

        system("cls");
        cout << "_____________________________________________________________________________________________ \n";
        cout << "  L E G E N D R Y  S A N N I N S  M A R T                                  S I G N  I N   \n";
        cout << "_____________________________________________________________________________________________\n\n";
        cout << "\t\t\t\t\t   Sign in\n\n";
        cout << "\t\t\t\t\t1. Sign in as Owner" << endl;
        cout << "\t\t\t\t\t2. Sign in as User" << endl;
        cout << "\t\t\t\t\t0. Back" << endl
             << endl;
        cout << "\t\t\t\t\t   Your Choice: ";

        switch (getch())
        {
        case 49:
            // Sign in as Owner
            system("cls");
            cout << "_____________________________________________________________________________________________ \n";
            cout << "  L E G E N D R Y  S A N N I N S  M A R T                                  O W N E R          \n";
            cout << "_____________________________________________________________________________________________\n\n";

            cout << "\t\t\t\t\t Owner Authentication\n\n";
            cout << "\t\t\t\t\t Login    : ";
            cin >> Login_Sign;

            Password_Sign = Function_Password_Val(); // the funcation will take Password as input

            if (Login_Sign == Owner_Login_Sign && Password_Sign == Owner_Password_Sign)
            {
                // Owner's Menu
                Owner_Main_Menu();
                system("pause");
            }
```

```
253                 // If Login hasn't Registered
254                 cout << endl
255                      << endl;
256                 cout << "\t\t\t   Your Login and Password are Invalid." << endl;
257                 cout << "\t\t\t   Please press any key to go back to 'Sign in' Menu.\n";
258                 cout << endl
259                      << endl;
260                 system("pause");
261                 Sign_in();
262             }
263             break;
264
265         case 50:
266         {
267             // Sign in as User
268             system("cls");
269             cout << "_____________________________________________________________________________\n";
270             cout << "  L E G E N D R Y  S A N N I N S  M A R T                    C U S T O M E R  \n";
271             cout << "_____________________________________________________________________________\n\n";
272             cout << "\t\t\t\t      Customer Authentication\n\n";
273             cout << "\t\t\t\t\tLogin    : ";
274             cin >> Login_Sign;
275
276             Password_Sign = Function_Password_Val();
277
278             // File Handling For User Info
279             ifstream Search;
280             Search.open("User_Info.txt");
281
282             while (Search)
283             {
284                 Search >> Name_Memory;
285                 Search >> TellNum_Memory;
286                 Search >> Login_Memory;
287                 Search >> Password_Memory;
288                 if (Login_Sign == Login_Memory && Password_Sign == Password_Memory)
289                 {
290                     User user(Name_Memory, TellNum_Memory, Login_Memory, Password_Memory);
291                     User_Main_Menu();
292                 }
293             }
294             Search.close();
295             // End of the File Handling
296
297             cout << "\n\n\t\t\t        Your Login and Password are Invalid." << endl;
298             cout << "\t\t\t   Please press any key to go back to 'Sign in' Menu.\n\n"
299                  << endl;
300             system("pause");
301             Sign_in();
302         }
303         break;
304
305         case 48:
306             // Back
307             system("cls");
308             i = 1000;
309             General_Menu();
310             break;
311
312         default:
313             cout << "\n\n\t\t\t    Your choice is not available in Menu." << endl;
314             cout << "\t\t\t\t    Please enter correct keys.\n"
315                  << endl;
316             system("pause");
317             break;
318         }
319     }
320 }
```

```
322
323 void LegendrySanninsMart :: Owner_Main_Menu()
324 {
325     system("cls");
326     cout << endl
327          << endl;
328
329     for (int k = 0; k < 1000; k++)
330     {
331         system("cls");
332         cout << "_____________________________________________________________________________\n";
333         cout << " L E G E N D R Y  S A N N I N S  M A R T                    O W N E R  M E N U  \n";
334         cout << "_____________________________________________________________________________\n\n";
335
336         cout << "       Main Menu\n\n";
337         cout << "       1. Products in stock \n\n";
338         cout << "       2. Customers list \n\n";
339         cout << "       0. Back\n\n";
340         cout << "       Your choice: ";
341
342         switch (getch())
343         {
344         case 49:
345             Owner_Products_Stotage();
346             break;
347
348         case 50:
349             Owner_Customers_List();
350             system("pause");
351             break;
352
353         case 48:
354             // Back to Menu
355             system("cls");
356             k = 1000;
357             Sign_in();
358             break;
359
360         default:
361             cout << "\t\t\t\t    Your choice is not available in Menu." << endl;
362             cout << "\t\t\t\t    Please enter correct keys.\n"
363                  << endl;
364             system("pause");
365             break;
366         }
367     }
368 }
369 // DONE
```

```
void LegendrySanninsMart :: Owner_Products_Stotage()
{
    for (int i = 0; i < 1000; i++)
    {
        system("cls");
        cout << "_____________________________________________________________________________________\n";
        cout << "  L E G E N D R Y  S A N N I N S  M A R T                  P R O D U C T S  I N  S T O C K  \n";
        cout << "_____________________________________________________________________________________\n\n";

        cout << "    Products List  \t\t    Category\t\t      Price\t\t    In Stock\n";
        Product1_1.Display();
        Product1_2.Display();
        Product1_3.Display();
        Product2_1.Display();
        Product2_2.Display();
        Product2_3.Display();
        Product3_1.Display();
        Product3_2.Display();
        Product3_3.Display();

        position1 = 1; // numbering of products return to 1

        cout << endl
             << endl;
        cout << "   0. Back\n";
        cout << "   Make changes in: ";
        switch (getch())
        {
        case '1':
            Modify_Info_Products(1);
            break;

        case '2':
            Modify_Info_Products(2);
            break;

        case '3':
            Modify_Info_Products(3);
            break;

