

*MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI*

**GOVERNMENT POLYTECHNIC OSMANABAD**

**CERTIFICATE**

This is to certify that the micro project entitled-

**Implementing** **Hotel Management System using**

**concepts of OOP**

Submitted by :- **Kshiragar Yogesh Dattatraya**

Roll no:- **29** in third semester of diploma in computer engineering has completed micro project satisfactorily in the course **Object Oriented Programming Using C++ (22316)** academic year 2022-2023 as prescribed in the curriculum.

Place: Osmanabad Enrolment No-2101180277

Date: / /2022 Exam Seat No- 331734

Subject Teacher Head of the Department principle

Seal of

Institution

*MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI*



**GOVERNMENT POLYTECHNIC OSMANABAD**

**CERTIFICATE**

Micro project title

**Implementing** **Hotel Management System Using concepts of OOP**

Submitted by

|  |  |  |  |
| --- | --- | --- | --- |
| Roll No | Name of student | Enrollment no | Seat no |
| 29 | KSHIRSAGAR YOGESH DATTATRAYA | 2101180277 | 331734 |

**UNDER THE GUIDANCE MR A.D. AMBURE SIR**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **TITLE** | **PAGE NO** |
| 1 | Acknowledgement | 4 |
| 2 | Introduction | 5 |
| 3 | Rationale | 6 |
| 4 | Course Outcomes | 6 |
| 5 | AIM | 7 |
| 6 | Prroject Outcomes | 7 |
| 7 | Review Of Literature | 7 |
| 8 | Concepts of OOP | 8 |
| 9 | File Handeling | 11 |
| 10 | Algorithm | 14 |
| 11 | Flowchart | 15 |
| 12 | Code | 17 |
| 13 | Output | 37 |
| 14 | Resources Used | 43 |
| 15 | Reference And Outcomes | 44 |

**ACKNOWLADGEMENT**

I am grateful to Almighty God for giving me the strength, knowledge and understanding to complete this project. His love has been more than sufficient to keep and sustain me.

My profound gratitude goes to my wonderful supervisor, Mr A.D. AMBURE sir for him invaluable support, patience, time and guidance in seeing me to the completion of this research work. Also, my gratitude goes to my head of department Mr. P.J. Bansode sir who patiently saw me to the completion of this research work.

I extend gratitude and appreciation to my lecturers Mr A.D.AMBURE sir in department who have taught me at one point or the other. May God continue to bless, protect and guide you all.

I also wish to acknowledge the great support of my parents, siblings who have been a source of inspiration towards my academic pursuit. God bless you all.

I will not cease to acknowledge the support of my friends: Samarth , Purushottam ,Om, and Sandesh . God bless you all

Kshirsagar Yogesh

Computer eng

**INTRODUCTION**

In this project we are going to know more about the concept and the OOP, we are going to implement the real life problem namely hotel management system. We implement it using the c++ because of, the C++ is an extension to C language and was developed by Bjarne stroustrup at bell labs. C++ is an intermediate level language, as it comprises a confirmation of both high level and low level language features. C++ is a statically typed, free form, multiparadigm, compiled general-purpose language. C++ is an Object Oriented Programming language but is not purely Object Oriented. Its features like Friend and Virtual, violate some of the very important OOPS features, rendering this language unworthy of being called completely Object Oriented. It's a middle level language.

**RATIONALE**

**Key Question**

What is OOP concepts ?, how to implement it using c++ language?

**Project Rationale**

The implementation of the hotel management can be done by the programming languages which supports the concept of OOP. In this project we implement the hotel management with the c++ language.

This is maybe a proper implementation of the hotel management Because in this project I use the best implementation of OOP.

In future this is become easy to implement the any real life problem

method because of the artificial development.

My project questions that I am answering are "How to implement the real life problem?" and "What is the concept of OOP?" these questions helps me to completing my project

**COURSES OUTCOMES**

This course have following outcomes

1. Describe OOPs concepts

2. Use functions and pointers in your C++ program

3. Understand tokens, expressions, and control structures

4. Explain arrays and strings and create programs using them

5. Describe and use constructors and destructors

6. Understand and bank management system

7. How to use OPEN CV for security 8. How to implement the data structure in c++

**AIM**

Implementation of the hotel management system using the concept of OOP and getting more information about the OOP.

