Online Bazar

Mini Project of SDF-II Lab

Submitted by:

Sahilsher Singh (9921103131)
Aman Dixit (9921103133)
Praveen Raj (9921103121)
Sarthak Chawla (9921103132)

Under the supervision of:
Shariq Murtuza
Arti Jain

Department of CSE/IT
Jaypee Institute of Information Technology University, Noida

May 2022

Table of Contents

Introduction

BAZAR is an e - Shopping system that allows the customers to buy products by following just a few simple steps. Moreover , it allows the owner to manage the products. Our system facilitates easy shopping for customers and easy management of products for the owner.

Problem statement :

1. Allow the user to sign up.

2. Allow the user to sign in and buy products.

3. System should provide information about the available products with their respective prices and stock and allow the users to buy them..

4. Allow the owner to manage the products buy changing their prices and stock.

Objectives:

1. Design a system where user can

- **I. Sign up by providing his/her details**
- **II. Sign in using his/her details**
- **III. Buy Products by choosing the desired products**

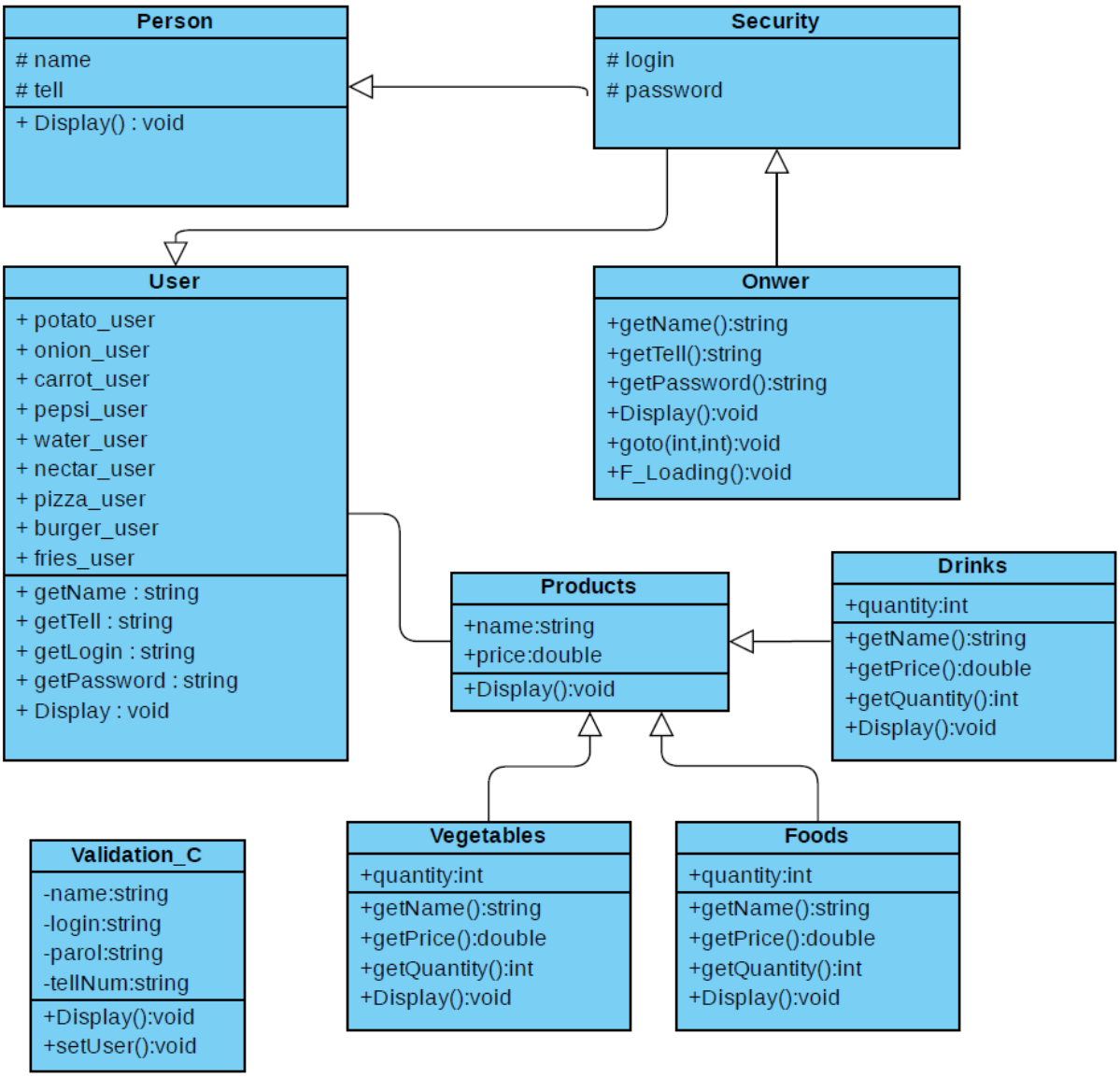
2. Make use of file handling to store the users' data that would be used later for sign in.