        case '4':
            Modify_Info_Products(4);
            break;

        case '5':
            Modify_Info_Products(5);
            break;

        case '6':
            Modify_Info_Products(6);
            break;
```

```
422
423         case '7':
424             Modify_Info_Products(7);
425             break;
426
427         case '8':
428             Modify_Info_Products(8);
429             break;
430
431         case '9':
432             Modify_Info_Products(9);
433             break;
434
435         case '0':
436             // Back to Menu
437             system("cls");
438             i = 1000;
439             Owner_Main_Menu();
440             break;
441
442         default:
443             cout << "\n\t\t      Your choice is not available in Menu" << endl;
444             cout << "\t\tPlease press any keyboard to continue program\n"
445                  << endl;
446             system("pause");
447             break;
448         }
449     }
450 }
451 // DONE
452
453 void LegendrySanninsMart :: Modify_Info_Products(int take_product_location)
454 {
455     for (int j = 0; j < 1000; j++)
456     {
457         system("cls");
458
459         cout << "_____________________________________________________________________________________ \n";
460         cout << "  L E G E N D R Y   S A N N I N S   M A R T                                          \n";
461         cout << "_____________________________________________________________________________________ \n\n";
462
463         cout << "    Products List  \t\t    Category\t\t      Price\t\t In Stock\n";
464
465         if (take_product_location == 1)
466             Product1_1.Display();
467
468         else if (take_product_location == 2)
469             Product1_2.Display();
470
471         else if (take_product_location == 3)
472             Product1_3.Display();
473
474         else if (take_product_location == 4)
475             Product2_1.Display();
476
477         else if (take_product_location == 5)
478             Product2_2.Display();
479
480         else if (take_product_location == 6)
481             Product2_3.Display();
482
483         else if (take_product_location == 7)
484             Product3_1.Display();
485
486         else if (take_product_location == 8)
487             Product3_2.Display();
```

```
        else if (take_product_location == 9)
            Product3_3.Display();

        position1 = 1;
        cout << "------------------------------------------------------------------------------------------------------ \n\n";
        cout << " 1. Change the price \n";
        cout << " 2. Change the quantity in storage\n";
        cout << " 3. Change the name of product\n";
        cout << " 4. Change the class of product\n";
        cout << " 0. Go back \n\n";
        cout << "    Your choice : \n";

        switch (getch())
        {
        // FOR CHANGING A PRICE
        case 49:
            cout << "   Enter a new price: ";
            double change_price;
            cin >> change_price;

            if (change_price >= 0)
            {
                if (take_product_location == 1)
                    Product1_1.price = change_price;

                else if (take_product_location == 2)
                    Product1_2.price = change_price;

                else if (take_product_location == 3)
                    Product1_3.price = change_price;

                else if (take_product_location == 4)
                    Product2_1.price = change_price;

                else if (take_product_location == 5)
                    Product2_2.price = change_price;

                else if (take_product_location == 6)
                    Product2_3.price = change_price;

                else if (take_product_location == 7)
                    Product3_1.price = change_price;

                else if (take_product_location == 8)
                    Product3_2.price = change_price;

                else if (take_product_location == 9)
                    Product3_3.price = change_price;

                cout << "    Successfully changed!\n";
                Sleep(0700);
                Sleep(0700);
            }

            else
            {
                cout << " Price cannot be negative! Please check one more time.\n";
                Sleep(0700);
                Sleep(0700);
            }
            break;
```

```
550
551         // FOR CHANGING THE QUANTITY IN STORAGE
552         case 50:
553             cout << "    Enter a new quantity in storage: ";
554             int change_Quantity;
555             cin >> change_Quantity;
556
557             if (change_Quantity > 0)
558             {
559                 if (take_product_location == 1)
560                     Product1_1.quantity = change_Quantity;
561
562                 else if (take_product_location == 2)
563                     Product1_2.quantity = change_Quantity;
564
565                 else if (take_product_location == 3)
566                     Product1_3.quantity = change_Quantity;
567
568                 else if (take_product_location == 4)
569                     Product2_1.quantity = change_Quantity;
570
571                 else if (take_product_location == 5)
572                     Product2_2.quantity = change_Quantity;
573
574                 else if (take_product_location == 6)
575                     Product2_3.quantity = change_Quantity;
576
577                 else if (take_product_location == 7)
578                     Product3_1.quantity = change_Quantity;
579
580                 else if (take_product_location == 8)
581                     Product3_2.quantity = change_Quantity;
582
583                 else if (take_product_location == 9)
584                     Product3_3.quantity = change_Quantity;
585
586                 cout << "    Successfully changed!\n";
587                 Sleep(0700);
588                 Sleep(0700);
589             }
590
591             else
592             {
593                 cout << " Quantity cannot be negative\n";
594                 Sleep(0700);
595                 Sleep(0700);
596             }
597             break;
598
599         // FOR CHANGING THE NAME
600         case 51:
601         {
602             cout << "    Enter a new name: ";
603             string product_name = " ";
604             getline(cin, product_name);
605
606             if (take_product_location == 1)
607                 Product1_1.name = product_name;
608
609             else if (take_product_location == 2)
610                 Product1_2.name = product_name;
611
612             else if (take_product_location == 3)
613                 Product1_3.name = product_name;
614
615             else if (take_product_location == 4)
616                 Product2_1.name = product_name;
617
```

