MEDICINE PURCHASE SYSTEM

Object Oriented Programming with C++ Assignment

Submitted by:

Varada Abhay Deshpande

Student, Semester 2,
B. Tech. Computer Science and Engineering,
GSFC University, Vadodara, Gujarat.
(Enrollment ID: 20BT04051)

Guided by: Ms. Rujul Desai

20BT04051

TABLE OF CONTENTS

ABOUT THE SYSTEM 1

Functionalities Available 1
The Customer Portal 1
The Owner Portal 2
Concepts Used 2

SOURCE CODE OF THE SYSTEM 3

SCREENSHOTS OF THE SYSTEM 21

Menus 21
Customer Menu 22
Owner Menu 22
The Customer Portal 23
View Available Medicines 23
Search for the Current Availability of a Medicine 24
View Store Contact Details 25

The Owner Portal 26
Checking Password 26
Deleting Medicine Record 27
Search for Medicine.....28
Addition of Medicine Records 29
Modification of Medicine Records.....30
Incorrect Options Selected in any of the ‘Menu’s 31
Main Menu 31
Customer Menu 32
Owner Menu.....32

ABOUT THE SYSTEM

The "Medicine Purchase System", a C++ based project, has distinct "portals" for the customer and the owner. Various distinct functionalities have been made available through each of these portals.

There is a total of three databases available for the functionalities of the program. These include the following:-

- *Database 1: "AllOrders.txt" has the details of all the fulfilled as well as the unfulfilled orders*
- *Database 2: "PendingOrders.txt" has the details of the pending orders only*
- *Database 3: "AvailableMedicines.txt" holds the details of all the (currently) available medicines.*

FUNCTIONALITIES AVAILABLE

The Customer Portal

A customer has the following facilities:

- *Viewing the list of available medicines*
- *Searching for the availability of a particular medicine*
- *Placing order*

- *Viewing their order details by providing order number*
- *Viewing the contact information of the store.*

The order number here, is generated automatically.

The Owner Portal

Accessing the "Owner Portal" and any of its functionalities requires the entry of correct password. It makes the following functionalities available to the user:

- *Viewing the list of available medicines*
- *Searching for the availability of a particular medicine*
- *Adding or modifying medicine details*
- *Deleting medicine details*
- *Searching order by order number or by customer name*
- *Viewing the details of pending orders*
- *Viewing the details of all the orders (fulfilled + unfulfilled)*
- *Deleting the records of fulfilled orders.*

CONCEPTS USED

The "Medicine Purchase System" uses the following main concepts:-

- *Classes*
- *Objects*
- *Constructors*
- *Functions*
- *Polymorphism (through Overriding)*
- *Inheritance*
- *File Handling*
- *Dynamic Sized Arrays (in the form of vectors)*