3. Make use of file handling and oops to store movies' information.

4. Make use of functions to allow the user to select the desired product.

5. Make use of basic C++ and oops functions to allow the owner to manage the products.

UML- Class Diagram



Project Code

```
#include <iostream>  // I/O stream
#include <string>    // Text
#include <ctime>     // Time sleep
#include <ctype.h>   // Validation
#include <conio.h>   // Getch
#include <Windows.h> // Loading
#include <fstream>   // File handling
#include <iomanip>   // setfill

using namespace std;


class Person {
protected:
    string name;
    string tell;
public:
    // Constructor for Person class
    Person(string name, string tell) {
        this->name = name; this->tell = tell;
    }

    virtual void Display() = 0;
};


////////////////////////////////////////////////
class Security : public Person { // Sub class of Person
protected:
    string login;
    string password;
public:

    // Constructor for Securoty class
    Security(string name, string tell, string login, string password)
:Person(name, tell) {
        this->login = login; this->password = password;
    }

};


////////////////////////////////////////////////
class User : public Security { // 1st Sub class of Securit
```

```
public:
    // Storage
    int Potatoes_User = 0, Onion_User = 0, Carrot_User = 0;
    int Water_User = 0, Pepsi_User = 0, Nectar_User = 0;
    int Pizza_User = 0, Burger_User = 0, Fries_User = 0;
    // Constructor for User's sign up
    User(string name, string tell, string login, string password) :
Security(name, tell, login, password) {
        this->name = name;
        this->tell = tell;
        this->login = login;
        this->password = password;
    }
    // get Name of User
    string getName() {
        return name;
    }
    // get Tell of User
    string getTell() {
        return tell;
    }
    // get Login of User
    string getLogin() {
        return login;
    }
    // get Password of User
    string getPassword() {
        return password;
    }
    // Display Info
    void Display() {
        cout << "\t\tName    : " << name << endl;
        cout << "\t\tTell    : " << tell << endl;
        cout << "\t\tLogin   : " << login << endl;
        cout << "\t\tPassword: " << password << endl;
    }
};


////////////////////////////////////////////////
class Owner : public Security { // 2nd Sub class of Security
public:
    // Constructor for User's sign up
    Owner(string name, string tell, string login, string password) :
Security(name, tell, login, password) { }
    // get Name of User
    string getName() {
```

```
        return name;
    }
    // get Tell of User
    string getTell() {
        return tell;
    }
    // get Login of User
    string getLogin() {
        return login;
    }
    // get Password of User
    string getPassword() {
        return password;
    }
    // Display Info
    void Display() {
        cout << "Name:     " << name << endl;
        cout << "Tell:     " << tell << endl;
        cout << "Login:    " << login << endl;
        cout << "Password: " << password << endl;
    }
};


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

void F_Loading() {
    cout << "\n\n\n\n\n\n";
    cout << "\t\t\t      ------------------------------------  \n";
    cout << "\t\t\t        E I G H T S O F T  A C A D E M Y     \n";
    cout << "\t\t\t      ------------------------------------  \n";
    cout << "\t\t\t                  B A Z A R           \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 <= 110000000; speed++);
        cout << a;
    }
    cout << endl;
}


////////////////////////////////////////////// price and name of Products
//// virtual class
class Products
{
public:
    string name; double price;

public:
    Products(string name, double price) {
        this->name = name; this->price = price;
    }
    virtual void Display() = 0;
};


////////////////////////////////////////////////
class Vegetables : public Products
{
public:
    int quantity;

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

    string getName() {
        return name;
    }

    double getPrice() {
        return price;
    }

    int getQuantity() {
        return quantity;
    }
```

```
    void Display() {
        cout << setw(11) << name << "\t" << price << setw(4) << "\t" <<
quantity << endl;
    }

};


////////////////////////////////////////////////
class Drinks : public Products
{
public:
    int quantity;

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

    string getName() {
        return name;
    }

    double getPrice() {
        return price;
    }

    int getQuantity() {
        return quantity;
    }

    void Display() {
        cout << setw(11) << name << "\t" << price << setw(4) << "\t" <<
quantity << endl;
    }
};


////////////////////////////////////////////////
class Foods : public Products
{
public:
    int quantity;

public:
    Foods(string name, double price, int quantity) : Products(name, price) {
```

```
        this->quantity = quantity;
    }

    string getName() {
        return name;
    }

    double getPrice() {
        return price;
    }

    int getQuantity() {
        return quantity;
    }

    void Display() {
        cout << setw(11) << name << "\t" << price << setw(4) << "\t" <<
quantity << endl;
    }
};


// Global Identifires for validation
int Num_Upper = 0, Num_Lower = 0, Num_Number = 0, Validation = 0;
string Login_Sign, Parol_Sign;


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


class Validation_C {
private:
    // Identifires
    string Name, Login, Parol, TellNum;
public:

    // Defoult Constructor
    Validation_C() {
        Name = "";
        Login = "";
        Parol = "";
        TellNum = "";
    }

    //Function display
    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   : " << Parol << endl;
    }

    // Set Info of User
    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 >> Parol;

        // Info sended to Memory
        Name_Memory = Name;
        Parol_Memory = Parol;
        Login_Memory = Login;
        TellNum_Memory = TellNum;
    }

    // Friend Functions
    //Validation Check for parol
    friend void ValidationParol(Validation_C User) {
        if (User.Parol.length() >= 6 && User.Parol.length() <= 15) {
            for (int i = 0; i < User.Parol.length(); i++) {
                if (isupper(User.Parol[i])) {  // Number of Upper Letters
                    Num_Upper += 1;
                }
                if (islower(User.Parol[i])) {  // Number of Lower Letters
                    Num_Lower += 1;
                }
                if (isdigit(User.Parol[i])) {  // Number of Digits
                    Num_Number += 1;
                }
            }
            if (Num_Upper >= 1 && Num_Upper <= 10 && Num_Lower >= 4 &&
Num_Lower <= 10 && Num_Number >= 2 && Num_Number <= 10) {
                Validation++;
                Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
            }
            else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
        }
        else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
    }
```

```
    //Validation Check for Name
    friend void ValidationName(Validation_C  User) {
        if (User.Name.length() >= 1 && User.Name.length() <= 15) {
            for (int i = 0; i < User.Name.length(); i++) {
                if (isupper(User.Name[i])) {
                    Num_Upper += 1;
                }
                if (islower(User.Name[i])) {
                    Num_Lower += 1;
                }
                if (isdigit(User.Name[i])) {
                    Num_Number += 1;
                }
            }

            if (Num_Upper <= 1 && Num_Lower >= 1 && Num_Lower <= 14 &&
Num_Number == 0) {
                Validation++;
                Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
            }
            else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
        }
        else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
    }
    //Validation Check for TellNum
    friend void ValidationTellNum(Validation_C User) {
        if (User.TellNum.length() >= 9 && User.TellNum.length() <= 12) {
            for (int i = 0; i < User.TellNum.length(); i++) {
                if (isupper(User.TellNum[i])) {
                    Num_Upper += 1;
                }
                if (islower(User.TellNum[i])) {
                    Num_Lower += 1;
                }
                if (isdigit(User.TellNum[i])) {
                    Num_Number += 1;
                }
            }
            if (Num_Upper == 0 && Num_Lower == 0 && Num_Number >= 9 &&
Num_Number <= 12) {
                Validation++;
                Num_Upper = 0; Num_Lower = 0; Num_Number = 0;
            }
            else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
        }
        else { Num_Upper = 0; Num_Lower = 0; Num_Number = 0; }
```

```
    }

};


// Objects
Validation_C User_Validtaion;
User User_1(Name_Memory, TellNum_Memory, Login_Memory, Parol_Memory);

// Product Types
Vegetables Onion("Onion", 3600.0, 10), Potatoes("Potatoes", 7890.0, 10),
Carrot("Carrot", 4890.0, 10);