```
617
618             else if (take_product_location == 5)
619                 Product2_2.name = product_name;
620
621             else if (take_product_location == 6)
622                 Product2_3.name = product_name;
623
624             else if (take_product_location == 7)
625                 Product3_1.name = product_name;
626
627             else if (take_product_location == 8)
628                 Product3_2.name = product_name;
629
630             else if (take_product_location == 9)
631                 Product3_3.name = product_name;
632
633             cout << "    Successfully changed!\n";
634             Sleep(0700);
635             Sleep(0700);
636         }
637         break;
638
639         case 52:
640         {
641             cout << "    Enter a new Class Name: ";
642             string class_name = " ";
643             getline(cin, class_name);
644
645             if (take_product_location == 1)
646                 Product1_1.class_of_products = class_name;
647
648             else if (take_product_location == 2)
649                 Product1_2.class_of_products = class_name;
650
651             else if (take_product_location == 3)
652                 Product1_3.class_of_products = class_name;
653
654             else if (take_product_location == 4)
655                 Product2_1.class_of_products = class_name;
656
657             else if (take_product_location == 5)
658                 Product2_2.class_of_products = class_name;
659
660             else if (take_product_location == 6)
661                 Product2_3.class_of_products = class_name;
662
663             else if (take_product_location == 7)
664                 Product3_1.class_of_products = class_name;
665
666             else if (take_product_location == 8)
667                 Product3_2.class_of_products = class_name;
668
669             else if (take_product_location == 9)
670                 Product3_3.class_of_products = class_name;
671
672             cout << "    Successfully changed!\n";
673             Sleep(0700);
674             Sleep(0700);
675         }
676         break;
677
678         case 48:
679             j = 1000;
680             break;
681         }
682     }
683 }
```

```
685
686  void LegendrySanninsMart :: Owner_Customers_List()
687  {
688      system("cls");
689      cout << "_____________________________________________________________________________________\n";
690      cout << "  L E G E N D R Y  S A N N I N S  M A R T                    C U S T O M E R S  L I S T   \n";
691      cout << "_____________________________________________________________________________________\n\n";
692
693      cout << "\n\t\t\t   Customers list : " << endl;
694      cout << endl
695           << endl;
696
697      ifstream in;
698      int Num = 1;
699      string Info;
700
701      in.open("User_Info.txt");
702      while (in)
703      {
704          cout << "\t " << Num << "." << endl;
705          cout << "\t------------------------" << endl;
706          getline(in, Info);
707          cout << "\t User Name : " << Info << endl;
708          getline(in, Info);
709          cout << "\t Phone     : " << Info << endl;
710          getline(in, Info);
711          cout << "\t Login     : " << Info << endl;
712          getline(in, Info);
713          cout << "\t Password  : " << Info << endl;
714          Num++;
715          cout << endl;
716      }
717      cout << "\t------------------------" << endl;
718      in.close();
719  }
720  // DONE
```

```cpp
void LegendrySanninsMart :: User_Main_Menu()
{
    for (int i = 0; i < 1000; i++)
    {
        system("cls");
        cout << "_____________________________________________________________________________________ \n";
        cout << "  L E G E N D R Y  S A N N I N S  M A R T                    U S E R  M A I N  M E N U   \n";
        cout << "_____________________________________________________________________________________\n\n";

        cout << "\t\t Categories\n\n";
        cout << "\t\t 1. Vegetables & Fruits\n";
        cout << "\t\t 2. Water & Beverages\n";
        cout << "\t\t 3. Fast Food Products\n";
        cout << "\t\t 4. Cart and Overall Sums\n";
        cout << "\t\t 0. Go Back\n\n";
        cout << "\t\t Your choice: ";

        switch (_getch())
        {
        case 49:
            Class_Of_Products1_Menu();
            break;

        case 50:
            Class_Of_Products2_Menu();
            break;

        case 51:
            Class_Of_Products3_Menu();
            break;

        case 52:
            Cart_Check();
            break;

        case 48:
            // Back to Menu
            system("cls");
            i = 1000;
            Sign_in();
            break;

        default:
            cout << endl
                 << endl;
            cout << "\t\t\t\t    Your choice is not available in Menu." << endl;
            cout << "\t\t\t\t    Please enter correct keys.\n"
                 << endl;
            system("pause");
            break;
        }
    }
}
```

```
void LegendrySanninsMart :: Table_For_Increasing_And_Decreasing()
{
    cout << "_____________________________________________________________________________________\n";
    cout << "     (+)   'Press 1'                    \n";
    cout << "     (-)   'Press 2'                    \n";
    cout << "     (0)   'Back'                       \n";
    cout << "_____________________________________________________________________________________\n\n";
    cout << "     Add to Cart:  \n";
}
// DONE

void LegendrySanninsMart :: Class_Of_Products1_Menu()
{
    for (int k = 0; k < 1000; k++)
    {
        system("cls");
        cout << "_____________________________________________________________________________ \n";
        cout << "  L E G E N D R Y  S A N N I N S  M A R T                    P R O D U C T  1   \n";
        cout << "_____________________________________________________________________________\n\n";

        cout << "    Products List  \t\t    Category \t\t      Price \t\t   In Stock\n";
        Product1_1.Display();
        Product1_2.Display();
        Product1_3.Display();

        cout << " 0. Back\n\n";
        cout << " Your choice: ";

        switch (getch())
        {
            // for Product1_1
            case 49:
                for (int j = 0; j < 1000; j++) {
                    system("cls");
                    cout << "    Products List  \t\t    Category\t\t      Price\t\t In Stock\n";
                    Product1_1.Display();
                    position1 = 1;

                    Table_For_Increasing_And_Decreasing();

                    switch (getch())
                    {
                    case 49:
                        if (Product1_1.getQuantity() > 0) {
                            user.Product1_1_User++;
                            Product1_1.quantity--;
                            cout << " Quantity of " << Product1_1.getName() << " - " << user.Product1_1_User << endl;
                            cout << " Successfully added \n";
                            Sleep(0700);
                            Sleep(0700);
                        }