SOURCE CODE OF THE SYSTEM

```
#include <iostream>     // For general input-output

#include <conio.h>      // For getch()

#include <string>       // For string operations

#include <fstream>
                // To Read from or Write into an External File (Database)

#include <vector>       // For Dynamic Arrays of int/string types


#include <ctime>
                /* For the time() Function, Which Returns a time_t Value
that Varies All the Time, and hence, the "pseudo-random" Number Varies for
Every Program Call. Also used for getting the current date and time from
the system to store with the details of the placed orders.*/


#include <cstdlib>
                 /* For the srand() Function used to Generate a Random
"Order Number" for Each Order. The srand() Function Sets the Starting
Point for Producing a Series of "pseudo-random" Integers. */


#include <ctype.h>      // To use the isdigit() Function

using namespace std;
```

```
class System
    {
        public:
            void MainMenu();
            void CustomerMenu();
            void OwnerMenu();
            void StoreContactInfo();
            void DisplayMed();
            void SearchMed();
            void PrintBill();
            void SearchOrder();
    };



class Customer : public System
   {
      string CustomerName, CustomerAddress, DoctorName, CustomerContactNo,
 OrderDateTime;
       int CustomerAge, TotalPrice;
       long long  OrderNumber;
       char CustomerGender;
       vector <string> MedicineName;
       vector <int> MedicineQty;
       vector <int> MedicinePrice;
       public:
          Customer()
           {
             CustomerName=CustomerAddress=DoctorName=CustomerContactNo="";
             CustomerName="";
             CustomerAge=TotalPrice=0;
            }
          void GetData();
          static long long GetOrderNumber();
          void PutData();
          void PrintBill();
          static string GetOrderDateTime();
    };


class Owner : public System
   {
      string Password;
      public:
          Owner() { Password = "Admin@XYZ.Medicals"; }
          static bool CheckPwd();
          void DeleteData(int);
          void ViewOrders(int);
          void SearchOrderMenu();
          void SearchOrder();
          bool Add_Modify_MedDetails();
   };
```

```
void System::MainMenu()
   {
      int choice;
      do {
          system("cls");
          cout<<endl<<"********************** WELCOME TO XYZ MEDICALS *
**********************"<<endl
              <<endl<<"----------------------------->> MAIN MENU <<------
------------------------"<<endl;
         cout<<endl<<"Select one of the following (please enter the option
 number)"
              <<endl<<"       1. Customer Portal"
              <<endl<<"       2. Owner Portal"
              <<endl<<"       3. Contact Us"
              <<endl<<"       4. Exit"
              <<endl;

          cin>>choice;
          cout<<endl<<endl;
          switch(choice)
             {
                 case 1: CustomerMenu();
                         break;
                 case 2: if(Owner::CheckPwd())
                             OwnerMenu();
                         break;
                 case 3: StoreContactInfo();
                         break;
                 case 4: exit(0);
                         break;
                 default: cout<<"Invalid Choice! Please Re-Enter.";
             }
        cout<<endl<<"<---- Press Enter to Continue ---->";
        int key=getch();
      }while(choice!=4);
   }



void System::OwnerMenu()
   {
      int choice;
      Owner temp;
      do {
         system("cls");
         cout<<endl<<"****************************** XYZ  MEDICALS ******
************************"<<endl
             <<endl<<"------------------------------>> Owner MENU <<------
------------------------"<<endl;
         cout<<endl<<"Select one of the following (please enter the option
 number)"
```

```
            <<endl<<"       1. View Medicines Inventory"
            <<endl<<"       2. Search for a Medicine in Inventory"
            <<endl<<"       3. Add Medicines in Inventory"
            <<endl<<"       4. Delete Medicines from Inventory"
            <<endl<<"       5. Modify Details of Medicines in Inventory"
            <<endl<<"       6. View All Placed Orders (Complete Database:
 Fulfilled + Unfulfilled Orders)"
            <<endl<<"       7. Search for a Customer's Order in Complete
Database"
            <<endl<<"       8. View Pending Orders"
            <<endl<<"       9. Delete Fulfilled Orders"
            <<endl<<"       10. Back to Main Menu"
            <<endl;

         cin>>choice;
         cout<<endl<<endl;

         switch(choice)
            {
                case 1: DisplayMed();
                        break;

                case 2: temp.SearchMed();
                        break;

                case 3: if(!temp.Add_Modify_MedDetails())
                            cout<<"\nRecord Added Successfully!";
                        else
                            cout<<"\nThere was a pre -
existing record for the given medicine name. That record has been updated
with the new data (item and quantity)";
                        break;

                case 4: temp.DeleteData(1);
                        break;

                case 5: if (temp.Add_Modify_MedDetails())
                            cout<<"\nRecord Modified Successfully!";
                        else
                           cout<<"\nNo record for the given medicine name
 was found in the database. The medicine details have been added as a new
record.";
                           break;

                   case 6: temp.ViewOrders(1);
                           break;

                   case 7: temp.SearchOrderMenu();
                           break;

                   case 8: temp.ViewOrders(2);
```

```
                            break;

                case 9: temp.DeleteData(2);
                        break;

                case 10: MainMenu();
                        break;

                default: cout<<"Invalid Choice! Please Re-Enter.";
             }
          cout<<endl<<"<---- Press Enter to Continue ---->";
          int key=getch();
       }while(choice!=10);
   }


void System::SearchOrder()
   {
      fstream AllOrdersDatabase;
      AllOrdersDatabase.open("AllOrders.txt", ios::in);
      string line, SearchStr, SearchNo;
      bool IsFound = false;
      if(AllOrdersDatabase.fail())
          cout<<"Sorry! File not found";
      else
         {
             cout<<"Please enter the order number: ";
             cin>>SearchNo;
             SearchStr = "Order Number: " + SearchNo;
             while (!AllOrdersDatabase.eof())
                 {
                     getline(AllOrdersDatabase, line);
                     if (line==SearchStr)
                        {
                           while(line!="**********************************
******")
                               {
                                   cout<<line<<endl;
                                   getline(AllOrdersDatabase, line);
                               }
                            IsFound = true;
                            break;
                        }
                 }
             if (!IsFound)
                 cout<<"Sorry! Record not found.";
         }
       AllOrdersDatabase.close();
   }
```

```
void Owner::ViewOrders(int file)
   {
      fstream Db;
      if (file==1)
          Db.open("AllOrders.txt", ios::in);
      else
          Db.open("PendingOrders.txt", ios::in);
      string line;
      if(Db.fail())
          cout<<"Sorry! File not found.";
      else
         {
             while (!Db.eof())
                 {
                     getline(Db, line);
                     cout<<line<<endl;
                 }
         }
      Db.close();
   }


void Owner::SearchOrderMenu()
   {
      int choice;
      do{
         system("cls");
         cout<<"(Please Enter the Option Number):-"
             <<endl<<"\t1. Search Order By Order Number"
             <<endl<<"\t2. Search Order By Customer Name"
             <<endl<<"\t3. Go Back to The Owner Portal"
             <<endl;

         cin>>choice;
         cout<<endl;

         switch(choice)
           {
              case 1: System::SearchOrder();
                      break;
              case 2: Owner::SearchOrder();
                      break;
              case 3: OwnerMenu();
                      break;
              default: cout<<endl<<"Invalid Option! Please Re-Enter.";
           }

         cout<<endl<<"<-- Press any key to continue -->";
         int key=getch();
      } while (choice!=3);
   }
```

```
void Owner::SearchOrder()
   {
      fstream AllOrdersDatabase;
      AllOrdersDatabase.open("AllOrders.txt", ios::in);
      string line, SearchStr, SearchNm;
      bool IsFound = false;
      getchar();
      if(AllOrdersDatabase.fail())
         cout<<"Sorry! File not found.";
      else
        {
          cout<<"Please enter the customer name: ";
          getline(cin, SearchNm);
          SearchStr = "Customer Name: " + SearchNm;
          while (!AllOrdersDatabase.eof())
            {
               getline(AllOrdersDatabase, line);
               if (line==SearchStr)
                {
                   while(line!="****************************************")
                     {
                        cout<<line<<endl;
                        getline(AllOrdersDatabase, line);
                     }
                   IsFound = true;
                   break;
                 }
            }
         if (!IsFound)
            cout<<"Sorry! Record not found.";
       }
     AllOrdersDatabase.close();
   }


void Owner::DeleteData(int file)
   {
      string line, Search, SearchStr;
      bool IsFound=false;
      fstream OldDb, NewDb;
      if (file==1)
          OldDb.open("AvailableMedicines.txt", ios::in);
                                        // Opening Database in "Read" Mode

      else
          OldDb.open("PendingOrders.txt", ios::in);
                                        // Opening Database in "Read" Mode

      if(OldDb.fail())
          cout<<"Sorry! File not found.";
```

```
      else
         {
            getchar();
            if(file==1)
              {
                 cout<<"Please enter the medicine name: ";
                 getline(cin, Search);
                 SearchStr = "Medicine Name: " + Search;
              }
            else
               {
                 cout<<"Please enter the order number: ";
                 cin>>Search;
                 SearchStr = "Order Number: " + Search;
               }
            OldDb.seekg(0, ios::beg);
                        // Set the file pointer to the beginning position

            NewDb.open("TempDb.txt", ios::out);
               // Creating and Opening New Database in "Write/Append" Mode

            while (getline(OldDb, line))
               {
                  if (line==SearchStr)    // Condition helps skip selected
 value from entering new database (data thus deleted)
                     {
                        while(line!="")
                           getline(OldDb, line);
                        IsFound=true;
                     }

                  else if (line!=SearchStr)   // Condition helps write the
 remaining values into new file
                     {
                        while(line!="")
                           {
                              NewDb<<line<<'\n';     // Store line to new
database
                               getline(OldDb, line);
                           }
                        NewDb<<'\n';
                     }
               }
           if (!IsFound)   // if IsFound = false
               cout<<"Sorry! Record not found.";
           else    // "if (IsFound)" Condition
               cout<<"Record Deleted Successfully!";
           NewDb.close();
         }

        OldDb.close();
```

```
     if (file==1)
        {
           remove("AvailableMedicines.txt");   // Deleting Old Database
           rename("TempDb.txt", "AvailableMedicines.txt");
                                                  // Renaming New Database
        }

     else
        {
           remove("PendingOrders.txt");   // Deleting Old Database
           rename("TempDb.txt", "PendingOrders.txt");
                                                  // Renaming New Database
        }
   }



bool Owner:: Add_Modify_MedDetails()
  {
     system("cls");
     string nm, line, SearchStr;
     int qty, p;
     bool IsFound=false;
     cout<<"\nPlease enter the following details:-";
     cout<<"\n\nMedicine Name: ";
     getchar();
     getline(cin, nm);
     cout<<"Quantity: ";
     cin>>qty;
     cout<<"Price: ";
     cin>>p;
     SearchStr = "Medicine Name: " + nm;
     fstream OldDb, NewDb;
     OldDb.open("AvailableMedicines.txt", ios::in);
                                        // Opening Database in "Read" Mode

    OldDb.seekg(0, ios::beg);
                          // Set the file pointer to the beginning position

   NewDb.open("TempDb.txt", ios::out);
               // Creating and Opening New Database in "Write/Append" Mode
    while (getline(OldDb, line))
     {
       if (line==SearchStr)    // Condition helps skip selected value from
 entering new database (data thus deleted)
            {
                while(line!="")
                   getline(OldDb, line);
                IsFound=true;
            }
```

```
      else if (line!=SearchStr)   // Condition helps write the remaining
values into new file
          {
               while(line!="")
                  {
                     NewDb<<line<<'\n';     // Store line to new database
                     getline(OldDb, line);
                   }
               NewDb<<'\n';
          }
       }
    NewDb<<"Medicine Name: "<<nm
         <<"\nAvailable Quantity: "<<qty
         <<"\nPrice per Strip/Bottle: "<<p
         <<"\n**************************************\n\n";
   OldDb.close();
   NewDb.close();
   remove("AvailableMedicines.txt");   // Deleting Old Database
   rename("TempDb.txt", "AvailableMedicines.txt");
                                                 // Renaming New Database
   return IsFound;
  }


bool Owner::CheckPwd()
  {
    Owner temp;
    string Pwd;
    bool CheckPwd;
    char ch;
    system("cls");
    getchar();
    cout<<"Enter Password: ";
    ch = getch();      // Get(input) the First Character/Key


    while(ch != 13)     // Loop until "Enter" key is pressed
     {

       if (ch==8)       // If the key pressed is "Backspace"
         {
           Pwd.pop_back();     // Delete the last-stored character of Pwd
           cout<<'\b';     // To remove a star displayed at the end: what
 the "Backspace" key usually does
         }
```

```
      else     // If the key pressed is not "Backspace"
         {
            Pwd.push_back(ch);
                        // Store (append) the character into the Pwd string
            cout<<'*';      // Display "*" (i.e., the "Encoded Character"
 of entered password)
                    }
        ch = getch();      // Get(input) the next Character/Key
       }


    // Check entered password with the original one
    if(Pwd==temp.Password)      // If they match
      return true;        // return true (Access to Owner Portal Approved)

    else        // If they do not match
      {
        cout<<endl<<"Sorry! Invalid Password. You can't access the Owner
Portal.";
      //Message to let user know that the password they enter is incorrect
       return false;       // return false (Access to Owner Portal Denied)
      }
  }



void System::SearchMed()
  {
     fstream MedsDatabase;
     MedsDatabase.open("AvailableMedicines.txt", ios::in);
     string line, SearchMedName, SearchStr;
     bool IsFound = false;
     getchar();
     if(MedsDatabase.fail())
       cout<<"Sorry! File not found.";
     else
       {
          cout<<"Please enter the medicine name: ";
          getline(cin, SearchMedName);
          SearchStr = "Medicine Name: " + SearchMedName;
          while (!MedsDatabase.eof())
            {
               getline(MedsDatabase, line);
               if (line==SearchStr)
                 {
                   while(line!="****************************************")
                      {
                         cout<<line<<endl;
                         getline(MedsDatabase, line);
                       }
                   IsFound = true;
```

```
                    break;
                }
            }
        if (!IsFound)
            cout<<"Sorry! Record not found.";
     }
    MedsDatabase.close();
  }



void System::DisplayMed()
  {
    system("cls");
    ifstream Inventory;
    string line;
    Inventory.open("AvailableMedicines.txt");
    if(Inventory.fail())
       cout<<"Sorry! File not found";
    else
       while (!Inventory.eof())
          {
             getline(Inventory, line);
             cout<<line<<endl;
          }
    Inventory.close();
  }



void System::CustomerMenu()
  {
    int choice;
    Customer temp;
    do {
        system("cls");
        cout<<endl<<"******************************** XYZ  MEDICALS *****
**************************"<<endl;

     cout<<endl<<"KINDLY NOTE:-"

         <<endl<<"   1) All your orders will be delivered to your address
 within 12 hrs. Payment to be given on successful delivery."

            <<endl<<"   2) Kindly note down your order number for future
references. A bill can be generated only through your order number on plac
ing order, if required. "

            <<endl<<"   3) You will be notified about unavailability/inad
equacy and alternate arrangement of any of the ordered medicines within 2
hrs."
```

```
            <<endl<<"   4) Since this system is under development, we cur
rently do not provide the option to modify/cancel a placed order through o
ur system."<<endl<<"      Any such requests must be made to the shop throu
gh call/mail within 6 hours of order placement."

            <<endl<<endl<<"Kindly bear with us! Thank you!"
            <<endl<<endl;


               cout<<endl<<"------------------------------
>> CUSTOMER MENU <<------------------------------"<<endl;

               cout<<endl<<"Select one of the following (please enter the
 option number)"
                   <<endl<<"      1. View Available Medicines"
                   <<endl<<"      2. Search for the current availability
of a medicine"
                   <<endl<<"      3. Place Order"
                   <<endl<<"      4. View Order Details"
                   <<endl<<"      5. View Store Contact Information"
                   <<endl<<"      6. Back to Main Menu"
                   <<endl;

               cin>>choice;
               cout<<endl<<endl;

               switch(choice)
                 {
                   case 1: DisplayMed();
                           break;
                   case 2: SearchMed();
                           break;
                   case 3: temp.GetData();
                           break;
                   case 4: SearchOrder();
                           break;
                   case 5: StoreContactInfo();
                           break;
                   case 6: MainMenu();
                           break;
                   default: cout<<"Invalid Choice! Please Re-Enter.";
               }

           cout<<endl<<"<---- Press Enter to Continue ---->";
           int key=getch();
       }while(choice!=6);
   }
```

```
void System::StoreContactInfo()
    {
        system("cls");
        cout<<"Contact Us:-"
            <<endl<<"   Phone Number: XXXXXXXXXX"
            <<endl<<"   Email ID: XYZMedicals@example.com"
            <<endl<<"   Website: http://www.XYZMedicals.com"
            <<endl;
    }



void Customer::GetData()
    {
        system("cls");
        int i, mp;
        Customer temp;
        char ch=' ';
        temp.OrderNumber=GetOrderNumber();
        temp.OrderDateTime=GetOrderDateTime();
        cout<<endl<<"Please enter the following details (press enter after
 entering each value):-";
        getchar();
        cout<<endl<<"Patient Name: ";
        getline(cin, temp.CustomerName);
        cout<<"Patient Age (years): ";
        cin>>temp.CustomerAge;
        cout<<"Patient Gender (Please enter M/F/O for Male/Female/Other,
respectively): ";
        bool DataChecker=false;
        while (!DataChecker)
           {
              cin>>temp.CustomerGender;
              if (temp.CustomerGender=='F' || temp.CustomerGender=='f' ||
temp.CustomerGender=='M' || temp.CustomerGender=='m' ||
temp.CustomerGender=='O' || temp.CustomerGender=='o')
                 DataChecker=true;
              else
                 cout<<"Invalid Character. Please Re-Enter: ";
           }

      getchar();
      cout<<"Patient Contact Number: ";
      getline(cin, temp.CustomerContactNo);
      cout<<"Patient Address: ";
      getline(cin, temp.CustomerAddress);
      cout<<"Prescribing Doctor: ";
      getline(cin, temp.DoctorName);
      fstream MedsDb;
      MedsDb.open("AvailableMedicines.txt", ios::in);
```

```
      char choice;
      bool AddMore=true;
      string MedName, SearchStr, line;
      int MedQty;
      bool IsFound;
      cout<<"\n<----- Press any key to enter medicine details ----->";

      while (AddMore)
        {
           getchar();
           cout<<"\nPlease Enter the following Medicine Details:- "<<endl;
           cout<<"Medicine Name: ";
           getline(cin, MedName);
           temp.MedicineName.push_back(MedName);
           cout<<"Medicine Quantity (No. of Strips/Bottles Required): ";
           cin>>MedQty;
           temp.MedicineQty.push_back(MedQty);
           SearchStr = "Medicine Name: " + MedName;
           mp = 0;
           IsFound=false;
           MedsDb.seekg(0, ios::beg);
           while (!MedsDb.eof())
             {
                getline(MedsDb, line);
                   if (line==SearchStr)
                       {
                          getline(MedsDb, line);
                                     // Medicine Quantity Available (skip)
                          getline(MedsDb, line);
                               // Medicine Price (extract price from here)
                         for (i=0; i<line.length(); i++)
                           {
                             ch = line[i];
                                        // Getting each character of line
                              if (isdigit(ch))
                                           // If the character is a digit:

                                  mp = (10*mp) + ch - '0';
            // Calculate (get) the price of the medicine per bottle/strip
                           }

                        temp.MedicinePrice.push_back(mp);        // Store
the price of the Medicine with customer order/bill for further reference
and calculations
                       IsFound = true;
                       break;
                     }
             }
          if (!IsFound)       // If the given medicine is not found in the
database, its price is set to 0
              temp.MedicinePrice.push_back(0);
```

```
        cout<<"Do you want to add more medicines? Please enter y/n -->";
        DataChecker = false;

       while (!DataChecker)
         {
            cin>>choice;
            if (choice=='Y' || choice=='y' || choice=='N' || choice=='n')
               DataChecker = true;
            else
               cout<<"Invalid Character. Please Re-Enter: ";
         }
       if (choice=='N' || choice=='n')
         AddMore = false;
     }

    MedsDb.close();
    fstream OrderDatabase, CompleteOrderDatabase;
    OrderDatabase.open("PendingOrders.txt", ios::out | ios::app);
    CompleteOrderDatabase.open("AllOrders.txt", ios::out | ios::app);

    OrderDatabase<<"Order Number: "<<temp.OrderNumber<<"\n"
                 <<"Order Date and Time: "<<temp.OrderDateTime<<"\n"
                 <<"Customer Name: "<<temp.CustomerName<<"\n"
                 <<"Customer Age: "<<temp.CustomerAge<<" years\n"
                 <<"Customer Gender: "<<temp.CustomerGender<<"\n"
                 <<"Customer Contact No: "<<temp.CustomerContactNo<<"\n"
                 <<"Customer Address: "<<temp.CustomerAddress<<"\n"
                 <<"Doctor Name: "<<temp.DoctorName<<"\n";

    CompleteOrderDatabase<<"Order Number: "<<temp.OrderNumber<<"\n"
                 <<"Order Date and Time: "<<temp.OrderDateTime<<"\n"
                 <<"Customer Name: "<<temp.CustomerName<<"\n"
                 <<"Customer Age: "<<temp.CustomerAge<<" years\n"
                 <<"Customer Gender: "<<temp.CustomerGender<<"\n"
                 <<"Customer Contact No: "<<temp.CustomerContactNo<<"\n"
                 <<"Customer Address: "<<temp.CustomerAddress<<"\n"
                 <<"Doctor Name: "<<temp.DoctorName<<"\n";


    for(i=0; i<temp.MedicineName.size(); i++)
      {
         CompleteOrderDatabase<<"Medicine Name: "<<temp.MedicineName[i]<<
"\t\tQuantity: "<<temp.MedicineQty[i]<<"\t\tPrice per strip/bottle: "<<
MedicinePrice[i]<<"\n";

         OrderDatabase<<"Medicine Name: "<<temp.MedicineName[i]<<
"\t\tQuantity: "<<temp.MedicineQty[i]<<"\t\tPrice per strip/bottle: "<<
MedicinePrice[i]<<"\n";
        temp.TotalPrice += (temp.MedicinePrice[i]*temp.MedicineQty[i]);
       }
```