Drinks Water("Water", 1590.0, 10), Pepsi("Pepsi", 3590.0, 10),
Nectar("Nectar", 7550.0, 10);
Foods Pizza("Pizza", 48000.0, 10), Burger("Burger", 19000, 10), Fries("Potatoe
Fries", 15000, 10);

// Global Values for CART part
long double Overall_Sum;
long double Ch_Price, Ch_Quantity;

//Password and login for Owner
string Owner_Login_Sign = "sahil_26";
string Owner_Parol_Sign = "Sahil123@";


//Functions
// Declaretion Functions
void F_General_Menu();
void F_Sign_in();
void F_Developers();
void F_Logo(); // Logo "BAZAR" for User
void F_Logo_Owner(); // Logo "BAZAR" for Owner


void F_User_Main_Menu(); // User's Main Menu
void F_Table_For_Increasing_And_Decreasing(); // User
void F_Vegetables_Fruits_Menu(); // User
void F_Water_Beverages_Menu(); // User
void F_Bread_Bakery_Menu(); // User
void F_Cart_Check();

void F_Owner_Main_Menu(); // Owner's Main Menu
void F_Owner_Products_Stotage();
void F_Owner_Customers_List();
```

```
//////////////////////////////////////////////
int main() {

    //Loading
    //F_Loading();
    F_General_Menu();

    system("pause");
    return 0;
}


void F_General_Menu() {
    // Main Menu
    for (int i = 0; i < 1000; i++) {
        system("cls");
        cout <<
"_____________________________________________________________________________
_____________________ \n";
        cout << "                          B A Z A R                          A U
T H E N T I C A T I O N              \n";
        cout <<
"_____________________________________________________________________________
_____________________\n\n";
        cout << "\t\t\t\t\t  Authentication \n" << endl;
        cout << "\t\t\t\t\t   1. Sign in" << endl;
        cout << "\t\t\t\t\t   2. Sign up" << endl;
        cout << "\t\t\t\t\t   3. About" << endl;
        cout << "\t\t\t\t\t   0. Exit" << endl << endl;
        cout << "\t\t\t\t\t  Your Choice: ";
        switch (_getch()) {
        case 49: { // Sign in

            system("cls");
            F_Sign_in();
            system("pause");
        }
               break;

        case 50: {  // Register
                    // Set details with validation
            for (int i = 0; i != 1;) {
                system("cls");
                cout <<
"_____________________________________________________________________________
_____________________ \n";
```

```
                cout << "                        B A Z A
R                           R E G I S T R A T I O N         \n";
                cout <<
"_____________________________________________________________________________
_____________________\n\n";
                cout << "\t\t\t\t       Example of Registration: " << endl;
                cout << "\t\t\t\t       _____________________________" << endl;
                cout << "\t\t\t\t       User Name  : sahil_sandhu " << endl;
                cout << "\t\t\t\t       Telephone  : 9700000002" << endl;
                cout << "\t\t\t\t       Login      : sahil_26 " << endl;
                cout << "\t\t\t\t       Password   : Sahil123@ " << endl;
                cout << "\t\t\t\t       _____________________________" << endl <<
endl;
                User_Validtaion.SetUser();

                ////////////
                ValidationParol(User_Validtaion);
                ValidationName(User_Validtaion);
                ValidationTellNum(User_Validtaion);
                if (Validation == 3) {
                    ////// File Handling For User Info
                    ofstream User_Info;
                    User_Info.open("User_Info.txt", ios::app);  // Opening
File
                    // Setting User Info in Memory
                    User_Info << Name_Memory << endl;
                    User_Info << TellNum_Memory << endl;
                    User_Info << Login_Memory << endl;
                    User_Info << Parol_Memory << endl;
                    User_Info.close();
                    ///////// End of the File Handling
                    cout << "\t\t\t\t _____________________________________" <<
endl;
                    cout << "\t\t\t\t      Correct Validation" << endl;
                    i++;
                    cout << "\t\t\t\tPlease press any key to continue..." <<
endl << endl;
                    system("Pause");
                    Validation = 0;
                }
                else {
                    cout << "\t\t\t\t _____________________________________" <<
endl;
                    cout << "\t\t\t\t      Invalid Validation \n\t\t\t\t(Write
correct format for password)" << endl;
```

```
                    cout << "\t\t\t    Please press any key to rewrite
details..." << endl << endl;
                    system("Pause"); system("cls");
                    Validation = 0;
                }
            }
        }
               break;

        case 51: {  // Developers
            system("cls");
            F_Developers();
        }
        case 48:{
            return ;
        }
               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");
        }
        }
    }

}


void F_Developers() {
    cout <<
"_____
_____ \n";
    cout << "                              B A Z A
R                             A B O U T        \n";
    cout <<
"_____
_____\n\n";
    cout << "\n\t\t\t\t'Bazar' Online Shopping Aplication " << endl << endl;
    cout << "\t\t\t\tTeam Members: " ;
    cout << "Sahilsher Singh  [9921103131]\n\t\t\t\t\t\t      Aman
Dixit   [9921103133]\n\t\t\t\t\t\t      Praveen
Raj   [9921103121]\n\t\t\t\t\t\t      Sarthak Chawla   [9921103132] " << endl;

    cout << "\n\n\t\t\t\tPress any key to go back to Menu \n\n" << endl;
    system("pause");
    F_General_Menu();
```

```
}

// Sign in Function
void F_Sign_in() {

    for (int i = 0; i < 1000; i++) {
        system("cls");
        cout <<
"_____________________________________________________________________________
_____________________ \n";
        cout << "                        B A Z A R                        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 << "                        B A Z A R                        O W N E
R          \n";
            cout <<
"_____________________________________________________________________________
_____________________\n\n";
            cout << "\t\t\t\t      Owner Authentication\n\n";
            cout << "\t\t\t\t\tLogin    : "; cin >> Login_Sign;
            cout << "\t\t\t\t\tPassword : "; cin >> Parol_Sign;
            if (Login_Sign == Owner_Login_Sign && Parol_Sign ==
Owner_Parol_Sign) {
                /////// Owner's Menu
                F_Owner_Main_Menu();
                //////
                system("pause");
            }
            else {  // If Login hasn't Registered
                cout << "\n\n\t\t\t        Your Login and Password are
Invalid." << endl;
```

```
                cout << "\t\t\t   Please press any key to go back to 'Sign in'
Menu.\n\n" << endl;
                system("pause");
                F_Sign_in();
            }
        }   break;

        case 50: {  // Sign in as User
            system("cls");
            cout <<
"_____________________________________________________________________________
_____________________ \n";
            cout << "                        B A Z A R                    C U S
T O M E R         \n";
            cout <<
"_____________________________________________________________________________
_____________________\n\n";
            cout << "\t\t\t\t      Customer Authentication\n\n";
            cout << "\t\t\t\t\tLogin    : "; cin >> Login_Sign;
            cout << "\t\t\t\t\tPassword : "; cin >> Parol_Sign;

            //////   File Handling For User Info
            ifstream Search;
            Search.open("User_Info.txt");
            while (Search) {
                Search >> Name_Memory;
                Search >> TellNum_Memory;
                Search >> Login_Memory;
                Search >> Parol_Memory;
                if (Login_Sign == Login_Memory && Parol_Sign == Parol_Memory)
{
                    User User_1(Name_Memory, TellNum_Memory, Login_Memory,
Parol_Memory);
                    F_User_Main_Menu();
                }
            }
            Search.close();
            ///////// End of the File Handling
            cout << "\n\n\t\t\t        Your Login and Password are Invalid."
<< endl;
            cout << "\t\t\t   Please press any key to go back to 'Sign in'
Menu.\n\n" << endl;
            system("pause");
            F_Sign_in();

        }
```

```
                break;

        case 48: {  // Back
            system("cls");
            i = 1000;
            F_General_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");
        }
        }
    }

}

// User
void F_Logo() {
    system("cls");
    cout <<
"_____________________________________________________________________________
_____________________ \n";
    cout << "                          B A Z A R                    8. Account
Info         \n";
    cout <<
"_____________________________________________________________________________
_____________________\n\n";
}


void F_User_Main_Menu() {
    // Entering as a Guest
    for (int i = 0; i < 1000; i++) {

        F_Logo();

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