                        else {
                            cout << " Product is over / finished. Sorry!\n";
                            Sleep(0700);
                            Sleep(0700);
                        }
                        break;

                    case 50:
                        if (user.Product1_1_User > 0) {
                            user.Product1_1_User--;
                            Product1_1.quantity++;
                            cout << " Quantity of " << Product1_1.getName() << " - " << user.Product1_1_User << endl;
                            cout << " Successfully decreased \n";
                            Sleep(0700);
                            Sleep(0700);
```

```
                    }

                    else {
                        cout << " 0 (kg/pc) can not decrease \n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

                case 48:
                    j = 1000;
                    break;
                }
            }
            break;

        // for Product1_2
        case 50:
        {
            for (int j = 0; j < 1000; j++)
            {
                system("cls");
                cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
                Product1_2.Display();
                position1 = 1;

                Table_For_Increasing_And_Decreasing();

                switch (getch())
                {
                case 49:
                    if (Product1_2.getQuantity() > 0) {
                        user.Product1_2_User++;
                        Product1_2.quantity--;
                        cout << " Quantity of " << Product1_2.getName() << " - " << user.Product1_2_User << endl;
                        cout << "    Successfully added \n";
                        Sleep(0700);
                        Sleep(0700);
                    }

                    else {
                        cout << " Product is over / finished. Sorry!\n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

                case 50:
                    if (user.Product1_2_User > 0) {
                        user.Product1_2_User--;
                        Product1_2.quantity++;
                        cout << " Quantity of " << Product1_2.getName() << " - " << user.Product1_2_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700);
                        Sleep(0700);
                    }

                    else {
                        cout << " 0 (kg/pc) can not decrease \n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

```

```
907
908                     case 48:
909                         j = 1000;
910                         break;
911                     }
912                 }
913             }
914             break;
915
916
917             // for Product1_3
918             case 51:
919             {
920                 for (int j = 0; j < 1000; j++)
921                 {
922                     system("cls");
923                     cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
924                     Product1_3.Display();
925                     position1 = 1;
926
927                     Table_For_Increasing_And_Decreasing();
928
929                     switch (_getch())
930                     {
931                     case 49:
932                         if (Product1_3.getQuantity() > 0) {
933                             user.Product1_3_User++;
934                             Product1_3.quantity--;
935                             cout << " Quantity of " << Product1_3.getName() << " - " << user.Product1_3_User << endl;
936                             cout << "    Successfully added \n";
937                             Sleep(0700);
938                             Sleep(0700);
939                         }
940
941                         else {
942                             cout << " Product is over / finished. Sorry!\n";
943                             Sleep(0700);
944                             Sleep(0700);
945                         }
946                         break;
947
948                     case 50:
949                         if (user.Product1_3_User > 0) {
950                             user.Product1_3_User--;
951                             Product1_3.quantity++;
952                             cout << " Quantity of " << Product1_3.getName() << " - " << user.Product1_3_User << endl;
953                             cout << "    Successfully decreased \n";
954                             Sleep(0700);
955                             Sleep(0700);
956                         }
957
```

```
                        }

                        else {
                            cout << " 0 (kg/pc) can not decrease \n";
                            Sleep(0700);
                            Sleep(0700);
                        }
                        break;

                    case 48:
                        j = 1000;
                        break;
                    }
                }
            }
            break;

            // Back to User menu
            case 48:
                k = 1000;
                User_Main_Menu();
                break;


            default:
                cout << "\n\n\t\t\t\t      Your choice is not available in Menu." << endl;
                cout << "\t\t\t\t    Please enter correct keys.\n\n";
                system("pause");
                break;
        }
    }
}

void LegendrySanninsMart :: Class_Of_Products2_Menu()
{
    for (int k = 0; k < 1000; k++)
    {
        system("cls");
        cout << "_____________________________________________________________________________________ \n";
        cout << "  L E G E N D R Y  S A N N I N S  M A R T                          P R O D U C T  2   \n";
        cout << "_____________________________________________________________________________________\n\n";

        cout << "    Products List  \t\t    Category\t\t      Price\t\t In Stock\n";
        Product2_1.Display();
        Product2_2.Display();
        Product2_3.Display();

        cout << " 0. Back\n\n";
        cout << " Your choice: ";
```

```
1005        switch (_getch())
1006        {
1007
1008        // Product2_1
1009        case 49:
1010        {
1011            for (int j = 0; j < 1000; j++)
1012            {
1013                system("cls");
1014                cout << "    Products List  \t\t    Category\t\t      Price\t\t In Stock\n";
1015                Product2_1.Display();
1016                position1 = 1;
1017
1018                Table_For_Increasing_And_Decreasing();
1019                switch (_getch())
1020                {
1021                case 49:
1022                    if (Product2_1.getQuantity() > 0) {
1023                        // checking for storage and user needs
1024                        Product2_1.quantity--;
1025                        user.Product2_1_User++;
1026                        cout << " Quantity of " << Product2_1.getName() << " - " << user.Product2_1_User << endl;
1027                        cout << "    Successfully added \n";
1028                        Sleep(0700);
1029                        Sleep(0700);
1030                    }
1031
1032                    else {
1033                        cout << " Product is over / finished. Sorry!\n";
1034                        Sleep(0700);
1035                        Sleep(0700);
1036                    }
1037                    break;
1038
1039                case 50:
1040                    if (user.Product2_1_User > 0) {
1041                        // Check for (-1 kg/pc)
1042                        Product2_1.quantity++;
1043                        user.Product2_1_User--;
1044                        cout << " Quantity of " << Product2_1.getName() << " - " <<user.Product2_1_User << endl;
1045                        cout << "    Successfully decreased \n";
1046                        Sleep(0700);
1047                        Sleep(0700);
1048                    }
1049
1050                    else {
1051                        cout << " 0 (kg/pc) can not decrease \n";
1052                        Sleep(0700);
1053                        Sleep(0700);
1054                    }
1055                    break;
1056
```

```
                case 48:
                    j = 1000;
                    break;
                }
            }
        }
        break;