**PROJECT OUTCOMES**

We can implement or model the real life problem like hotel

management etc by using the concept of OOP. We got

the proper information and knowledge about the concepts of OOP

**REVIEW OF LITERATURE**

We implement this project because of getting more information about the concept of OOP and model or demonstrate our knowledge about the OOP. I collect the information of OOP concept from Books, web pages and more. From books I know more about the data structure from websites I get whole information about OOP concept.

* **Concepts of oop**

1] CLASS :

Class is used to define data type in which data type & function are defined Class is collection of similar type of object .Class acts as template or pattern for objects

EX : flower , car , furniture, are example of the class

2] OBJECT :

Object is a instance of a class , each object has its own data which is different from another object but object of some class perform similar function .

Object is any real world entity like chair, pen, board, college bag etc.

Each object can be logical are physical thing

Ex. Dog is a object because it has colour , name breed etc and it has function like bark , bite, eating, walking etc .

3] DATA ABSTRACTION :

Data abstraction is nothing but hiding the details of operation from user . The data and operation in program are hidden from user where user does not aware about how operation are carried out .

EX: suppose circle is a class where radius is defined then user does not other about object creation and memory allocation for it .

4] DATA ENCAPSULATION :

Data encapsulation means protecting the data from outside function.

In oop encapsulation is achieved by wrappering up data and functions together in a class

EX. A class consist property and function together in it

5] INHERITANCE :

Inheritance is mechanism to create a new class . Due to this new class has ability to reuse the property and function of old class .

The main benifit of inheritance is reusability

6] POLYMORPHISM :

It is ability to take more than one from in programming language .

Poly means many and morph means different form .

In oop function overloading , operator overloading and virtual functions are used in polymorphism.

EX. (-)minus operator is used for substraction when written as a-b but it will act as negative sign operator when written as -x.

7] DYNAMIC BINDING :

The function call is unware about to bind it's defination until runtime is known as dynamic binding .

This concept is mainly used with combination of inheritance and polymorphism

Dynamic binding is also known as late binding .

8] MESSAGE PASSING :

When two object want to communicate with each other it uses message. Passing where one object send parameter to another object through function .

**FILE HANDLING**

File handling in C++ is a mechanism to store the output of a program in a file and help perform various operations on it. File handling is used for store a data permanently in computer. Using file handling we can store our data in secondary memory (Hard disk).

STREAMS IN C++

We give input to the executing program and the execution program gives back the output. The sequence of bytes given as input to the executing program and the sequence of bytes that comes as output from the executing program are called stream. In other words, streams are nothing but the flow of data in a sequence.

FILE STREAM CLASSES

The I/O (input/output) system of C++ contains a set of classes which define the file handling methods. These include ifstream, ofstream and fstream classes. These classes are derived from fstream and from the corresponding iostream class. These classes, designed to manage the disk files, are declared in fstream and therefore we must include this file in any program that uses files.

1.IOS

a) ios stands for input output stream.

b) This class contains the necessary facilities that are used by all the other derived classes for input and output operations.

2.istream

a) istream stands for input stream.

b) This class is derived from the class 'ios'. The extraction operator(>>) is overloaded in this class to handle input streams from files to the program execution.

c) This class declares input functions such as get(), getlline(), and read().

3. ifstream

a) This class provides input operations.

b) It contains open() function with default input mode. c) Inherits the functions get(), getline(), read(), seekg(), and tellg() functions from istream.

4. ofstream

a) This class provides output operations.

b) It contains open() function with default output mode.

c) Inherits the functions put(), write(), seekp() and tellp() functions

from the ostream.

5. fstream

a) This class provides support for simultaneous input and output

operations.

b) Inherits all the functions from istream and ostream classes through iostream. In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream available in fstream headerfile.