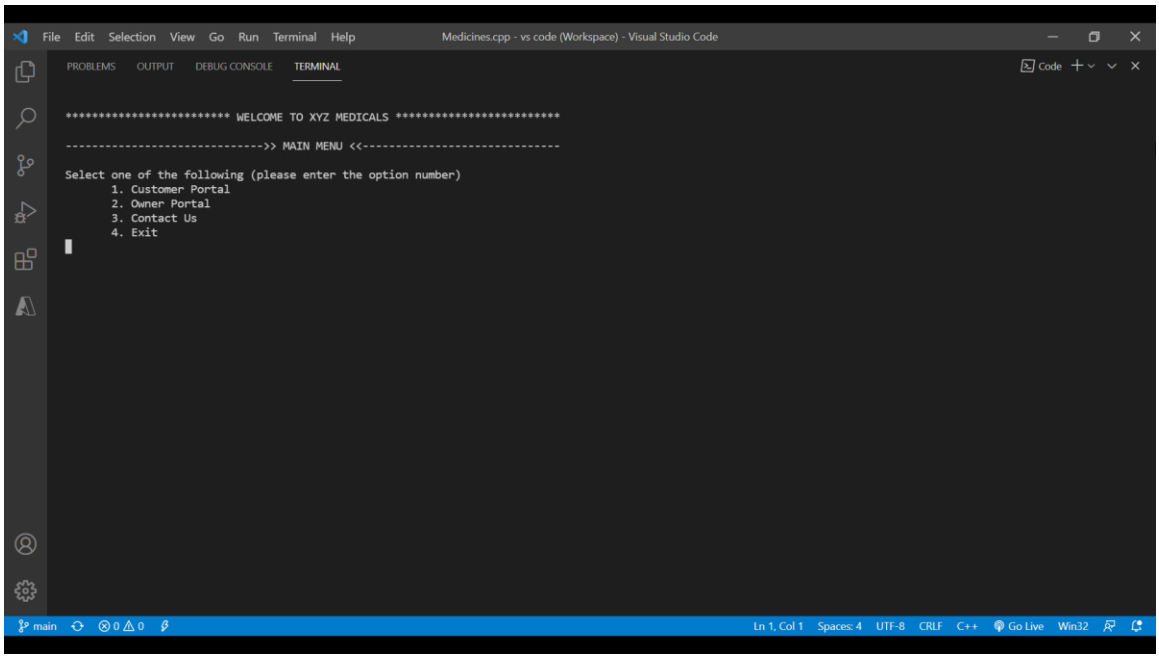
```
       CompleteOrderDatabase<<"Total Bill Amount (exclusive of unavailabe
medicines): "<<temp.TotalPrice;

      OrderDatabase<<"Total Bill Amount (exclusive of unavailabe medicines
): "<<temp.TotalPrice;

      CompleteOrderDatabase<<"\n**************************************\n
\n";

      OrderDatabase<<"\n**************************************\n\n";
      OrderDatabase.close();
      CompleteOrderDatabase.close();
    cout<<endl<<"<---- Press Enter to View the Details you Entered ---->";
     temp.PutData();
   }



long long Customer::GetOrderNumber()
    {
        srand(time(0));
        return rand();
    }


string Customer::GetOrderDateTime()
    {
        time_t now = time(0);
        char* date_time = ctime(&now);
        return date_time;
    }



void Customer::PutData()
   {
      cout<<endl<<"**************************************"
          <<endl<<"ORDER NUMBER: "<<OrderNumber
          <<endl<<"Order Date and Time: "<<OrderDateTime
          <<endl<<"Patient Name: "<<CustomerName
          <<endl<<"Patient Age: "<<CustomerAge
          <<endl<<"Patient Gender: ";

      if (CustomerGender=='F' || CustomerGender=='f')
          cout<<"Female";
      else if (CustomerGender=='M' || CustomerGender=='m')
          cout<<"Male";
      else if (CustomerGender=='O' || CustomerGender=='o')
          cout<<"Other";
```

```
      cout<<endl<<"Patient Contact Number: "<<CustomerContactNo
          <<endl<<"Patient Address: "<<CustomerAddress
          <<endl<<"Prescribing Doctor: "<<DoctorName;

      for(int i=0; i<MedicineName.size(); i++)
          cout<<endl<<"Medicine Name: "<<MedicineName[i]<<"\t\tQuantity: "
<<MedicineQty[i]<<"\t\tPrice per strip/bottle: "<<MedicinePrice[i];

       cout<<endl<<"Total Bill Amount (Exclusive of currently unavailable
medicines): Rs. "<<TotalPrice;
}


int main()
    {
        System obj;
        obj.MainMenu();
        return 0;
    }
```