        // Product2_2
        case 50:
        {
            for (int j = 0; j < 1000; j++)
            {
                system("cls");
                cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
                Product2_2.Display();
                position1 = 1;

                Table_For_Increasing_And_Decreasing();
                switch (_getch())
                {
                case 49:
                    if (Product2_2.getQuantity() > 0) {
                        // checking for storage and user needs
                        Product2_2.quantity--;
                        user.Product2_2_User++;
                        cout << " Quantity of " << Product2_2.getName() << " - " << user.Product2_2_User << endl;
                        cout << "    Successfully added \n";
                        Sleep(0700);
                        Sleep(0700);
                    }

                    else {
                        cout << " Product is over / finished. Sorry!\n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

                case 50:
                    if (user.Product2_2_User > 0) {
                        // Check for (-1 kg/pc)
                        Product2_2.quantity++;
                        user.Product2_2_User--;
                        cout << " Quantity of " << Product2_2.getName() << " - " << user.Product2_2_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700);
                        Sleep(0700);
                    }

                    else {
                        cout << " 0 (kg/pc) can not decrease \n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

                case 48:
                    j = 1000;
                    break;
                }
            }
        }
        break;
```

```
1122        // Product2_3
1123        case 51:
1124        {
1125            for (int j = 0; j < 1000; j++)
1126            {
1127                system("cls");
1128                cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
1129                Product2_3.Display();
1130                position1 = 1;
1131
1132                Table_For_Increasing_And_Decreasing();
1133                switch (_getch())
1134                {
1135                case 49:
1136                    if (Product2_3.getQuantity() > 0) {
1137                        // checking for storage and user needs
1138                        Product2_3.quantity--;
1139                        user.Product2_3_User++;
1140                        cout << " Quantity of " << Product2_3.getName() << " - " << user.Product2_3_User << endl;
1141                        cout << "    Successfully added \n";
1142                        Sleep(0700);
1143                        Sleep(0700);
1144                    }
1145
1146                    else {
1147                        cout << " Product is over / finished. Sorry!\n";
1148                        Sleep(0700);
1149                        Sleep(0700);
1150                    }
1151                    break;
1152
1153                case 50:
1154                    if (user.Product2_3_User > 0) {
1155                        // Check for (-1 kg/pc)
1156                        Product2_3.quantity++;
1157                        user.Product2_3_User--;
1158                        cout << " Quantity of " << Product2_3.getName() << " - " << user.Product2_3_User << endl;
1159                        cout << "    Successfully decreased \n";
1160                        Sleep(0700);
1161                        Sleep(0700);
1162                    }
1163
1164                    else {
1165                        cout << " 0 (kg/pc) can not decrease \n";
1166                        Sleep(0700);
1167                        Sleep(0700);
1168                    }
1169                    break;
1170
1171                case 48:
1172                    j = 1000;
1173                    break;
1174                }
1175            }
1176        }
1177        break;
```

```
1178
1179
1180        case 48:
1181        {
1182            k = 1000;
1183            User_Main_Menu();
1184        }
1185        break;
1186
1187
1188        default:
1189            cout << "\n\n\t\t\t       Your choice is not available in Menu." << endl;
1190            cout << "\t\t\t\t    Please enter correct keys.\n"
1191                 << endl;
1192            system("pause");
1193            break;
1194
1195        } // switch ends
1196
1197    } // loop ends
1198
1199 } // function ends
1200
1201 void LegendrySanninsMart :: Class_Of_Products3_Menu()
1202 {
1203     for (int k = 0; k < 1000; k++)
1204     {
1205
1206         system("cls");
1207         cout << "_____________________________________________________________________________________________ \n";
1208         cout << "  L E G E N D R Y  S A N N I N S  M A R T                                P R O D U C T  3    \n";
1209         cout << "_____________________________________________________________________________________________\n\n";
1210
1211         cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
1212         Product3_1.Display();
1213         Product3_2.Display();
1214         Product3_3.Display();
1215
1216         cout << " 0. Back\n\n";
1217         cout << " Your choice: ";
1218         switch (_getch())
1219         {
1220         // Product3_1
1221         case 49:
1222         {
1223             for (int j = 0; j < 1000; j++)
1224             {
1225                 system("cls");
1226
1227                 cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
1228                 Product3_1.Display();
1229                 position1 = 1;
1230
1231                 Table_For_Increasing_And_Decreasing();
1232                 switch (_getch())
1233                 {
1234                 case 49:
1235                     if (Product3_1.getQuantity() > 0) {
1236                         Product3_1.quantity--;
1237                         user.Product3_1_User++;
1238                         cout << " Quantity of " << Product3_1.getName() << " - " << user.Product3_1_User << endl;
1239                         cout << "     Successfully added \n";
1240                         Sleep(0700);
1241                         Sleep(0700);
1242                     }
```