**ALGORITHM**

1. Start

2. Create the struct head , head1 and allocate the memory from heap to it

3. Display “welcome to Hotel management system”

4. Display the choices

1. Manager

2.Customer

3.Exit

5. Case : 1 then Display the options

1.New Customer

2.Display ID

3.Remove Customer

4.Display Information Of Customer

5.Display File

6.Search In File

7.Back

6.Case : 2 then Display the options

1.Display Free Rooms

2.Services

3.Back

7.Case : 3 then exit the program

8.Do these steps until while(true)

9.Stop

**FLOWCHART**

START

Declare All Required Variables

Display Options 1.Manager2.Customer 3.Exit

F

Switch(ch)

Case1

T

Show options to user

1.New Customer 2.Display ID 3.Remove Customer 4.Display Information Of Customer 5.Display File 6.Search In File 7.Back

Switch(ch)

T

Add New Customer

Case 1

F

T

Display ID of all customers

Case 2

F

T

Remove Customer

Case 3

A

T

Display Information Of Customer

Case 4

F

T

Display File

Case 5

F

T

Search in file

Case 6

F

If(ch == 7)

F

Case 2

T

Case 3

Display options to user

1.display free rooms

2 services

3.back

T

stop

Switch(ch)

T

Display free rooms

Case 1

F

T

services

Case 2

T

F

Case 3

**CODE**

#include <iostream>

#include <string.h>

#include <conio.h>

#include <fstream>

using namespace std;

fstream file;

class manager

{

    string name;

    string password;

public:

    void setpassword()

    {

        cout << "\t\tENTER YOUR NAME" << endl

             << "\t\t";

        cin >> name;

        cout << "\t\tENTER YOUR PASSWORD" << endl

             << "\t\t";

        cin >> password;

    }

    int checkpassword(string s)

    {

        if (password.compare(s) == 0)

        {

            return 1;

        }

        else

        {

            return 0;

        }

    }

};

class person

{

protected:

    string name;

    long long ph\_no;

    string adress;

public:

    person()

    {

        cout << "\t\tEnter Your Name" << endl

             << "\t\t";

        cin >> name;

        cout << "\t\tEnter Your Phone Number" << endl

             << "\t\t";

        cin >> ph\_no;

        cout << "\t\tEnter Your Address " << endl

             << "\t\t";

        cin >> adress;

    }

    string return\_name()

    {

        return name;

    }

    long long returnph()

    {

        return ph\_no;

    }

    string return\_adress()

    {

        return adress;

    }

    void putdata()

    {

        cout << "\t\tNAME :- " << name << endl;

        cout << "\t\tPHONE NO :- " << ph\_no << endl;

        cout << "\t\tADDRESS :- " << adress << endl;

    }

};

class room

{

protected:

    int no\_of\_days;

    static int no\_of\_rooms;

    static int room\_count;

    string Ac\_NonAc;

public:

    int room\_no;

    room()

    {

        if (no\_of\_rooms > 0)

        {

            no\_of\_rooms--;

            cout << "\t\tEnter No of days You live in hotel" << endl

                 << "\t\t";

            cin >> no\_of\_days;

            cout << "\t\tEnter Ac/Non Ac" << endl

                 << "\t\t";

            cin >> Ac\_NonAc;

        }

    }

    int returndays()

    {

        return no\_of\_days;

    }

    string returnac()

    {

        return Ac\_NonAc;

    }

    void putdata()

    {

        cout << "\t\tNUMBER OF DAYS :- " << no\_of\_days << endl;

        cout << "\t\tROOM NUMBER :- " << room\_no << endl;

        cout << "\t\tROOM :- " << Ac\_NonAc << endl

             << endl;

    }

    friend struct free\_room \*new\_customer(struct node \*head, int room, struct free\_room \*first);

    friend void \*display\_room(struct free\_room \*first);

    friend struct node \*remove\_customer(struct node \*head, int id, struct free\_room \*first);

    friend struct free\_room \*rooms(struct free\_room \*first);

};

int room ::no\_of\_rooms = 5;

int room ::room\_count = 1000;

class food

{

protected:

    string s;

    int total\_bill\_of\_food;

public:

    food()

    {

        total\_bill\_of\_food = 0;

    }

    void getdata();

    void calculate();

    void putdata()

    {

        cout << "\t\tFOOD BILL :- " << total\_bill\_of\_food << endl

             << "\t\t";

    }

};

inline void food ::getdata()

{

    cout << "\t\tEnter Food Name" << endl

         << "\t\t";

    cin >> s;

    calculate();