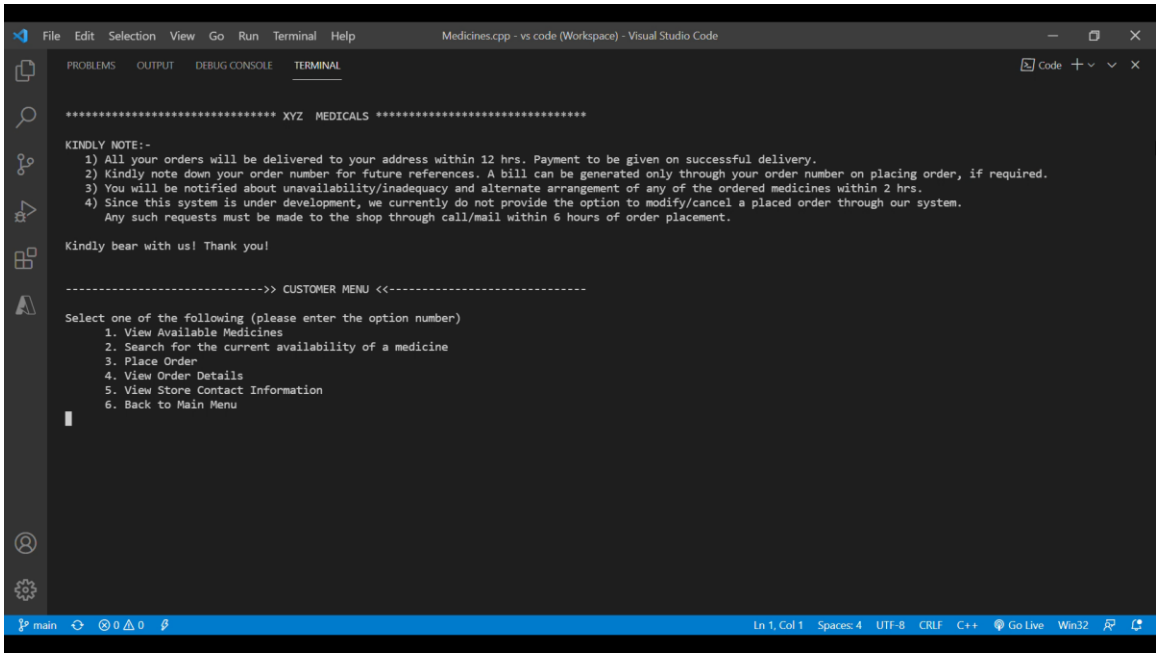
SCREENSHOTS OF THE SYSTEM

MENUS

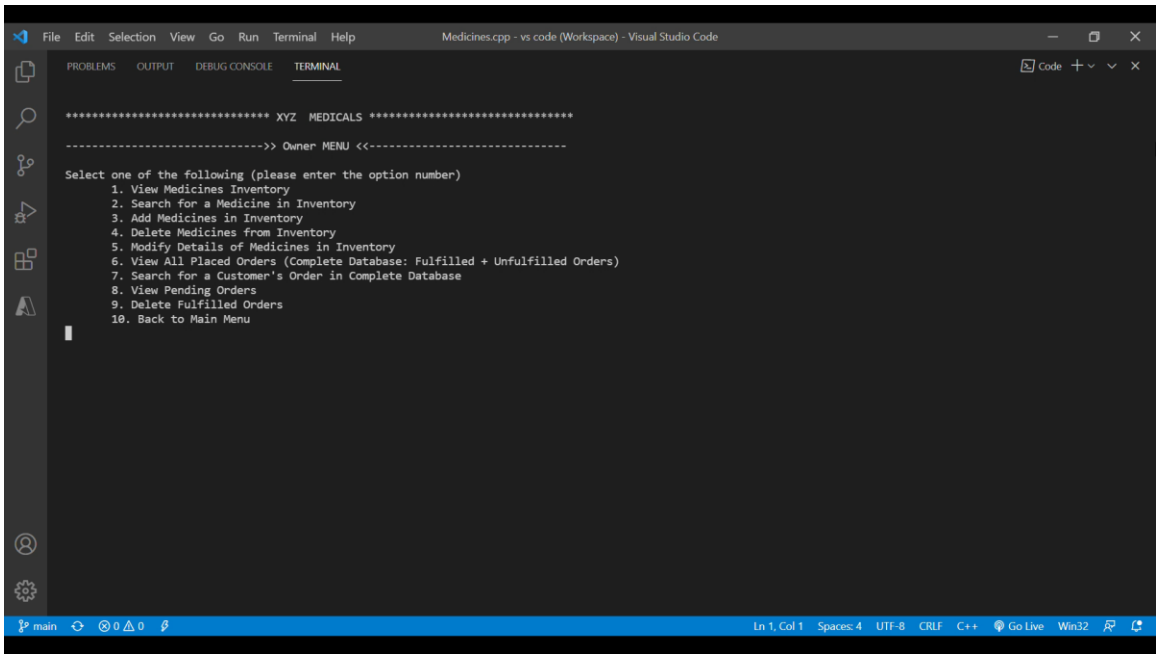
Main Menu



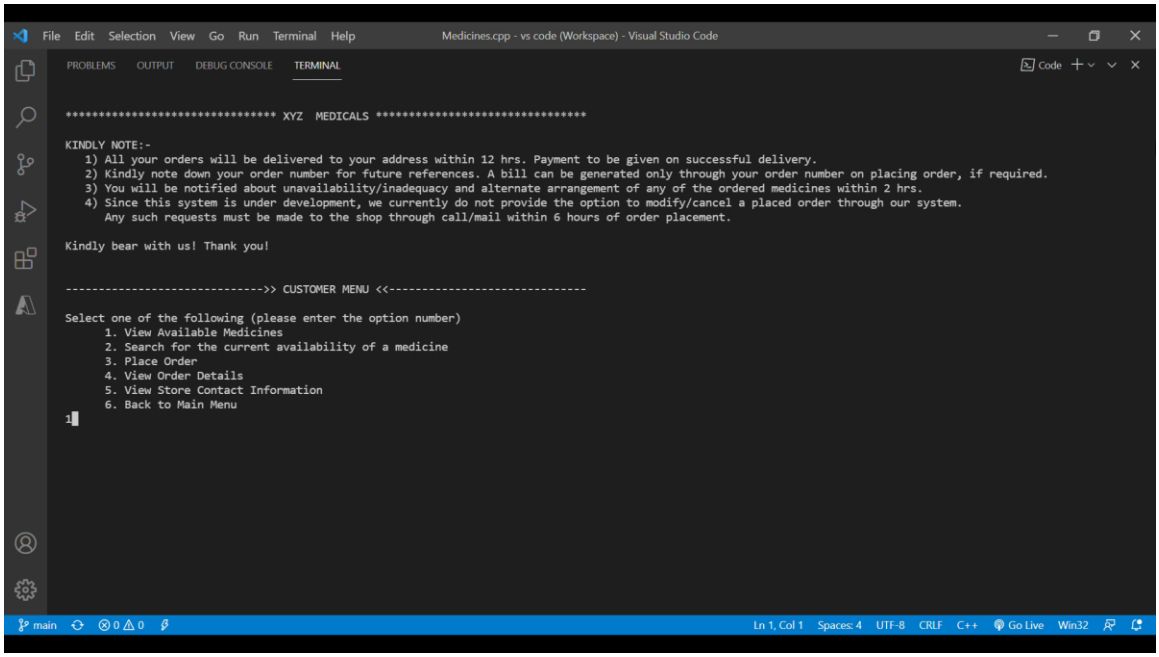
Customer Menu



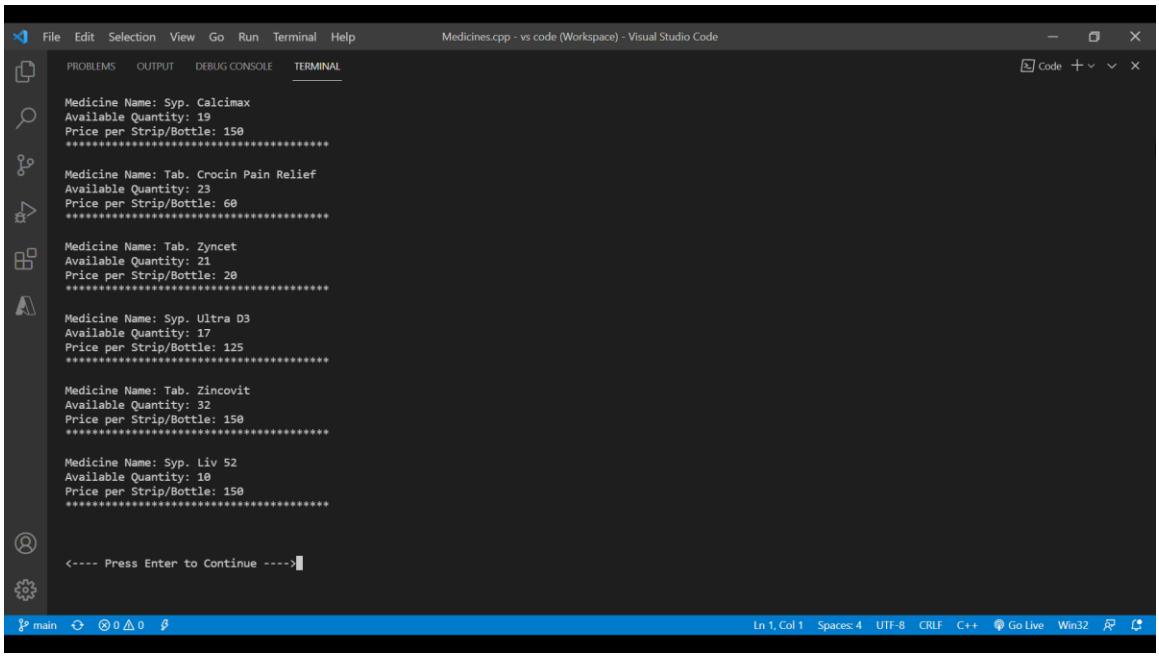
Owner Menu



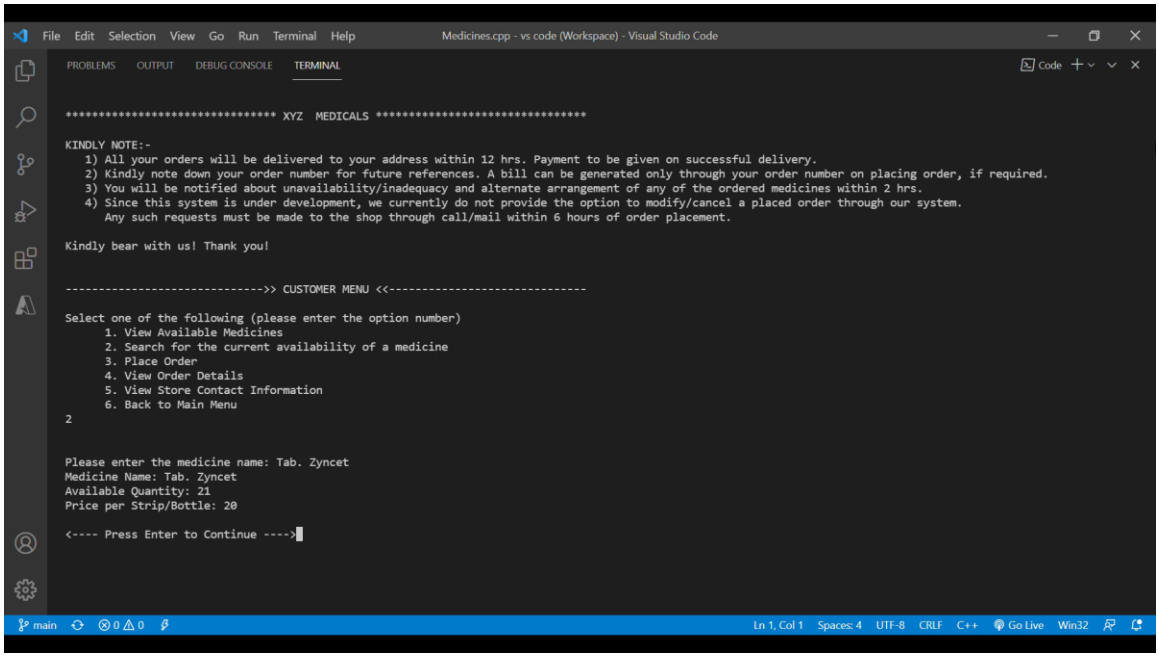
THE CUSTOMER PORTAL