```
1243
1244                    else {
1245                        cout << " Product is over / finished. Sorry!\n";
1246                        Sleep(0700);
1247                        Sleep(0700);
1248                    }
1249                    break;
1250
1251                case 50:
1252                    if (user.Product3_1_User > 0) {
1253                        Product3_1.quantity++;
1254                        user.Product3_1_User--;
1255                        cout << " Quantity of " << Product3_1.getName() << " - " << user.Product3_1_User << endl;
1256                        cout << "     Successfully decreased \n";
1257                        Sleep(0700);
1258                        Sleep(0700);
1259                    }
1260
1261                    else {
1262                        cout << " 0 (kg/pc) can not decrease \n";
1263                        Sleep(0700);
1264                        Sleep(0700);
1265                    }
1266                    break;
1267
1268                case 48:
1269                    j = 1000;
1270                    break;
1271                }
1272            }
1273        }
1274        break;
1275
1276        // Product3_2
1277        case 50:
1278        {
1279            for (int j = 0; j < 1000; j++)
1280            {
1281                system("cls");
1282                cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
1283                Product3_2.Display();
1284                position1 = 1;
1285
1286                Table_For_Increasing_And_Decreasing();
1287                switch (_getch())
1288                {
1289                case 49:
1290                    if (Product3_2.getQuantity() > 0) {
1291                        Product3_2.quantity--;
1292                        user.Product3_2_User++;
1293                        cout << " Quantity of " << Product3_2.getName() << " - " << user.Product3_2_User << endl;
1294                        cout << "     Successfully added \n";
1295                        Sleep(0700);
1296                        Sleep(0700);
1297                    }
1298
1299                    else {
1300                        cout << " Product is over / finished. Sorry!\n";
1301                        Sleep(0700);
1302                        Sleep(0700);
1303                    }
1304                    break;
1305                case 50:
1306                    if (user.Product3_2_User > 0) {
1307                        Product3_2.quantity++;
1308                        user.Product3_2_User--;
```

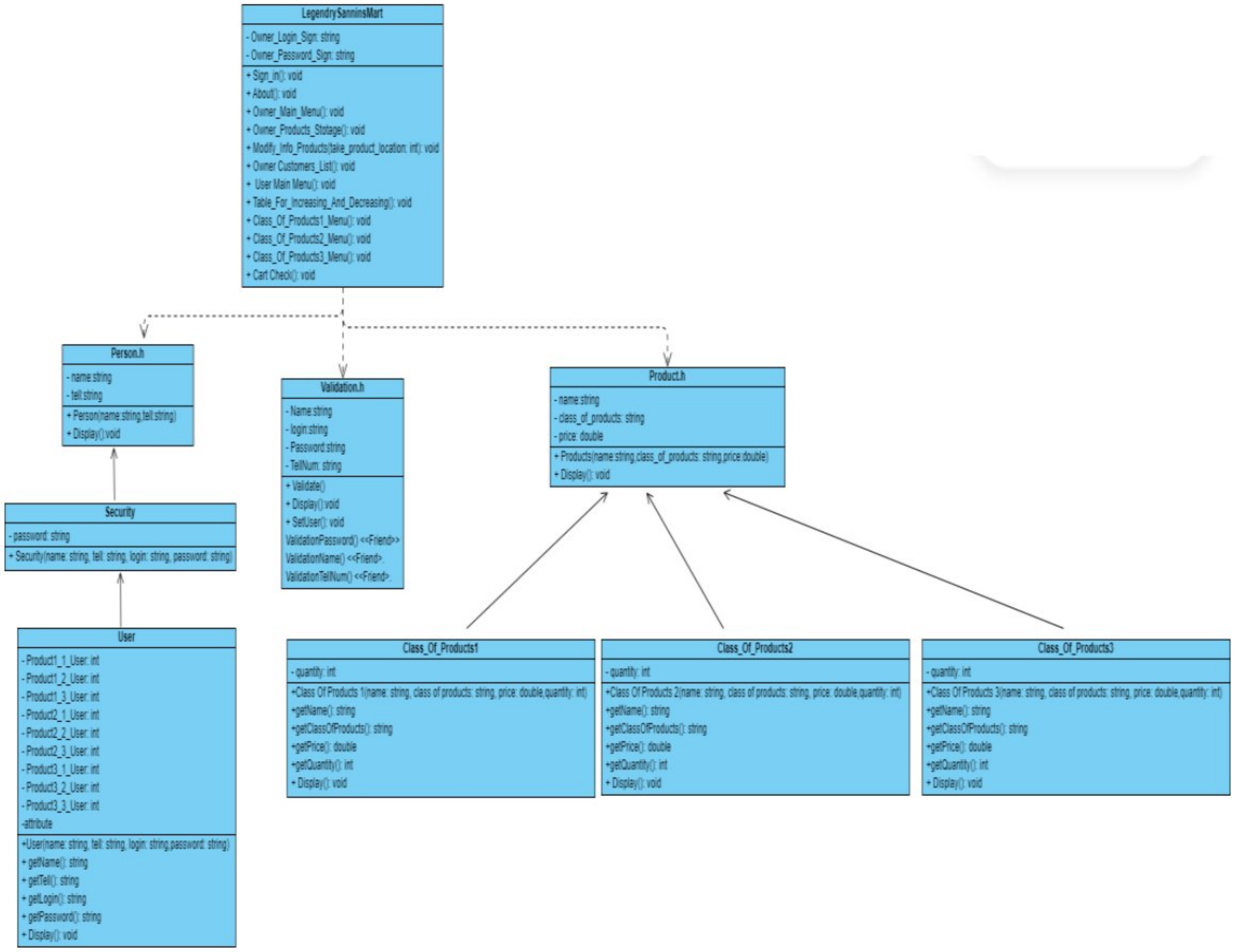
```
1305                case 50:
1306                    if (user.Product3_2_User > 0) {
1307                        Product3_2.quantity++;
1308                        user.Product3_2_User--;
1309                        cout << " Quantity of " << Product3_2.getName() << " - " << user.Product3_2_User << endl;
1310                        cout << "     Successfully decreased \n";
1311                        Sleep(0700);
1312                        Sleep(0700);
1313                    }
1314
1315                    else {
1316                        cout << " 0 (kg/pc) can not decrease \n";
1317                        Sleep(0700);
1318                        Sleep(0700);
1319                    }
1320                    break;
1321
1322                case 48:
1323                    j = 1000;
1324                    break;
1325                }
1326            }
1327        }
1328        break;
1329
1330        // Product3_3
1331        case 51:
1332        {
1333            for (int j = 0; j < 1000; j++)
1334            {
1335                system("cls");
1336                cout << "    Products List  \t\t    Category\t\t       Price\t\t In Stock\n";
1337                Product3_3.Display();
1338                position1 = 1;
1339
1340                Table_For_Increasing_And_Decreasing();
1341                switch (_getch())
1342                {
1343                case 49:
1344                    if (Product3_3.getQuantity() > 0) {
1345                        Product3_3.quantity--;
1346                        user.Product3_3_User++;
1347                        cout << " Quantity of " << Product3_3.getName() << " - " << user.Product3_3_User << endl;
1348                        cout << "     Successfully added \n";
1349                        Sleep(0700);
1350                        Sleep(0700);
1351                    }
1352
1353                    else {
1354                        cout << " Product is over / finished. Sorry!\n";
1355                        Sleep(0700);
1356                        Sleep(0700);
1357                    }
1358                    break;
1359
1360                case 50:
1361                    if (user.Product3_3_User > 0) {
1362                        Product3_3.quantity++;
1363                        user.Product3_3_User--;
1364                        cout << " Quantity of " << Product3_3.getName() << " - " << user.Product3_3_User << endl;
1365                        cout << "     Successfully decreased \n";
1366                        Sleep(0700);
1367                        Sleep(0700);
1368                    }
```