}

inline void food ::calculate()

{

    total\_bill\_of\_food = total\_bill\_of\_food + 100;

    cout << "\t\tBILL = " << total\_bill\_of\_food << endl

         << "\t\t";

}

class cost : public room, public food

{

protected:

    int total\_cost;

public:

    void cost\_calculate()

    {

        total\_cost = total\_bill\_of\_food + (500 \* no\_of\_days);

    }

    void putdata()

    {

        cost\_calculate();

        cout << "\t\*\*BILL\*\*" << endl;

        cout << "\t\tROOM RENT  :- " << (500 \* no\_of\_days) << endl;

        cout << "\t\tFOOD BILL  :- " << total\_bill\_of\_food << endl;

        cout << "\t\tTOTAL BILL :- " << total\_cost << endl;

    }

};

class customer : virtual public person, public cost

{

    static int count;

    string password;

public:

    int customer\_id;

public:

    customer()

    {

        count++;

        customer\_id = count;

        cout << "\t\tENTER YOUR PASSWORD" << endl

             << "\t\t";

        cin >> password;

        cout << "\t\tYour id is " << customer\_id << endl

             << "\t\t";

    }

    int checkpassword(string s)

    {

        if (s.compare(password) == 0)

        {

            return 1;

        }

        else

        {

            return 0;

        }

    }

    void getdata1()

    {

        cout << "\t\tID = " << customer\_id << endl

             << "\t\t";

    }

    void putdata()

    {

        person::putdata();

        cout << "\t\tCUSTOMER ID :- " << customer\_id << endl;

        room::putdata();

    }

    int returnid()

    {

        return customer\_id;

    }

};

int customer ::count = 100;

struct free\_room

{

    int data;

    struct free\_room \*next;

};

struct free\_room \*rooms(struct free\_room \*first)

{

    struct free\_room \*p;

    struct free\_room \*q;

    struct free\_room \*r;

    struct free\_room \*s;

    struct free\_room \*t;

    q = new struct free\_room;

    r = new struct free\_room;

    s = new struct free\_room;

    t = new struct free\_room;

    first = new struct free\_room;

    p = first;

    p->data = cost::room\_count + 1;

    p->next = q;

    q->data = cost::room\_count + 2;

    q->next = r;

    r->data = cost::room\_count + 3;

    r->next = s;

    s->data = cost::room\_count + 4;

    s->next = t;

    t->data = cost::room\_count + 5;

    t->next = NULL;

    return first;

}

struct free\_room \*empty(struct free\_room \*first)

{

    struct free\_room \*p;

    p = first;

    while (p != NULL)

    {

        cout << "\t\t" << p->data << endl;

        p = p->next;

    }

    return first;

}

struct node

{

    customer c;

    struct node \*next;

};

struct node \*new\_customer(struct node \*head, int room)

{

    if (head == NULL)

    {

        head = new struct node;

        head->c.room\_no = room;

        head->next = NULL;

    }

    else

    {

        struct node \*p1 = head;

        struct node \*ptr1 = new struct node;

        ptr1->c.room\_no = room;

        while (p1->next != NULL)

        {

            p1 = p1->next;

        }

        ptr1->next = p1->next;

        p1->next = ptr1;

    }

    cout << "\t\tcustomer inserted successfully" << endl;

    return head;

}

struct free\_room \*new\_customer(struct node \*head, int room, struct free\_room \*first, int \*check)

{

    int room\_found;

    struct free\_room \*p;

    struct free\_room \*q;

    p = first;

    if (p != NULL)

    {

        if (p->data == room)

        {

            if (p->next == NULL)

            {

                first = NULL;

                free(p);

            }

            else

            {

                first = first->next;

                free(p);

            }

            room\_found = 1;

        }

        else

        {

            q = first->next;

            while (q != NULL)

            {

                if (q->data == room)

                {

                    p->next = q->next;

                    free(q);

                    room\_found = 1;

                    break;

                }

                else

                {

                    room\_found = 0;

                }

                p = p->next;

                q = q->next;

            }

        }

        if (room\_found == 1)

        {

            \*check = 1;

        }

        else

        {

            \*check = 0;

            cout << "\t\tInvalid Room No\n"

                 << endl;