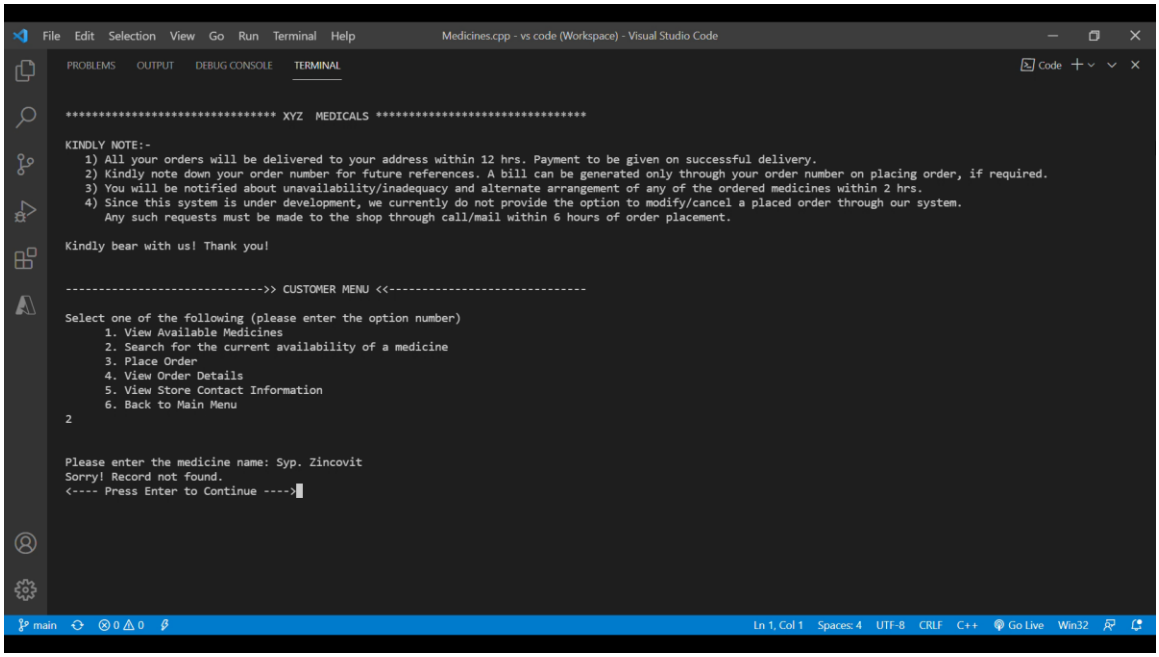


View Available Medicines

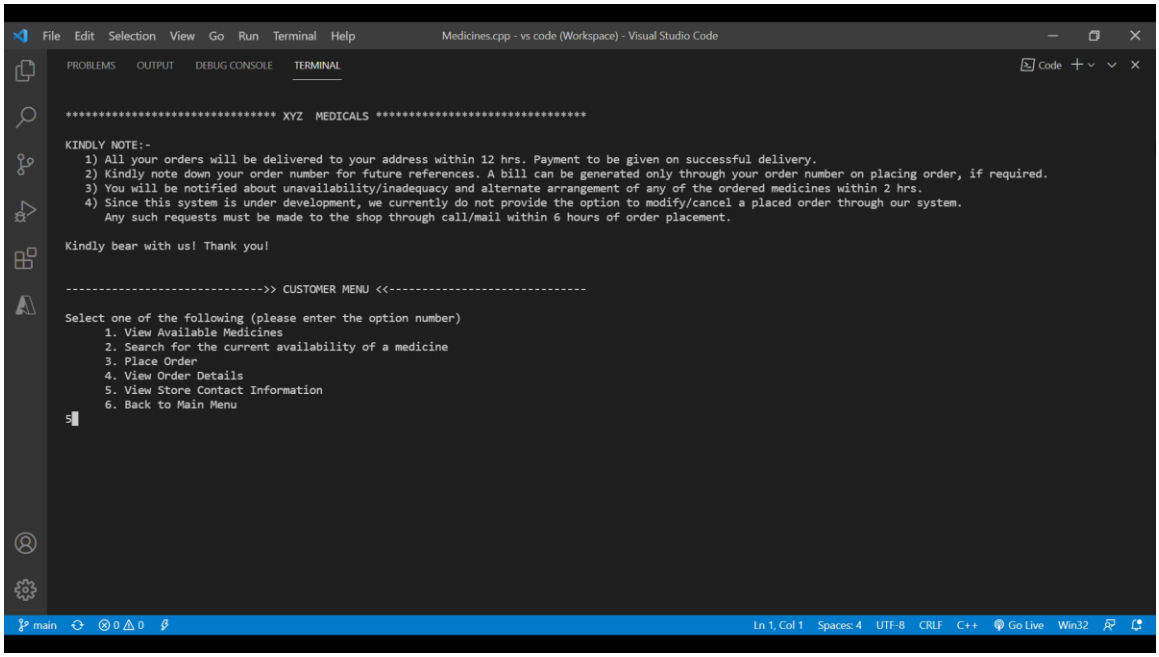


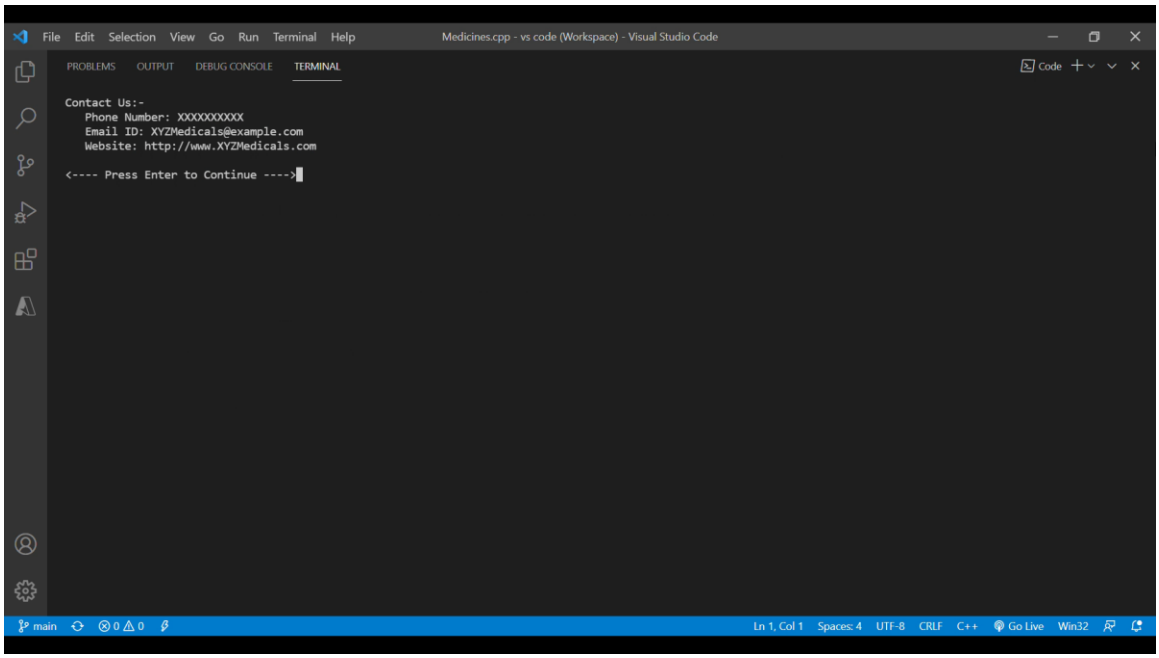
Search for the Current Availability of a Medicine





View Store Contact Details

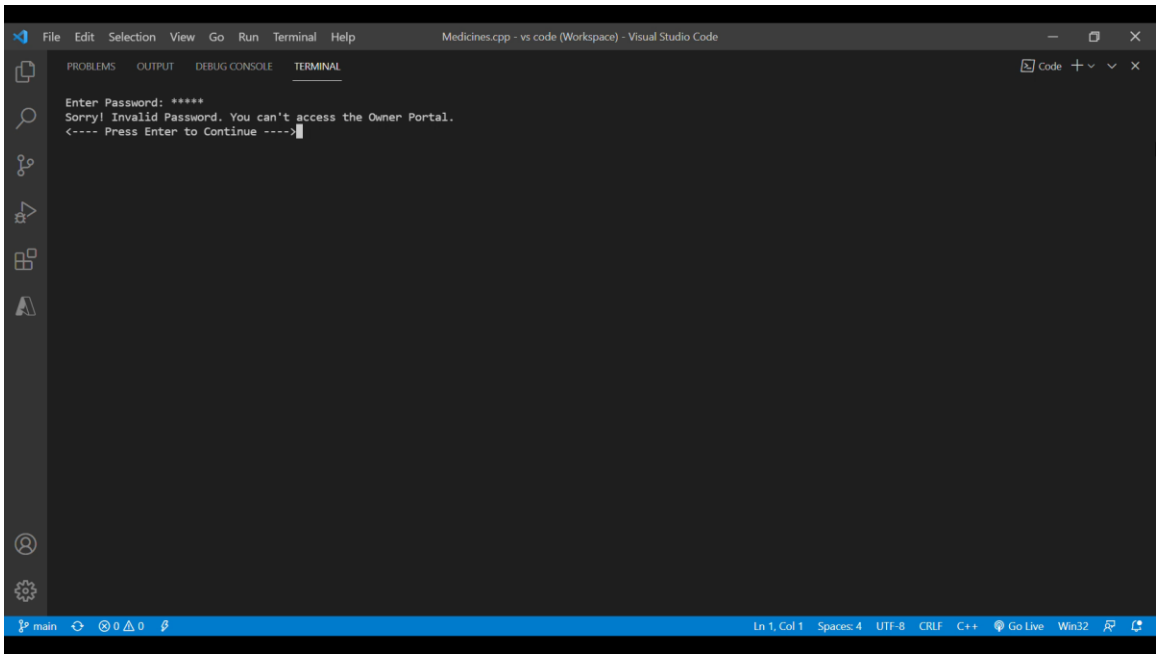




THE OWNER PORTAL

Checking Password

Incorrect Password Entered:

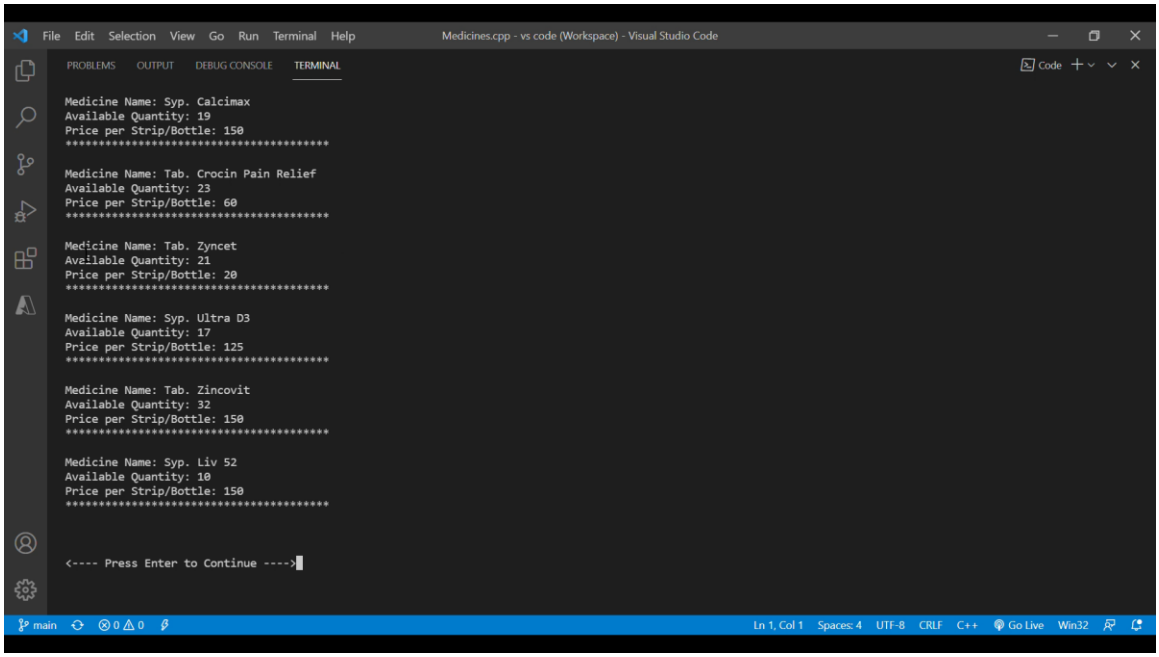


… And back to Main Menu

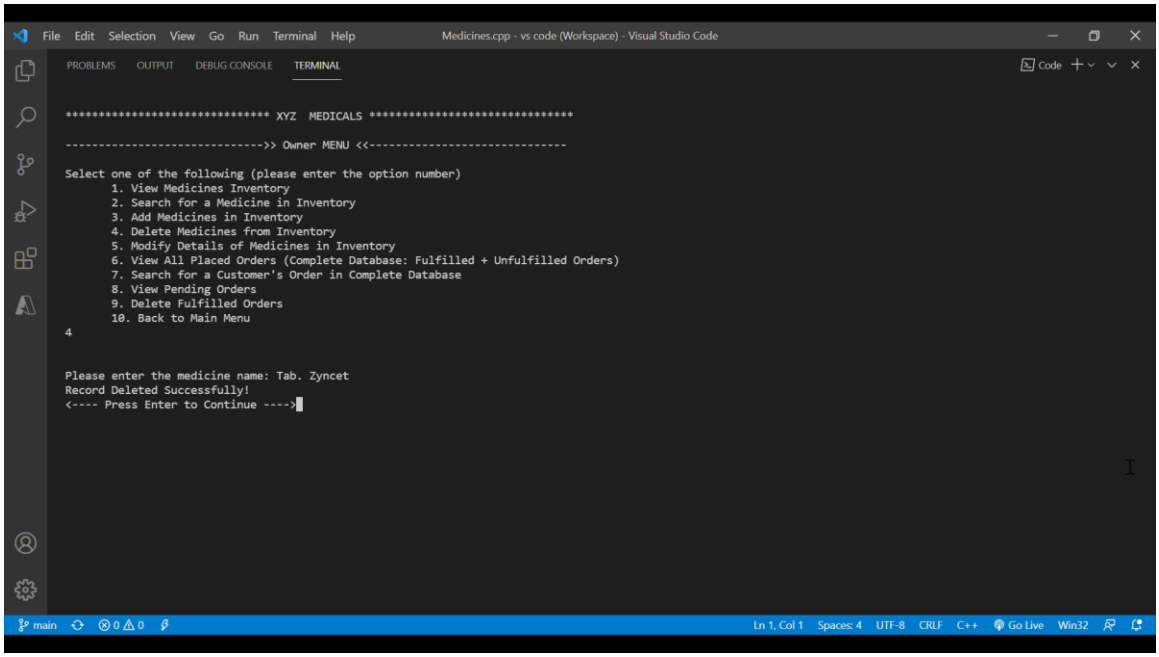
Correct Password Entered: Go to owner portal!

Deleting Medicine Record

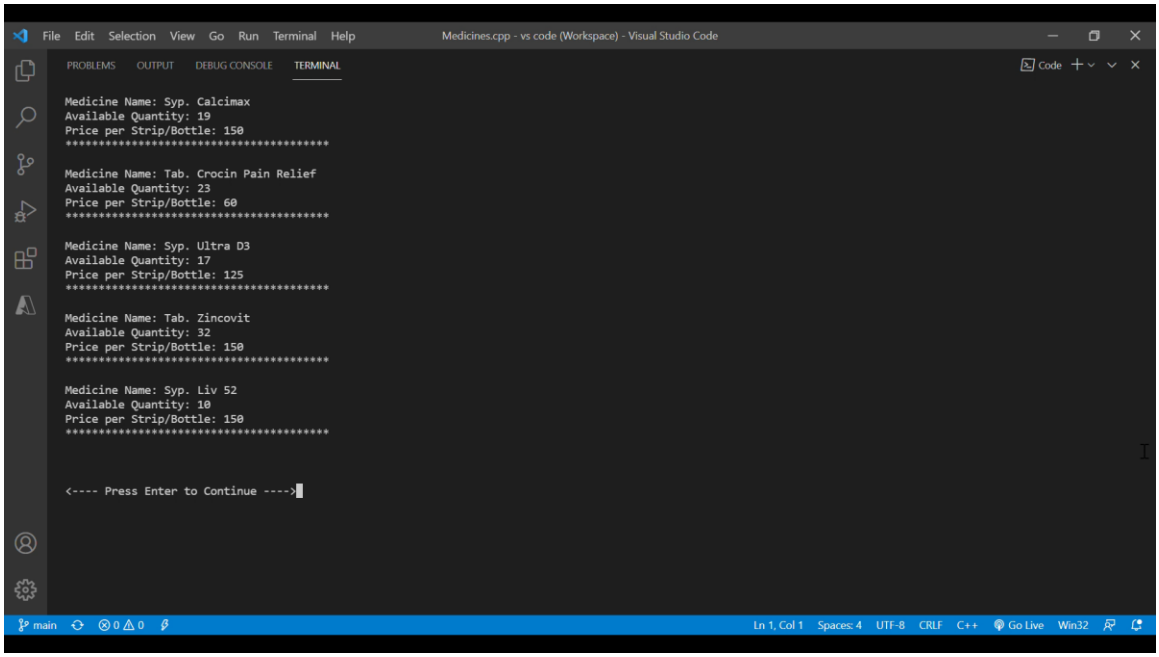
File Before Deletion:



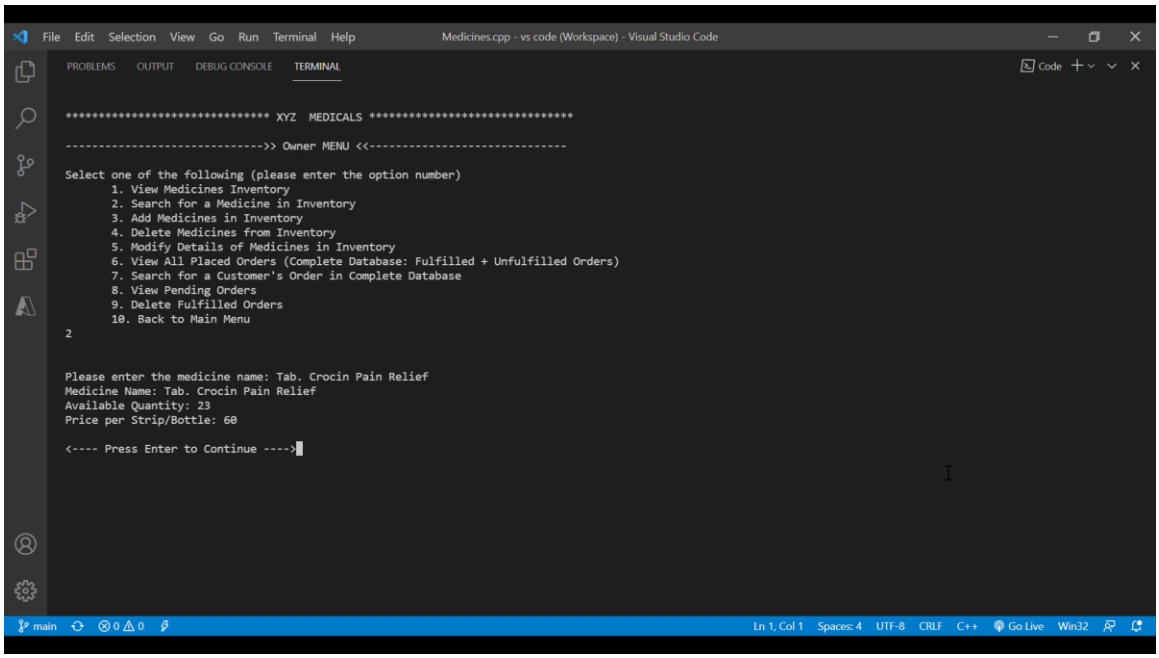
Deleting Record:

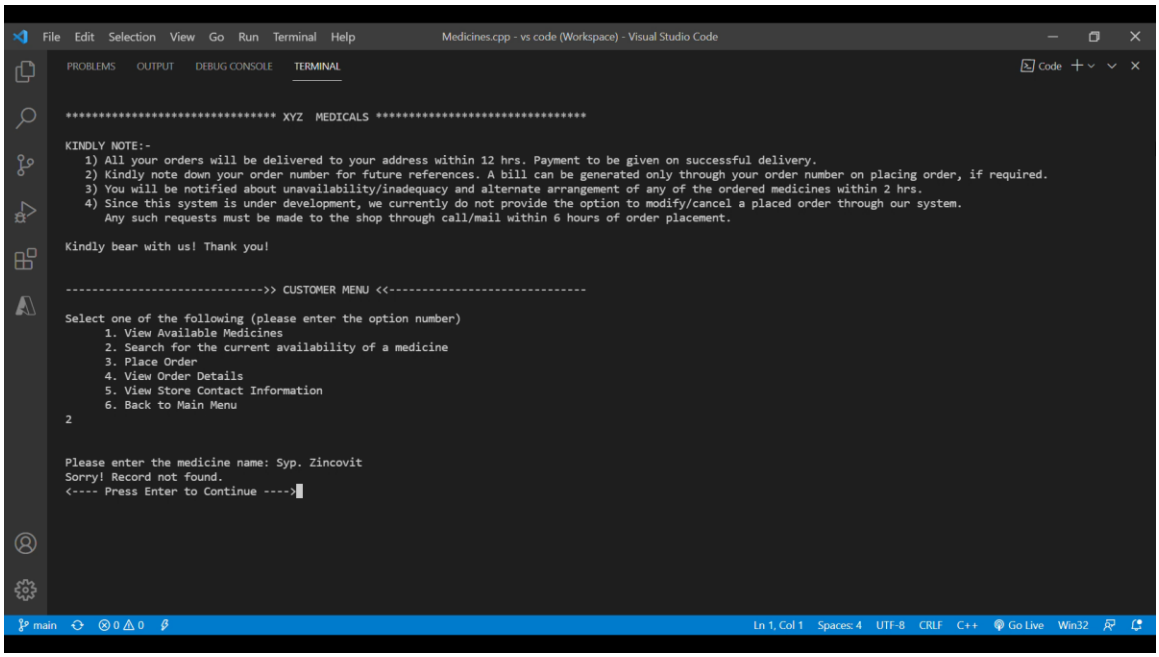


File After Deletion



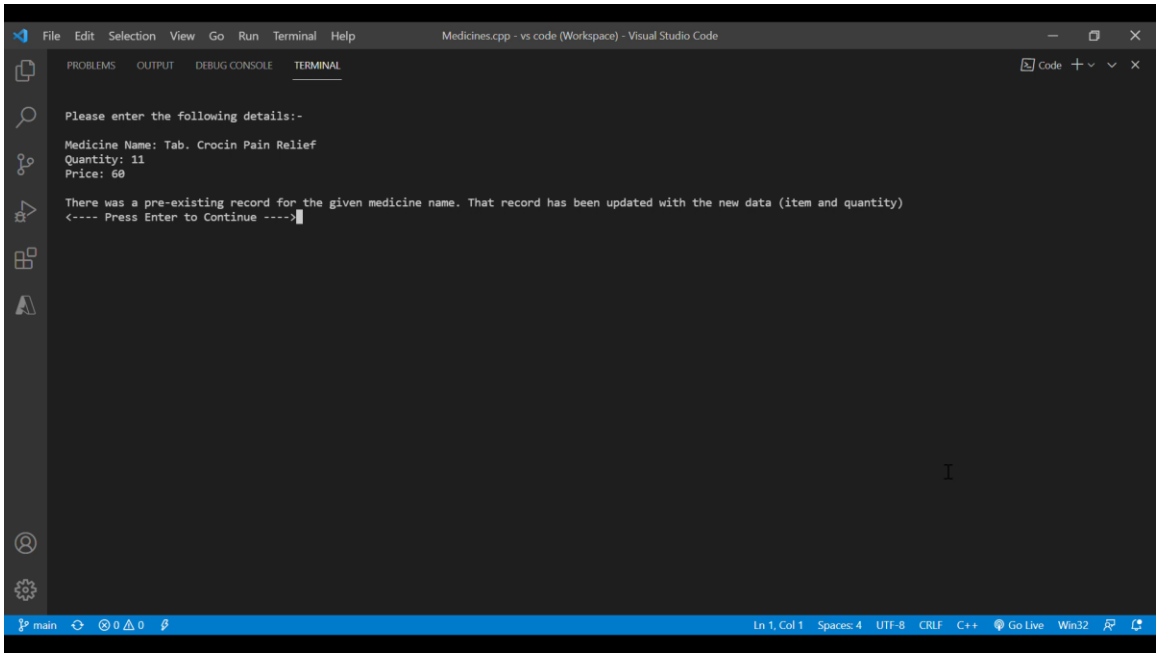
Search for Medicine



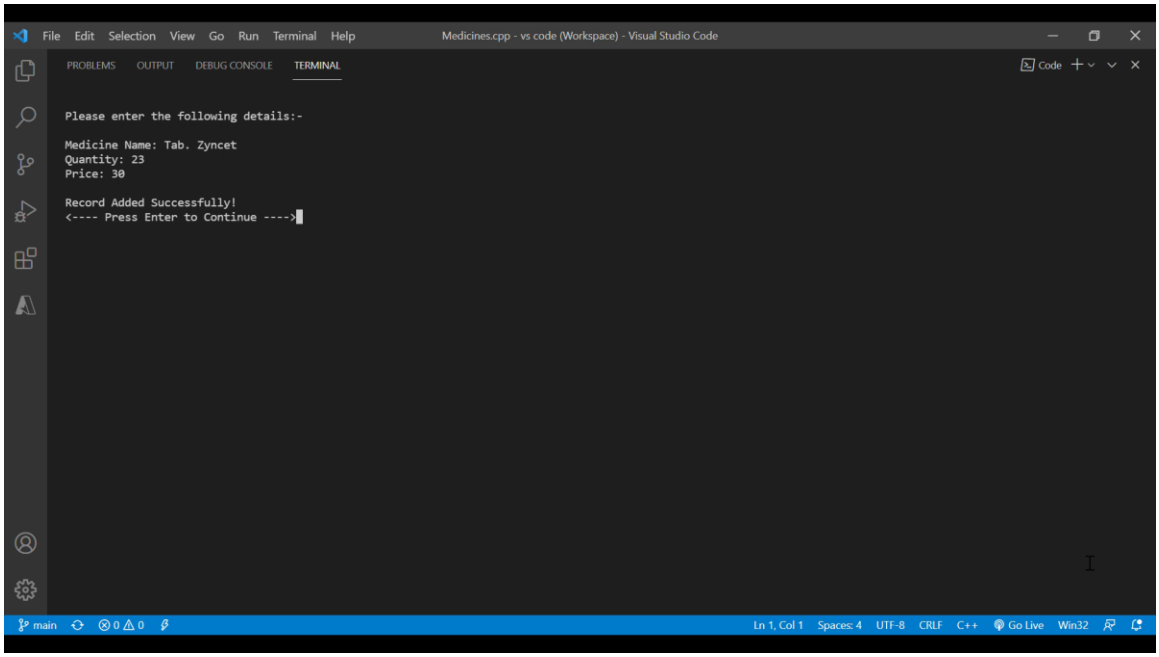


Addition of Medicine Records

If the medicine name exists in the database:

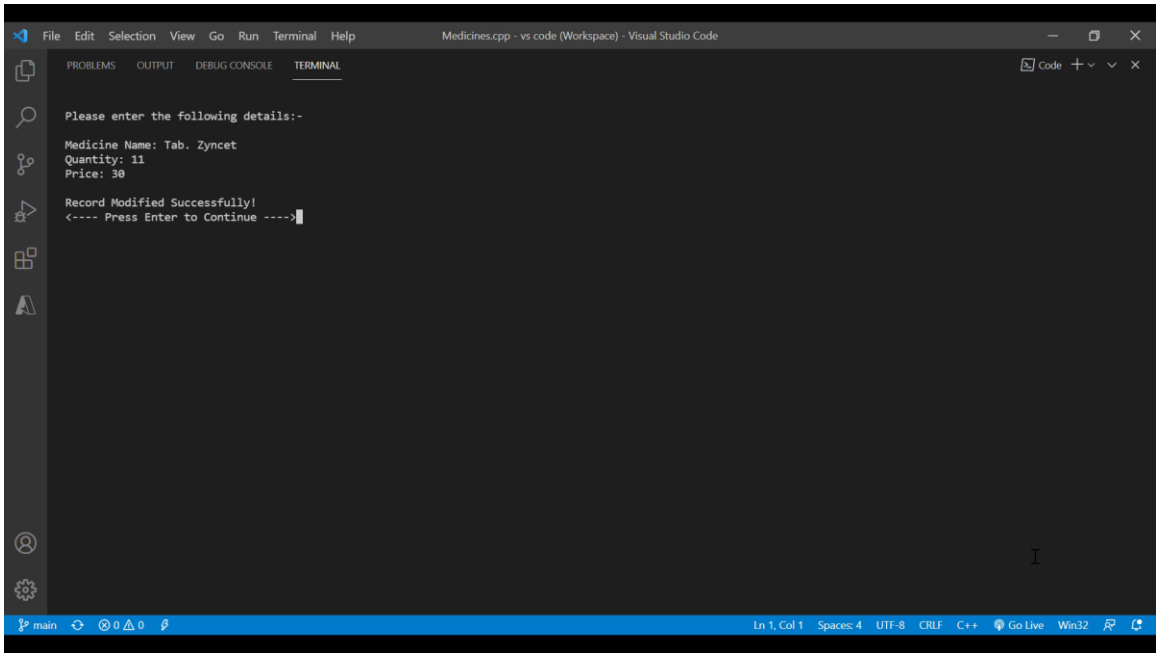


If the medicine name is not present in the database:

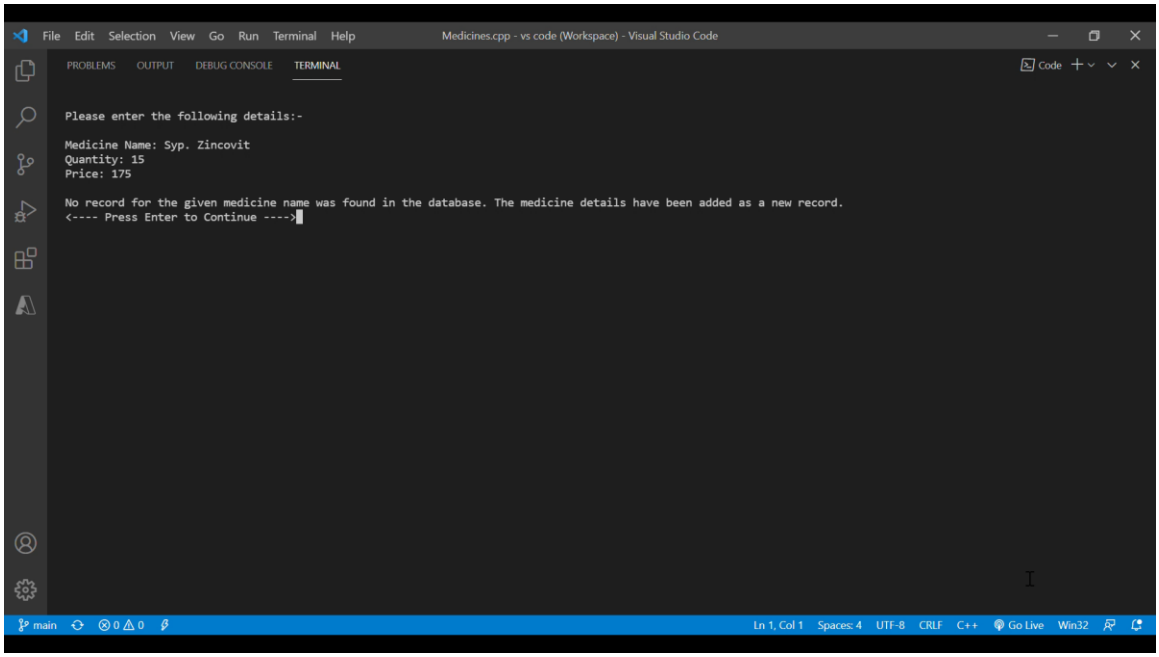


Modification of Medicine Records

If the medicine name exists in the database

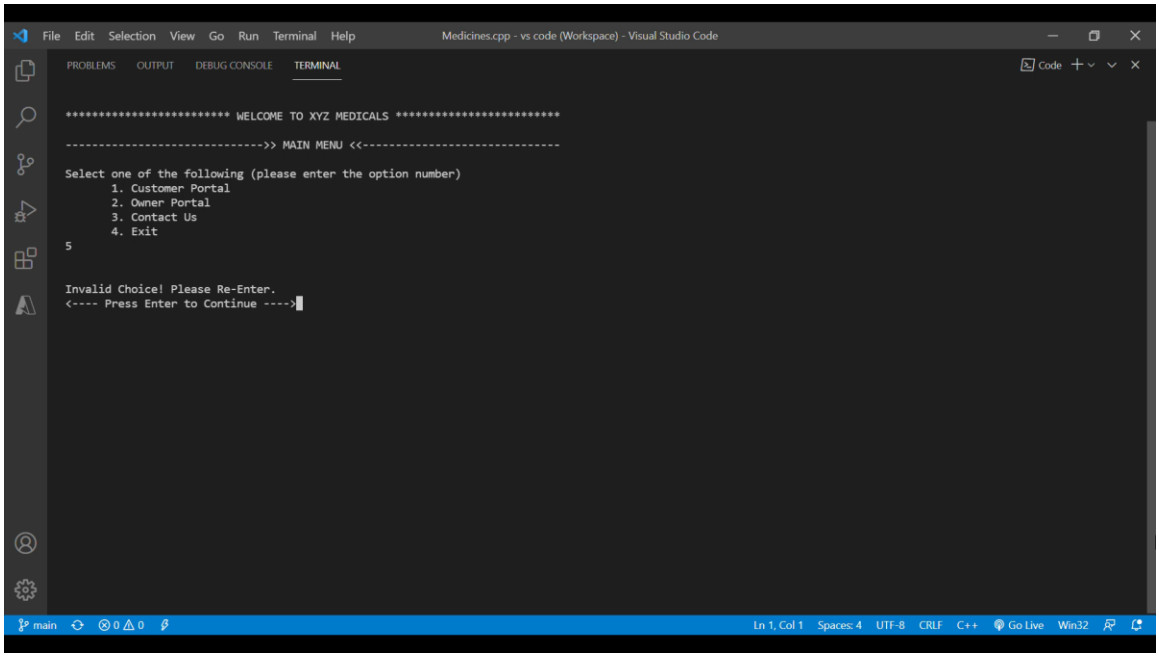


If the medicine name is not present in the database:

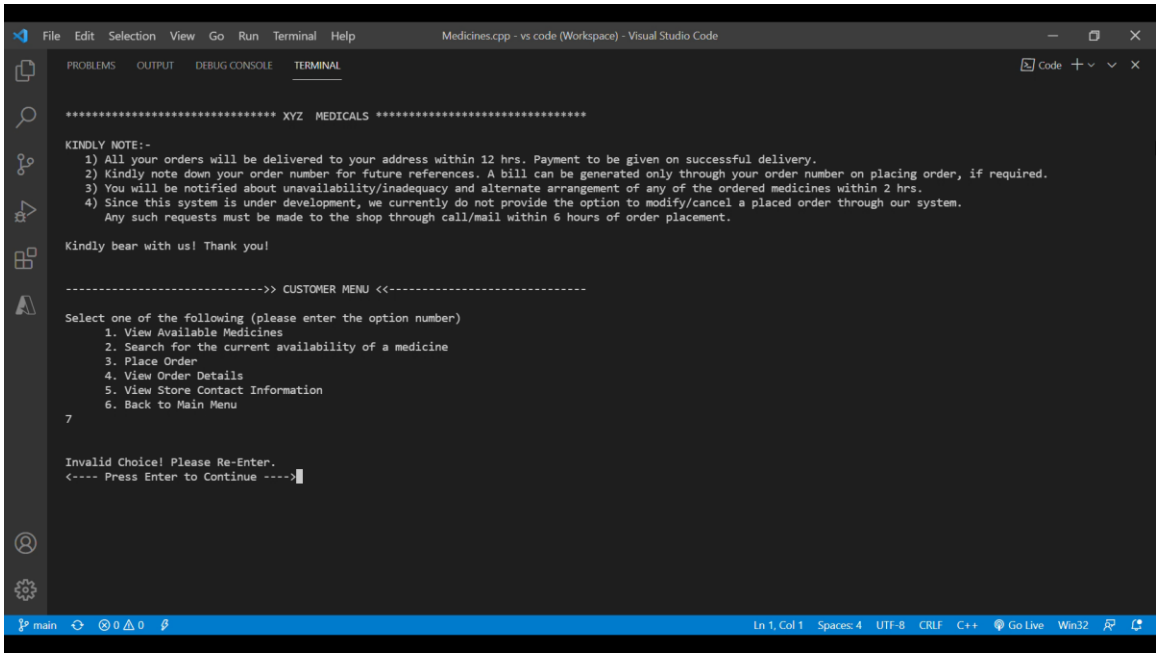
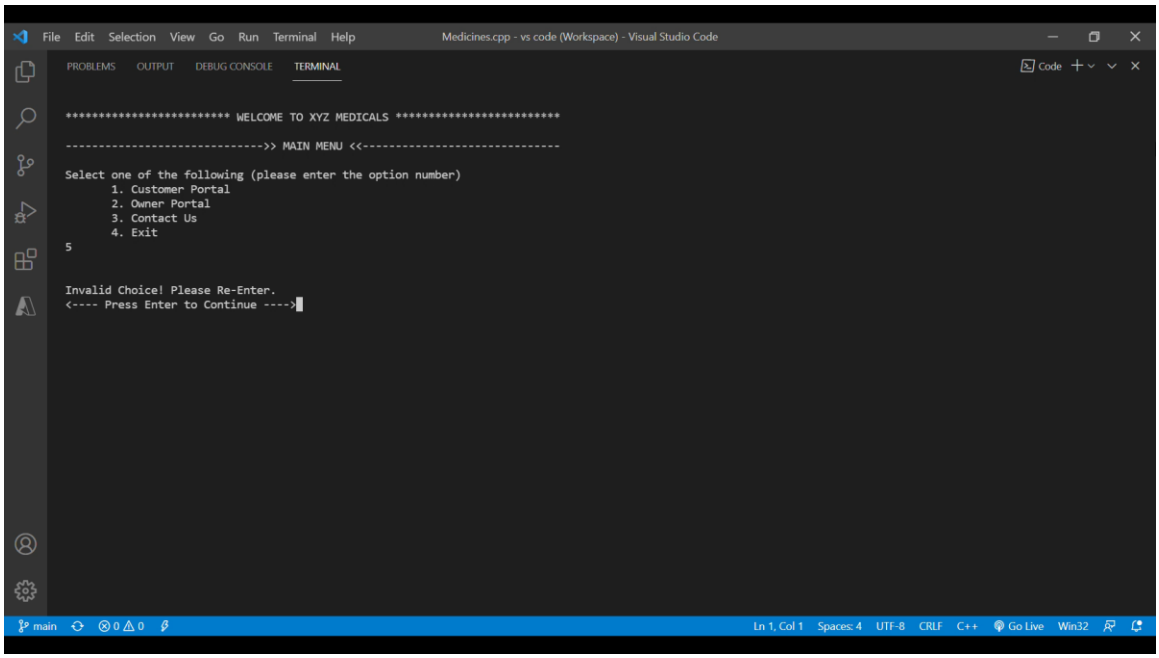


INCORRECT OPTIONS SELECTED IN ANY OF THE 'MENU'S

Main Menu



Customer Menu



Owner Menu