```
                    else {
                        cout << " 0 (kg/pc) can not decrease \n";
                        Sleep(0700);
                        Sleep(0700);
                    }
                    break;

                case 48:
                    j = 1000;
                    break;
                }
            }
        }
        break;

        // Back to F_User menu
        case 48:
            k = 1000;
            User_Main_Menu();
            break;

        default:
            cout << "\n\n\t\t\t       Your choice is not available in Menu." << endl;
            cout << "\t\t\t\t    Please enter correct keys.\n" << endl;
            system("pause");

        } // switch ends
    } // loop ends
} // function ends

void LegendrySanninsMart :: Cart_Check() {

    system("cls");
    cout << "_____________________________________________________________________________ \n";
    cout << "   L E G E N D R Y  S A N N I N S  M A R T                          C A R T   \n";
    cout << "_____________________________________________________________________________\n\n";

    // Check
    double Overall_Sum = 0;
    for (int i = 1; i <= 1; i++)
    {
        if (user.Product1_1_User > 0) {
            cout << "\n\t\t\t  " << i << ". " << Product1_1.getName() << " --- " << user.Product1_1_User << " --- Price: " << user.Product1_1_User * Product1_1.getPrice();
            i++;
            Overall_Sum += user.Product1_1_User * Product1_1.getPrice();
        }

        if (user.Product1_2_User > 0) {
            cout << "\n\t\t\t  " << i << ". " << Product1_2.getName() <<" --- " << user.Product1_2_User << " --- Price: " << user.Product1_2_User * Product1_2.getPrice();
            i++;
            Overall_Sum += user.Product1_2_User * Product1_2.getPrice();
        }
```

```
1422
1423        if (user.Product1_3_User > 0) {
1424            cout << "\n\t\t\t  " << i << ". " << Product1_3.getName() <<" --- " << user.Product1_3_User << " --- Price: " << user.Product1_3_User * Product1_3.getPrice();
1425            i++;
1426            Overall_Sum += user.Product1_3_User * Product1_3.getPrice();
1427        }
1428
1429        if (user.Product2_1_User > 0) {
1430            cout << "\n\t\t\t  " << i << ". " << Product2_1.getName() <<" --- " << user.Product2_1_User << " --- Price: " << user.Product2_1_User * Product2_1.getPrice();
1431            i++;
1432            Overall_Sum += user.Product2_1_User * Product2_1.getPrice();
1433        }
1434        if (user.Product2_2_User > 0) {
1435            cout << "\n\t\t\t  " << i << ". " << Product2_2.getName() <<" --- " << user.Product2_2_User << " --- Price: " << user.Product2_2_User * Product2_2.getPrice();
1436            i++;
1437            Overall_Sum += user.Product2_2_User * Product2_2.getPrice();
1438        }
1439        if (user.Product2_3_User > 0) {
1440            cout << "\n\t\t\t  " << i << ". " << Product2_3.getName() << " --- " << user.Product2_3_User << " --- Price: " << user.Product2_3_User * Product2_3.getPrice();
1441            i++;
1442            Overall_Sum += user.Product2_3_User * Product2_3.getPrice();
1443        }
1444        if (user.Product3_1_User > 0) {
1445            cout << "\n\t\t\t  " << i << ". " << Product3_1.getName() << " --- " << user.Product3_1_User << " --- Price: " << user.Product3_1_User * Product3_1.getPrice();
1446            i++;
1447            Overall_Sum += user.Product3_1_User * Product3_1.getPrice();
1448        }
1449        if (user.Product3_2_User > 0) {
1450            cout << "\n\t\t\t  " << i << ". " << Product3_2.getName() << " --- " << user.Product3_2_User << " --- Price: " << user.Product3_2_User * Product3_2.getPrice();
1451            i++;
1452            Overall_Sum += user.Product3_2_User * Product3_2.getPrice();
1453        }
1454        if (user.Product3_3_User > 0) {
1455            cout << "\n\t\t\t  " << i << ". " << Product3_3.getName()<< " --- " << user.Product3_3_User << " --- Price: " << user.Product3_3_User * Product3_3.getPrice();
1456            i++;
1457            Overall_Sum += user.Product3_3_User * Product3_3.getPrice();
1458        }
1459
1460        if (i == 1) {
1461            // if nothing go to Menu
1462            cout << "\n\t You do not have any product in 'CART'.\n";
1463            cout << "\tPress any key to go to 'Products Menu'\n\n" << endl;
1464            system("pause");
1465            Overall_Sum = 0;
1466            User_Main_Menu();
1467        }
1468        if (i > 1) {
1469            // Menu for buying or back
1470            cout << "\n\n\t\t\t  Overall Price: " << Overall_Sum << " Sums" << endl;
1471            cout << "\n\t   1. Buy now" << endl;
1472            cout << "\t   0. Products Menu" << endl;
1473
```