```
        switch (_getch()) {
        case 49: {
            F_Vegetables_Fruits_Menu();
        } break;

        case 50: {
            F_Water_Beverages_Menu();
        } break;

        case 51: {
            F_Bread_Bakery_Menu();
        }
        case 52:
        {
            F_Cart_Check();
        }
        break;
        case 48: {  // Back to Menu
            system("cls");
            i = 1000;
            F_Sign_in();
        } break;

        case 56: { // User info
            system("cls");
            cout <<
"_____________________________________________________________________________
_____________________ \n";
            cout << "                          B A Z A R                          U
S E R  I N F O         \n";
            cout <<
"_____________________________________________________________________________
_____________________\n\n";
            cout << "\t\t\t\t    User Information:" << endl << endl;
            cout << "\t\t\t\t      User Name  : " << Name_Memory << endl;
            cout << "\t\t\t\t      Telephone  : " << TellNum_Memory << endl;
            cout << "\t\t\t\t      Login      : " << Login_Memory << endl;
            cout << "\t\t\t\t      Password   : " << Parol_Memory << endl <<
endl << endl;
            system("pause");
        }
               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
    } // for loop
}


void F_Table_For_Increasing_And_Decreasing() {

    cout << "     (+)   'Press 1'                          \n";
    cout << "     (-)   'Press 2'                          \n";
    cout << "     (0)   'Back'                             \n";
    cout <<
"_____________________________________________________________________
_____________________\n\n";
    cout << "     Add to Cart:  \n";
    //cin >> VariableForIncreasingAndDecreasing;
}


void F_Vegetables_Fruits_Menu() {
    for (int k = 0; k < 1000; k++) {
        F_Logo();
        cout << "      Categories -> Vegetables & Fruits \t\t Cart \n";
        cout <<
"_____________________________________________________________________
_____________________\n\n";
        cout << " 1. Potatoes, Weight \t\t\t\t\t"; cout <<
Potatoes.getQuantity() << " (kg)" << endl;
        cout << "    " << Potatoes.getPrice() << " RS. for 1 kg\n\n";
        cout << " 2. Yellow Carrot, Weight \t\t\t\t"; cout <<
Carrot.getQuantity() << " (kg)" << endl;
        cout << "    " << Carrot.getPrice() << " RS. for 1 kg\n\n";
        cout << " 3. Onion, Weight \t\t\t\t\t"; cout << Onion.getQuantity() <<
" (kg)" << endl;
        cout << "    " << Onion.getPrice() << " RS. for 1 kg\n\n";
        cout << " 0. Back\n\n";
        cout << " Your choice: ";
        switch (_getch()) {
            // for potatoes
        case 49: {
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Vegetables & Fruits \t\t Cart
\n";
```

```
                cout <<
"_____
_____\n";
                cout << "     Potatoes, Weight \t\t\t\t\t"; cout <<
Potatoes.getQuantity() << " (kg)" << endl;
                cout << "     " << Potatoes.getPrice() << " RS. for 1 kg\n\n";
                //

                F_Table_For_Increasing_And_Decreasing();

                switch (_getch()) {
                case 49:
                    if (Potatoes.getQuantity() > 0) { // checking for storage
and user needs
                        User_1.Potatoes_User++;
                        Potatoes.quantity--;
                        cout << " Quantity of Potatoes (kg): " <<
User_1.Potatoes_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_1.Potatoes_User > 0) { // Check for (-1 kg)
                        User_1.Potatoes_User--;
                        Potatoes.quantity++;
                        cout << " Quantity of Potatoes (kg): " <<
User_1.Potatoes_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (kg) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for potato
            }// 'for' loop for potato
        }
               break;
               // for carrot
```

```
        case 50: {
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Vegetables & Fruits \t\t Cart
\n";
                cout <<
"_____________________________________________________________________________
_____________________\n";
                cout << "     Carrot, Weight \t\t\t\t\t"; cout <<
Carrot.getQuantity() << " (kg)" << endl;
                cout << "     " << Carrot.getPrice() << " RS. for 1 kg\n\n";
                //

                F_Table_For_Increasing_And_Decreasing();

                switch (_getch()) {
                case 49:
                    if (Carrot.getQuantity() > 0) { // checking for storage
and user needs
                        User_1.Carrot_User++;
                        Carrot.quantity--;
                        cout << " Quantity of Carrot (kg): " <<
User_1.Carrot_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_1.Carrot_User > 0) { // Check for (-1 kg)
                        User_1.Carrot_User--;
                        Carrot.quantity++;
                        cout << " Quantity of Carrot (kg): " <<
User_1.Carrot_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (kg) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
```

```
                }// 'switch' for potato
            }// 'for' loop for potato
        }
               break;
               // for onion
        case 51: {
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Vegetables & Fruits \t\t Cart
\n";
                cout <<
"_____________________________________________________________________
_____________________\n";
                cout << "     Onion, Weight \t\t\t\t\t"; cout <<
Onion.getQuantity() << " (kg)" << endl;
                cout << "     " << Onion.getPrice() << " RS. for 1 kg\n\n";
                //