        }

    }

    else

    {

        \*check = 0;

        cout << "\t\tRooms Housefull \n"

             << endl;

    }

    return first;

}

struct free\_room \*remove\_customer(struct free\_room \*first, int room)

{

    struct free\_room \*p1;

    p1 = first;

    if (p1 == NULL)

    {

        p1 = new struct free\_room;

        p1->data = room;

    }

    else

    {

        struct free\_room \*ptr;

        ptr = new struct free\_room;

        ptr->data = room;

        while (p1->next != NULL)

        {

            if (p1->next->data < room)

            {

                ptr->next = p1->next;

                p1->next = ptr;

                break;

            }

            p1 = p1->next;

        }

        if (p1->next == NULL)

        {

            ptr->next = p1->next;

            p1->next = ptr;

        }

    }

    return first;

}

struct node \*remove\_customer(struct node \*head, int id, struct free\_room \*first, int \*check, int &room)

{

    int found\_id = 0;

    struct node \*p;

    p = head;

    if (head != NULL)

    {

        if (p->c.customer\_id == id)

        {

            p->c.cost::putdata();

            getch();

            head = head->next;

            \*check = 1;

            free(p);

            cout << "\t\tCUSTOMER REMOVED" << endl

                 << "\t\t";

            goto end;

        }

        while (p != NULL)

        {

            if (p->c.customer\_id == id)

            {

                found\_id = 1;

                break;

            }

            p = p->next;

        }

        if (found\_id == 1)

        {

            struct node \*q;

            q = head;

            if (head->next == NULL && head->c.customer\_id == id)

            {

                head->c.cost::putdata();

                getch();

                room = head->c.room\_no;

                head == NULL;

            }

            else

            {

                while (q->next != p)

                {

                    q = q->next;

                }

                p->c.cost::putdata();

                getch();

                room = p->c.room\_no;

                q->next = p->next;

                free(p);

            }

            cout << "\t\tCUSTOMER REMOVED" << endl

                 << "\t\t";

            \*check = 1;

        }

        else

        {

            \*check = 0;

            cout << "\t\tInvalid Id\n"

                 << endl;

        }

    }

    else

    {

        \*check = 0;

        cout << "\t\tEMPTY HOTEL\n"

             << endl;

    }

end:

    return head;

}

struct node \*display(struct node \*head)

{

    int i = 1;

    struct node \*p = head;

    while (p != NULL)

    {

        cout << "\t\t" << i << ") Id = " << p->c.customer\_id << endl;

        p = p->next;

        i++;

    }

    return head;

}

struct node \*food\_order(struct node \*head, int id)

{

    string s;

    struct node \*p = head;

    while (p != NULL)

    {

        if (p->c.customer\_id == id)

        {

            break;

        }

        p = p->next;

    }

    if (p != NULL)

    {

        cout << "\t\tENTER YOUR PASSWORD" << endl

             << "\t\t";

        cin >> s;

        if (p->c.checkpassword(s) == 1)

        {

            p->c.getdata();

        }

        else

        {

            cout << "\t\tWRONG PASSWORD" << endl

                 << "\t\t";

        }

    }

    else

    {

        cout << "\t\tInvalid Customer ID" << endl

             << "\t\t";

    }

    return head;

}

struct node \*search(struct node \*head, int &id)

{

    struct node \*p = head;

    while (p != NULL)

    {

        if (p->c.customer\_id == id)

        {

            id = 1;

            break;

        }

        p = p->next;

    }

    if (p == NULL)

    {

        id = 0;

    }

    return head;

}

struct node \*search(struct node \*head, string name)

{

    struct node \*p = head;

    string s;

    while (p != NULL)

    {

        s = p->c.return\_name();

        if (s.compare(name) == 0)

        {

            cout << "\t\tCUSTOMER FOUND" << endl;

            break;

        }

        p = p->next;

    }

    if (p == NULL)

    {

        cout << "\t\tCUSTOMER NOT FOUND" << endl;

    }

    return head;

}

struct node \*info(struct node \*head, int id)

{

    string s;

    struct node \*p = head;

    while (p != NULL)

    {

        if (p->c.customer\_id == id)