```
            switch (_getch()) {
            case 49: { //buy
                system("cls");
                ////
                cout << "_____________________________________________________________________________ \n";
                cout << "                                    C A R T \n";
                cout << "_____________________________________________________________________________\n\n";

                cout << " Money will be taken from your 'Telephone Number': " << endl;
                cout << " 1. OK" << endl;
                cout << " Press any key to go back..." << endl;
                switch (_getch())
                {
                    case 49:
                    {
                        cout << "\n\t  Transaction Successful!\n\t  Congratulations ! :)" << endl;
                        system("pause");
                        Overall_Sum = 0;
                        // Storage
                        user.Product1_1_User = 0; user.Product1_3_User = 0; user.Product1_2_User = 0;
                        user.Product2_1_User = 0; user.Product2_2_User = 0; user.Product2_3_User = 0;
                        user.Product3_1_User = 0; user.Product3_2_User = 0; user.Product3_3_User = 0;
                        User_Main_Menu();
                    }
                    break;

                    default:
                        Cart_Check();
                }
                system("pause");
            }
            break;

            case 48:
            { // Back
                system("cls");
                Overall_Sum = 0;
                User_Main_Menu();
            }
            break;

            default:
                Cart_Check();

                } // switch ends
            } //if ends
        }// loop end
} // function ends
```

UML DIAGRAM:



TECHNIQUES USED IN THIS PROGRAM:

- **INHERITANCE**
- **POLYMORPHISM (VIRTUAL FUNCTION)**
- **ENCAPSULATION**
- **PURE VIRTUAL FUNCTION.**

MOST CHALLENGING PART:

The most challenging part we faced while coding is firstly to decide that which approach should we go for, we have used multiple approaches so that anyone can easily understand our code, we have tried dictionary, different functions, used if-else statements in different ways and have tried many other things.

Additionally, managing the flow of data and interactions between different classes can be complex. Ensuring proper encapsulation, information hiding, and maintaining the integrity of the data poses challenges. Handling exceptions, error conditions, and ensuring data consistency across classes and methods are also crucial aspects to address.

MEMBERS CONTRIBUTION:

The work was done in the group but the individual contributions are:

MUHAIB SHAMSHER: [CR-22029]

- Syntax of code.
- Defining class interference.
- Header files.
- Including necessary libraries.
- Logics and calculations.
- Designing class structures.
- Use of tokens
- Implementation of whole project into code.

MUNEEZA BADAR: [CR-22022]

- Testing and quality insurance.
- Designing class structures.

- Use of conditional statements and loops.
- Debugs the bugs.
- UML Diagram.
- Logics and calculations.
- Features of project.
- Project Report.

MUHAMMAD HAMZA: [CR-22048]

- Main idea of the project.
- Project planning.
- Idea for the features.
- Debugs the bugs.
- Documentations and comments.
- Logics and calculations.
- Use of keywords.
- Error handling.

FUTURE EXPANSIONS/ LIMITATION OF APPROACH:

- Customer reviews and ratings: Add a feedback system that allows customers to rate and review products, helping other customers make informed purchasing decisions.
- Payment gateway integration: Integrate popular payment gateways to enable secure online payments, such as credit/debit cards, digital wallets, or online banking.
- Discounts, promotions, and coupons: Implement a system to apply discounts, promotions, or coupon codes to specific products or orders, enhancing customer engagement and loyalty.

CONCLUSION:

In conclusion, the C++ OOP Online Grocery Mart project has successfully developed a user-friendly and efficient online platform for customers to conveniently shop

for groceries. The project incorporated essential features like the loading page, registration page, login page, menu page, and validation mechanisms to provide a seamless and secure shopping experience. The loading page created a visually appealing start, while the registration and login pages allowed users to create accounts and authenticate themselves securely. The menu page provided a categorized display of grocery items for easy browsing. Additionally, robust validation techniques were implemented throughout the project to ensure data integrity and prevent errors. Overall, the project successfully met its objectives by offering an intuitive interface and incorporating necessary features to enhance the online grocery shopping experience.

CT-260 Object Oriented Programming

Complex Computing Problem (CCP)

Choose a problem in any application domain of computing and ***define*** its solution through an effective use of object oriented programming principles. The solution must be in the form of a well-documented and well-formatted report. It must be explained by employing in-depth computing or domain knowledge, and an approach that is based on well-founded principles.

The following instructions must be taken into consideration while preparing the project report. The report must reflect conceptual thinking and must properly be modularized according to functionality. The report of the solution must be organized clearly and should:

- ✓ describe the steps of your solution,
- ✓ the main object oriented programming techniques used,
- ✓ the limitations of the adopted approach.

Complex Computing Problem Assessment Rubrics

Criteria and Scales		
Good (5-4)	**Average (3-2)**	**Poor (1-0)**
Criterion 1 Problem Understanding: To what extent has the student has understood the problem? **(CP3: *Depth of Knowledge Required*)**		
The student has clearly understood the Problem.	The student has some clarity of the Problem.	The student has misunderstood the Problem.
Criterion 2 Requirement Identification: To what extent has the student outlined the object oriented programming principles required for solving the problem. ***(CP10: Requirement Identification)***		
The student has clearly outlined the required object oriented programming principles.	The student has outlined some of the required object oriented programming principles.	The students has not outlined the correct object oriented programming principles.
Criterion 3 Clarity of Solution: To what extent has the student defined the solution of the problem. **(CP3: *Depth of Knowledge Required*)**		
The student has clearly defined the solution which works for all possible test cases.	The student has defined a solution which works well for a few test cases.	The student has not defined a correct solution.

Total marks: _____/5____

Teacher's signature: _____