                F_Table_For_Increasing_And_Decreasing();

                switch (_getch()) {
                case 49:
                    if (Onion.getQuantity() > 0) { // checking for storage and
user needs
                        User_1.Onion_User++;
                        Onion.quantity--;
                        cout << " Quantity of Onion (kg): " <<
User_1.Onion_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_1.Onion_User > 0) { // Check for (-1 kg)
                        User_1.Onion_User--;
                        Carrot.quantity++;
                        cout << " Quantity of Carrot (kg): " <<
User_1.Onion_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
```

```
                    else { cout << " 0 (kg) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for potato
            }// 'for' loop for potato
        }
               break;
               // Back to F_User menu
        case 48: {   k = 1000;
            //F_User_Main_Menu();
        }
               break;
        case 56: { // User info
            system("cls");
            cout <<
"_____________________________________________________________________________
_____________________ \n";
            cout << "                        B A Z A R                          U
S E R  I N F O         \n";
            cout <<
"_____________________________________________________________________________
_____________________\n\n";
            cout << "\t\t\t\t    User Information:" << endl << endl;
            cout << "\t\t\t\t      User Name  : " << Name_Memory << endl;
            cout << "\t\t\t\t      Telephone  : " << TellNum_Memory << endl;
            cout << "\t\t\t\t      Login      : " << Login_Memory << endl;
            cout << "\t\t\t\t      Password   : " << Parol_Memory << endl <<
endl << endl;
            system("pause");
        }
               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 F_Water_Beverages_Menu() {
    for (int k = 0; k < 1000; k++) {
        F_Logo();
```

```
        cout << "      Categories -> Water & Beverages \t\t\t Cart \n";
        cout <<
"_____
_____\n\n";
        cout << " 1. Water, Hydrolife without gas 500ml \t\t\t"; cout <<
Water.getQuantity() << " (pc)" << endl;
        cout << "    " << Water.getPrice() << " RS. for 1 pc\n\n";
        cout << " 2. Pepsi 500ml   \t\t\t\t\t"; cout << Pepsi.getQuantity() <<
" (pc)" << endl;
        cout << "    " << Pepsi.getPrice() << " RS. for 1 pc\n\n";
        cout << " 3. Nectar, Zet Apple 125ml \t\t\t\t"; cout <<
Nectar.getQuantity() << " (pc)" << endl;
        cout << "    " << Nectar.getPrice() << " RS. for 1 pc\n\n";
        cout << " 0. Back\n\n";
        cout << " Your choice: ";
        switch (_getch()) {
        case 49: { // Water
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Water & Beverages \t\t\t Cart
\n";
                cout <<
"_____
_____\n";
                cout << "     Water, Hydrolife without gas 500ml \t\t"; cout
<< Water.getQuantity() << " (pc)" << endl;
                cout << "     " << Water.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
                case 49:
                    if (Water.getQuantity() > 0) { // checking for storage and
user needs
                        Water.quantity--;
                        User_1.Water_User++;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Water_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_1.Water_User > 0) { // Check for (-1 kg)
                        Water.quantity++;
                        User_1.Water_User--;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Water_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for water
            }// 'for' loop for water
        }
               break;
        case 50: { // Pepsi
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Water & Beverages \t\t\t Cart
\n";
                cout <<
"_____________________________________________________________________________
_____________________\n";
                cout << "      Pepsi 500ml \t\t\t\t\t"; cout <<
Pepsi.getQuantity() << " (pc)" << endl;
                cout << "      " << Pepsi.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
                case 49:
                    if (Pepsi.getQuantity() > 0) { // checking for storage and
user needs
                        Pepsi.quantity--;
                        User_1.Pepsi_User++;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Pepsi_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_1.Pepsi_User > 0) { // Check for (-1 kg)
                        Pepsi.quantity++;
                        User_1.Pepsi_User--;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Pepsi_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for water
            }// 'for' loop for water
        }
               break;
        case 51: { // Nectar
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Water & Beverages \t\t\t Cart
\n";
                cout <<
"_____________________________________________________________________________
_____________________\n";
                cout << "      Nectar 500ml \t\t\t\t\t"; cout <<
Nectar.getQuantity() << " (pc)" << endl;
                cout << "      " << Nectar.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
                case 49:
                    if (Nectar.getQuantity() > 0) { // checking for storage
and user needs
                        Nectar.quantity--;
                        User_1.Nectar_User++;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Nectar_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_1.Nectar_User > 0) { // Check for (-1 kg)
                        Nectar.quantity++;
                        User_1.Nectar_User--;
                        cout << " Quantity of Bottles (pc): " <<
User_1.Nectar_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for water
            }// 'for' loop for water
        }
               break;
               // Back to F_User menu
        case 48: {   k = 1000;
            F_User_Main_Menu(); }
               break;
        case 56: { // User info
            system("cls");
            cout <<
"_____
_____ \n";
            cout << "                        B A Z A
R                          U S E R  I N F O         \n";
            cout <<
"_____
_____\n\n";
            cout << "\t\t\t\t    User Information:" << endl << endl;
            cout << "\t\t\t\t      User Name  : " << Name_Memory << endl;
            cout << "\t\t\t\t      Telephone  : " << TellNum_Memory << endl;
            cout << "\t\t\t\t      Login      : " << Login_Memory << endl;
            cout << "\t\t\t\t      Password   : " << Parol_Memory << endl <<
endl << endl;
            system("pause");
        }
               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 F_Bread_Bakery_Menu() {
    for (int k = 0; k < 1000; k++) {
        F_Logo();
        cout << "      Categories -> Bread & Bakery Products \t\t Cart \n";
        cout <<
"_____________________________________________________________________________
_____________________\n\n";
        cout << " 1. Pizza \t\t\t\t\t\t"; cout << Pizza.getQuantity() << "
(pc)" << endl;
        cout << "    " << Pizza.getPrice() << " RS. for 1 pc\n\n";
        cout << " 2. Burger \t\t\t\t\t"; cout << Burger.getQuantity() << "
(pc)" << endl;
        cout << "    " << Burger.getPrice() << " RS. for 1 pc\n\n";
        cout << " 3. Potatoe Fries \t\t\t\t"; cout << Fries.getQuantity() <<
" (pc)" << endl;
        cout << "    " << Fries.getPrice() << " RS. for 1 pc\n\n";
        cout << " 0. Back\n\n";
        cout << " Your choice: ";
        switch (_getch()) {
        case 49: { // Pizza
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Food Products \t\t Cart \n";
                cout <<
"_____________________________________________________________________________
_____________________\n";
                cout << "     Pizza \t\t\t\t\t"; cout << Pizza.getQuantity()
<< " (pc)" << endl;
                cout << "     " << Pizza.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
                case 49:
                    if (Pizza.getQuantity() > 0) { // checking for storage and
user needs
```