        {

            cout << "\t\tENTER YOUR PASSWORD" << endl;

            cin >> s;

            system("cls");

            if (p->c.checkpassword(s) == 1)

            {

                p->c.putdata();

            }

            else

            {

                cout << "\t\tWRONG PASSWORD" << endl;

            }

            break;

        }

        p = p->next;

    }

    if (p == NULL)

    {

        cout << "\t\tINVALID ID" << endl;

    }

    return head;

}

int main()

{

    fstream file;

    manager m;

    string name, pass;

    int a, id, room, check = 0, chk = 0;

    int b, c;

    struct free\_room \*first;

    first = NULL;

    first = rooms(first);

    struct node \*head;

    struct node \*f;

    head = NULL;

    system("cls");

    cout << endl

         << endl

         << "\t\t\t\t\t\t\tHOTEL MANAGEMENT SYSTEM" << endl;

    cout << endl

         << endl

         << "\t\tMANAGER LOGIN" << endl;

    m.setpassword();

    do

    {

        cout << "\t\t1.MANAGER" << endl

             << "\t\t2.CUSTOMER" << endl

             << "\t\t3.EXIT" << endl

             << "\t\t";

        cin >> b;

        system("cls");

        switch (b)

        {

        case 1:

            cout << "\t\tHELLO MANAGER" << endl

                 << "\t\tENTER YOUR PASSWORD" << endl

                 << "\t\t";

            cin >> pass;

            system("cls");

            if (m.checkpassword(pass) == 1)

            {

                do

                {

                    printf("1.NEW CUSTOMER \n2.DISPLAY ID \n3.REMOVE CUSTOMER\n4.DISPLAY INFORMATION OF CUSTOMER\n5.Display file\n6.Search in file\n7.BACK\n");

                    scanf("%d", &a);

                    system("cls");

                    if (a == 1)

                    {

                        if (first != NULL)

                        {

                            cout << "\t\tFree Rooms Are :" << endl;

                            first = empty(first);

                            cout << endl

                                 << "\t\tEnter Room No :" << endl

                                 << "\t\t";

                            cin >> room;

                            first = new\_customer(head, room, first, &check);

                            if (check == 1)

                            {

                                head = new\_customer(head, room);

                                f = head;

                                while (f->next != NULL)

                                {

                                    f = f->next;

                                }

                                file.open("file.csv", ios::in | ios::app | ios::out);

                                file.seekg(0, ios::end);

                                string init = "Customer Id,Customer Name,Phone Number,Address,No of days live in hotel,AC/NON AC,room no";

                                if (file.tellg() == 0)

                                {

                                    file << init << "\n";

                                }

                                file << f->c.customer\_id << ",";

                                file << f->c.return\_name() << ",";

                                file << f->c.returnph() << ",";

                                file << f->c.return\_adress() << ",";

                                file << f->c.returndays() << ",";

                                file << f->c.returnac() << ",";

                                file << room << ",\n";

                                file.close();

                                file.open("file2.txt", ios::in | ios::app | ios::out);

                                file << f->c.customer\_id << endl;

                                file << f->c.return\_name() << endl;

                                file << f->c.returnph() << endl;

                                file << f->c.return\_adress() << endl;

                                file << f->c.returndays() << endl;

                                file << f->c.returnac() << endl;

                                file << room << endl;

                                file.close();

                            }

                            check = 0;

                        }

                        else

                        {

                            printf("\t\tROOMS HOUSEFULL \n");

                        }

                        getch();

                    }

                    if (a == 2)

                    {

                        head = display(head);

                        getch();

                    }

                    if (a == 3)

                    {

                        printf("\t\tEnter Your ID\n");

                        scanf("%d", &id);

                        system("cls");

                        head = remove\_customer(head, id, first, &chk, room);

                        if (chk == 1)

                        {

                            first = remove\_customer(first, room);

                        }

                        chk = 0;

                        getch();

                    }

                    if (a == 4)

                    {

                        cout << "\t\tEnter ID" << endl

                             << endl

                             << "\t\t";

                        cin >> id;

                        system("cls");

                        head = info(head, id);

                        getch();

                    }

                    if (a == 5)

                    {

                        file.open("file2.txt");

                        while (file.good())

                        {

                            getline(file, name);

                            check = 1;