```
                        Pizza.quantity--;
                        User_1.Pizza_User++;
                        cout << " Quantity of Pizza (pc): " <<
User_1.Pizza_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_1.Pizza_User > 0) { // Check for (-1 pc)
                        Pizza.quantity++;
                        User_1.Pizza_User--;
                        cout << " Quantity of Pizza (pc): " <<
User_1.Pizza_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for bun bread
            }// 'for' loop for bun bread
        }
               break;
        case 50: { // Burger
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Food Products \t\t Cart \n";
                cout <<
"_____
_____\n";
                cout << "     Burger \t\t\t\t\t"; cout << Burger.getQuantity()
<< " (pc)" << endl;
                cout << "     " << Burger.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
                case 49:
```

```
                    if (Burger.getQuantity() > 0) { // checking for storage
and user needs
                        Burger.quantity--;
                        User_1.Burger_User++;
                        cout << " Quantity of Burger (pc): " <<
User_1.Burger_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_1.Burger_User > 0) { // Check for (-1 pc)
                        Burger.quantity++;
                        User_1.Burger_User--;
                        cout << " Quantity of Burger (pc): " <<
User_1.Burger_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for bun bread
            }// 'for' loop for bun bread
        }
               break;
        case 51: { // Fries
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout << "      Categories -> Food Products \t\t Cart \n";
                cout <<
"_____
_____\n";
                cout << "     Potatoe Fries \t\t\t\t"; cout <<
Fries.getQuantity() << " (pc)" << endl;
                cout << "     " << Fries.getPrice() << " RS. for 1 pc\n\n";

                F_Table_For_Increasing_And_Decreasing();
                switch (_getch()) {
```

```
                case 49:
                    if (Fries.getQuantity() > 0) { // checking for storage and
user needs
                        Fries.quantity--;
                        User_1.Fries_User++;
                        cout << " Quantity of Fries (pc): " <<
User_1.Fries_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_1.Fries_User > 0) { // Check for (-1 pc)
                        Burger.quantity++;
                        User_1.Fries_User--;
                        cout << " Quantity of Fries (pc): " <<
User_1.Fries_User << endl;
                        cout << "    Successfully decreased \n";
                        Sleep(0700); Sleep(0700);
                    }
                    else { cout << " 0 (pc) can not decrease \n"; Sleep(0700);
Sleep(0700); }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch' for bun bread
            }// 'for' loop for bun bread
        }
               break;
               // Back to F_User menu
        case 48: {   k = 1000;
            F_User_Main_Menu(); }
               break;
        case 56: { // User info
            system("cls");
            cout <<
"_____
_____ \n";
            cout << "                          B A Z A
R                       U S E R  I N F O         \n";
```

```
            cout <<
"_____
_____\n\n";
            cout << "\t\t\t\t    User Information:" << endl << endl;
            cout << "\t\t\t\t      User Name  : " << Name_Memory << endl;
            cout << "\t\t\t\t      Telephone  : " << TellNum_Memory << endl;
            cout << "\t\t\t\t      Login      : " << Login_Memory << endl;
            cout << "\t\t\t\t      Password   : " << Parol_Memory << endl <<
endl << endl;
            system("pause");
        }
               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

// Cart Function
void F_Cart_Check() {
    system("cls");
    cout <<
"_____
_____ \n";
    cout << "                                                  C A R T \n";
    cout <<
"_____
_____\n\n";
    // Check
    for (int i = 1; i <= 1; i++) {
        if (User_1.Potatoes_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Potatoes, Weight    " <<
User_1.Potatoes_User << " (kg)    Price: " << User_1.Potatoes_User *
Potatoes.getPrice();
            i++;
            Overall_Sum += User_1.Potatoes_User * Potatoes.getPrice();
        }

        if (User_1.Carrot_User > 0) {
```

```
            cout << "\n\t\t\t  " << i << "." << "Carrot, Weight      " <<
User_1.Carrot_User << " (kg)    Price: " << User_1.Carrot_User *
Carrot.getPrice();
            i++;
            Overall_Sum += User_1.Carrot_User * Carrot.getPrice();
        }
        if (User_1.Onion_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Onion, Weight       " <<
User_1.Onion_User << " (kg)    Price: " << User_1.Onion_User *
Onion.getPrice();
            i++;
            Overall_Sum += User_1.Onion_User * Onion.getPrice();
        }
        if (User_1.Water_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Water, 500ml        " <<
User_1.Water_User << " (pc)    Price: " << User_1.Water_User *
Water.getPrice();
            i++;
            Overall_Sum += User_1.Water_User * Water.getPrice();
        }
        if (User_1.Pepsi_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Pepsi, 500ml        " <<
User_1.Pepsi_User << " (pc)    Price: " << User_1.Pepsi_User *
Pepsi.getPrice();
            i++;
            Overall_Sum += User_1.Pepsi_User * Pepsi.getPrice();
        }
        if (User_1.Nectar_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Nectar, 500ml       " <<
User_1.Nectar_User << " (pc)    Price: " << User_1.Nectar_User *
Nectar.getPrice();
            i++;
            Overall_Sum += User_1.Nectar_User * Nectar.getPrice();
        }
        if (User_1.Pizza_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Pizza\t        " <<
User_1.Pizza_User << " (pc)\t  Price: " << User_1.Pizza_User *
Pizza.getPrice();
            i++;
            Overall_Sum += User_1.Pizza_User * Pizza.getPrice();
        }
        if (User_1.Burger_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Burger\t        " <<
User_1.Burger_User << " (pc)\t  Price: " << User_1.Burger_User *
Burger.getPrice();
            i++;
```

```
            Overall_Sum += User_1.Burger_User * Burger.getPrice();
        }
        if (User_1.Fries_User > 0) {
            cout << "\n\t\t\t  " << i << "." << "Fries\t        " <<
User_1.Fries_User << " (pc)\t  Price: " << User_1.Fries_User *
Fries.getPrice();
            i++;
            Overall_Sum += User_1.Fries_User * Fries.getPrice();
        }

        if (i == 1) { // if nothing go to Menu
            cout << "\n\t You do not have any product in 'CART'.\n";
            cout << "\tPress any key to go to 'Products Menu'\n\n" << endl;
            system("pause");
            Overall_Sum = 0;
            F_User_Main_Menu();
        }
        if (i > 1) { // Menu for buying or back
            cout << "\n\n\t\t\t  Overall Price: " << Overall_Sum << " Sums" <<
endl;
            cout << "\n\t   1. Buy now" << endl;
            cout << "\t   0. Products Menu" << endl;

            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  Transaction Successful!\n  Congratulations !
:)" << endl;
                    system("pause");
```

```
                    Overall_Sum = 0;
                    // Storage
                    User_1.Potatoes_User = 0; User_1.Onion_User = 0;
User_1.Carrot_User = 0;
                    User_1.Water_User = 0; User_1.Pepsi_User = 0;
User_1.Nectar_User = 0;
                    User_1.Pizza_User = 0; User_1.Burger_User = 0;
User_1.Fries_User = 0;
                    F_User_Main_Menu();
                }
                       break;

                default:  F_Cart_Check();

                } // switch of case 49

                ////
                system("pause");
            }
                   break;
            case 48: { // Back
                system("cls");
                Overall_Sum = 0;
                F_User_Main_Menu();
            }
                   break;
            default: F_Cart_Check();
            } // switch ends

        } //if ends

    }// loop end
    //system("pause");
} // function ends


////////////////////////////////////////////////////////////////////////////////
/
//  Owner part

void F_Logo_Owner() {
    system("cls");
    cout <<
"_____________________________________________________________________________
_____________________ \n";
    cout << "                        B A Z A R                    O W N E R  M
E N U        \n";
```

```
    cout <<
"_____
_____\n\n";
}


void F_Owner_Main_Menu() {
    system("cls");
    cout << endl << endl;
    // Entering as a Owner of shop
    for (int k = 0; k < 1000; k++) {
        F_Logo_Owner();
        cout << "       Main Menu\n\n";
        cout << "       1. Products in stock \n\n";
        cout << "       0. Back\n\n";
        cout << "       Your choice: ";

        switch (_getch()) {
        case 49: {
            F_Owner_Products_Stotage();
        } break;

        case 48: {  // Back to Menu
            system("cls");
            k = 1000;
            F_Sign_in();
        } 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
    } // for loop
}




void F_Owner_Products_Stotage() {
    for (int i = 0; i < 1000; i++) {
        F_Logo_Owner();
        cout << "  Products
List                                Category                    Price
 In Stock\n";
```

```
        cout <<
"_____
_____ \n";
        cout << " 1. Potatoes, Weight                          Vegetables &
Fruits          " << Potatoes.getPrice() << "\t\t  " << Potatoes.getQuantity()
<< endl;
        cout << " 2. Yellow Carrot, Weight                     Vegetables &
Fruits          " << Carrot.getPrice() << "\t\t  " << Carrot.getQuantity() <<
endl;
        cout << " 3. Onion, Weight                             Vegetables &
Fruits          " << Onion.getPrice() << "\t\t  " << Onion.getQuantity() <<
endl;
        cout << " 4. Water                                     Water &
Beverages            " << Water.getPrice() << "\t\t  " << Water.getQuantity()
<< endl;
        cout << " 5. Pepsi                                     Water &
Beverages            " << Pepsi.getPrice() << "\t\t  " << Pepsi.getQuantity()
<< endl;
        cout << " 6. Nector                                    Water &
Beverages            " << Nectar.getPrice() << "\t\t  " <<
Nectar.getQuantity() << endl;
        cout << " 7. Pizza                                     Bread & Bakery
Products      " << Pizza.getPrice() << "\t  " << Pizza.getQuantity() << endl;
        cout << " 8. Burger                                    Bread & Bakery
Products      " << Burger.getPrice() << "\t  " << Burger.getQuantity() <<
endl;
        cout << " 9. Potatoe Fries                             Bread & Bakery
Products      " << Fries.getPrice() << "\t  " << Fries.getQuantity() << endl;
        cout << " \n 0. Back\n";
        cout << " Make changes in: ";
        switch (_getch())
        {
        case '1':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____
_____ \n";
                cout << " Potatoes, Weight                          Vegetables
& Fruits             " << Potatoes.getPrice() << "\t\t " <<
Potatoes.getQuantity() << endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n";
```

```
                switch (_getch()) {
                case 49:
                    cout << " Enter a new price: ";
                    cin >> Ch_Price;
                    if (Ch_Price >= 0) {
                        Potatoes.price = Ch_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;
                case 50:
                    cout << " Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Potatoes.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }

                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '2':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
```

```
                cout << " Yellow Carrot, Weight                    Vegetables
& Fruits             " << Carrot.getPrice() << "\t\t " << Carrot.getQuantity()
<< endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 3. Go back \n Press '1' or '2' or '0'\n\n ";
                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;
                    if (Ch_Price >= 0) {
                        Carrot.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;

                    if (Ch_Quantity > 0) {
                        Carrot.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }

                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '3':
            for (int j = 0; j < 1000; j++) {
                system("cls");
```

```
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Onion, Weight                                  Vegetables
& Fruits             " << Onion.getPrice() << "\t\t " << Onion.getQuantity()
<< endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";
                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;
                    if (Ch_Price >= 0) {
                        Onion.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Onion.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '4':
```

```
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Water, Hydrolife without gas 750ml        Water &
Beverages                " << Water.getPrice() << "\t\t " <<
Water.getQuantity() << endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";

                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;

                    if (Ch_Price >= 0) {
                        Water.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Water.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
```

```
                }// 'switch'
            }// 'for' loop
            break;
        case '5':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Drink, Aloe Original 500ml                Water &
Beverages                " << Pepsi.getPrice() << "\t\t " <<
Pepsi.getQuantity() << endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";

                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;

                    if (Ch_Price >= 0) {
                        Pepsi.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;

                    if (Ch_Quantity > 0) {
                        Pepsi.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
```

```
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '6':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Nectar, Zet Apple 125ml                  Water &
Beverages                " << Nectar.getPrice() << "\t\t " <<
Nectar.getQuantity() << endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";

                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;
                    if (Ch_Price >= 0) {
                        Nectar.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Nectar.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
```

```
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '7':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Pizza                                      Bread &
Bakery Products           " << Pizza.getPrice() << "\t " <<
Pizza.getQuantity() << endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";
                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;
                    if (Ch_Price >= 0) {
                        Pizza.price = Ch_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;
                case 50:
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Pizza.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
```

```
                    Sleep(0700); Sleep(0700);
                }
                else {
                    cout << " Quantity cannot be negative\n";
                    Sleep(0700); Sleep(0700);
                }
                break;
            case 48:
                j = 1000;
                break;
            }// 'switch'
        }// 'for' loop
        break;
    case '8':
        for (int j = 0; j < 1000; j++) {
            system("cls");
            cout <<
"\n  Product                                    Category
Price         In Stock\n";
            cout <<
"_____________________________________________________________________________
_____________________ \n";
            cout << " Burger                                    Bread &
Bakery Products           " << Burger.getPrice() << "\t" <<
Burger.getQuantity() << endl;
            cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";
            switch (_getch()) {
            case 49:
                cout << "Enter a new price: ";
                cin >> Ch_Price;
                if (Ch_Price >= 0) {
                    Burger.price = Ch_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;
            case 50:
                cout << "Enter a new quantity in storage: ";
                cin >> Ch_Quantity;
                if (Ch_Quantity > 0) {
```

```
                        Burger.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '9':
            for (int j = 0; j < 1000; j++) {
                system("cls");
                cout <<
"\n  Product                                    Category
Price         In Stock\n";
                cout <<
"_____________________________________________________________________________
_____________________ \n";
                cout << " Potatoe Fries                          Bread &
Bakery Products           " << Fries.getPrice() << "\t" << Fries.getQuantity()
<< endl;
                cout << "\n 1. Change price \n 2. Change the quantity in
storage\n 0. Go back \n Press '1' or '2' or '0'\n\n ";
                switch (_getch()) {
                case 49:
                    cout << "Enter a new price: ";
                    cin >> Ch_Price;

                    if (Ch_Price >= 0) {
                        Fries.price = Ch_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;
                case 50:
```

```
                    cout << "Enter a new quantity in storage: ";
                    cin >> Ch_Quantity;
                    if (Ch_Quantity > 0) {
                        Fries.quantity = Ch_Quantity;
                        cout << " Successfully changed!\n";
                        Sleep(0700); Sleep(0700);
                    }
                    else {
                        cout << " Quantity cannot be negative\n";
                        Sleep(0700); Sleep(0700);
                    }
                    break;
                case 48:
                    j = 1000;
                    break;
                }// 'switch'
            }// 'for' loop
            break;
        case '0': {  // Back to Menu
            system("cls");
            i = 1000;
            F_Owner_Main_Menu();
        } break;

        case 'i' || 'I': { // User info
            system("cls");
            cout << "\n\t\t\t   User Information:" << endl;;
            cout << "\t\t    _____________________________" << endl <<
endl;;
            cout << "\t\t      User Name  : " << Name_Memory << endl;
            cout << "\t\t      Telephone  : " << TellNum_Memory << endl;
            cout << "\t\t      Login      : " << Login_Memory << endl;
            cout << "\t\t      Password   : " << Parol_Memory << endl << endl
<< endl;
            system("pause");
        }
                        break;
        default: { cout << "\n\t\t      Your choice is not available in Menu"
<< endl;
            cout << "\t\tPlease press any keyboard to continue program\n" <<
endl;
            system("pause");
        } break;
        } // switch
    } // for loop for products in stock
}
```

Output Screen

```
--------------------------------------------------------------------------------
                    B A Z A R                    A U T H E N T I C A T I O N
--------------------------------------------------------------------------------

                              Authentication

                               1. Sign in
                               2. Sign up
                               3. About
                               0. Exit

                              Your Choice:
```

```
--------------------------------------------------------------------------------
                    B A Z A R                    A B O U T
--------------------------------------------------------------------------------


                        'Bazar' Online Shopping Aplication

                        Team Members: Sahilsher Singh  [9921103131]
                                      Aman Dixit   [9921103133]
                                      Praveen Raj   [9921103121]
                                      Sarthak Chawla   [9921103132]


                        Press any key to go back to Menu
```

```
--------------------------------------------------------------------------------
                    B A Z A R                    R E G I S T R A T I O N
--------------------------------------------------------------------------------

                               Example of Registration:
                              _________________________
                              User Name  : sahil_sandhu
                              Telephone  : 9700000002
                              Login      : sahil_26
                              Password   : Sahil123@
                              _________________________

                             Register User:

                              User Name :
```

```
--------------------------------------------------------------------------------
                                    C A R T
--------------------------------------------------------------------------------

Money will be taken from your 'Telephone Number':
1. OK
Press any key to go back...
```

```
Categories -> Food Products                Cart
------------------------------------------------------------------------------------------
Burger                                     10 (pc)
19000 Rs. for 1 pc

(+)   'Press 1'
(-)   'Press 2'
(0)   'Back'
------------------------------------------------------------------------------------------

Add to Cart:
```

```
------------------------------------------------------------------------------------------
                    B A Z A R                    8. Account Info
------------------------------------------------------------------------------------------

      Categories

      1. Vegetables & Fruits

      2. Water & Beverages

      3. Bread & Bakery Products

      4. Cart and Overall Sums

      0. Go Back

      Your choice:
```

```
------------------------------------------------------------------------------------------
                                    C A R T
------------------------------------------------------------------------------------------


                    1.Potatoes, Weight    2 (kg)    Price: 15780
                    2.Onion, Weight       2 (kg)    Price: 7200
                    3.Pepsi, 500ml        1 (pc)    Price: 3590
                    4.Pizza               1 (pc)    Price: 48000
                    5.Fries               2 (pc)    Price: 30000

                    Overall Price: 209140 Sums

          1. Buy now
          0. Products Menu
```

```
------------------------------------------------------------------------------------------
                    B A Z A R                    O W N E R  M E N U
------------------------------------------------------------------------------------------

 Products List                    Category                  Price      In Stock
------------------------------------------------------------------------------------------
1. Potatoes, Weight               Vegetables & Fruits       7890       10
2. Yellow Carrot, Weight          Vegetables & Fruits       4890       10
3. Onion, Weight                  Vegetables & Fruits       3600       10
4. Water                          Water & Beverages         1590       10
5. Pepsi                          Water & Beverages         3590       10
6. Nector                         Water & Beverages         7550       10
7. Pizza                          Bread & Bakery Products   48000      10
8. Burger                         Bread & Bakery Products   19000      10
9. Potatoe Fries                  Bread & Bakery Products   15000      10

0. Back
Make changes in:
```

References

Book

[1] Pearson programming in c++ by Ashok N. Kamthane

[2] OOP's with c++ by Bala Guruswami

Online:

[3] Class Slides Deep Review

[4] Geeks for Geeks (small topics)