                            cout << "Customer ID               : " << name << endl;

                            getline(file, name);

                            cout << "NAME                      : " << name << endl;

                            getline(file, name);

                            cout << "PHONE NUMBER              : " << name << endl;

                            getline(file, name);

                            cout << "ADRESS                    : " << name << endl;

                            getline(file, name);

                            cout << "NO OF DAYS STAY IN HOTEL  : " << name << endl;

                            getline(file, name);

                            cout << "ROOM TYPE                 : " << name << endl;

                            getline(file, name);

                            cout << "ROOM NUMBER               : " << room << endl

                                 << endl;

                        }

                        if (check == 0)

                        {

                            cout << "NO DATA FOUND" << endl;

                        }

                        else

                        {

                            check = 0;

                        }

                        file.close();

                        getch();

                    }

                    if (a == 6)

                    {

                        check = 0;

                        cout << "ENTER NAME " << endl;

                        cin >> pass;

                        system("cls");

                        file.open("file2.txt");

                        while (file.good())

                        {

                            getline(file, name);

                            if (pass == name)

                            {

                                check = 1;

                                cout << "NAME                      : " << name << endl;

                                getline(file, name);

                                cout << "PHONE NUMBER              : " << name << endl;

                                getline(file, name);

                                cout << "ADRESS                    : " << name << endl;

                                getline(file, name);

                                cout << "NO OF DAYS STAY IN HOTEL  : " << name << endl;

                                getline(file, name);

                                cout << "ROOM TYPE                 : " << name << endl;

                                getline(file, name);

                                cout << "ROOM NUMBER               : " << room << endl

                                     << endl;

                            }

                        }

                        if (check == 0)

                        {

                            cout << "NO DATA FOUND" << endl;

                        }

                        else

                        {

                            check = 0;

                        }

                        file.close();

                        getch();

                    }

                    system("cls");

                } while (a != 7);

            }

            else

            {

                cout << "\t\tINVALID PASSWORD" << endl

                     << "\t\t";

                getch();

            }

            system("cls");

            break;

        case 2:

            do

            {

                cout << "\t\t1.DISPLAY FREE ROOMS" << endl

                     << "\t\t2.SERVICES" << endl

                     << "\t\t3.BACK" << endl

                     << "\t\t";

                cin >> c;

                system("cls");

                switch (c)

                {

                case 1:

                    first = empty(first);

                    getch();

                    system("cls");

                    break;

                case 2:

                    do

                    {

                        cout << endl

                             << "\t\t1.FOOD ORDER" << endl

                             << "\t\t2.SEARCHING CUSTOMER" << endl

                             << "\t\t3.BACK" << endl

                             << "\t\t";

                        cin >> a;

                        system("cls");

                        switch (a)

                        {

                        case 1:

                            cout << "\t\tEnter Your ID" << endl

                                 << "\t\t";

                            cin >> id;

                            system("cls");

                            head = food\_order(head, id);

                            getch();

                            system("cls");

                            break;

                        case 2:

                            cout << "\t\tSearch By :" << endl

                                 << "\t\t1.ID" << endl

                                 << "\t\t2.NAME" << endl

                                 << "\t\t3.BACK" << endl

                                 << "\t\t";

                            cin >> room;

                            system("cls");

                            switch (room)

                            {

                            case 1:

                                cout << "\t\tENTER ID OF CUSTOMER" << endl

                                     << "\t\t";

                                cin >> id;

                                head = search(head, id);

                                if (id == 1)

                                {

                                    cout << "\t\tCUSTOMER FOUND" << endl

                                         << "\t\t";

                                }

                                else if (id == 0)

                                {

                                    cout << "\t\tCUSTOMER  NOT FOUND" << endl

                                         << "\t\t";

                                }

                                getch();

                                system("cls");

                                break;

                            case 2:

                                cout << "\t\tENTER NAME OF CUSTOMER" << endl

                                     << "\t\t";

                                cin >> name;

                                head = search(head, name);

                                getch();

                                system("cls");

                                break;

                            }

                            break;

                        }

                        system("cls");

                    } while (a != 3);

                }

                system("cls");

            } while (c != 3);

            break;

        }

        system("cls");

    } while (b != 3);

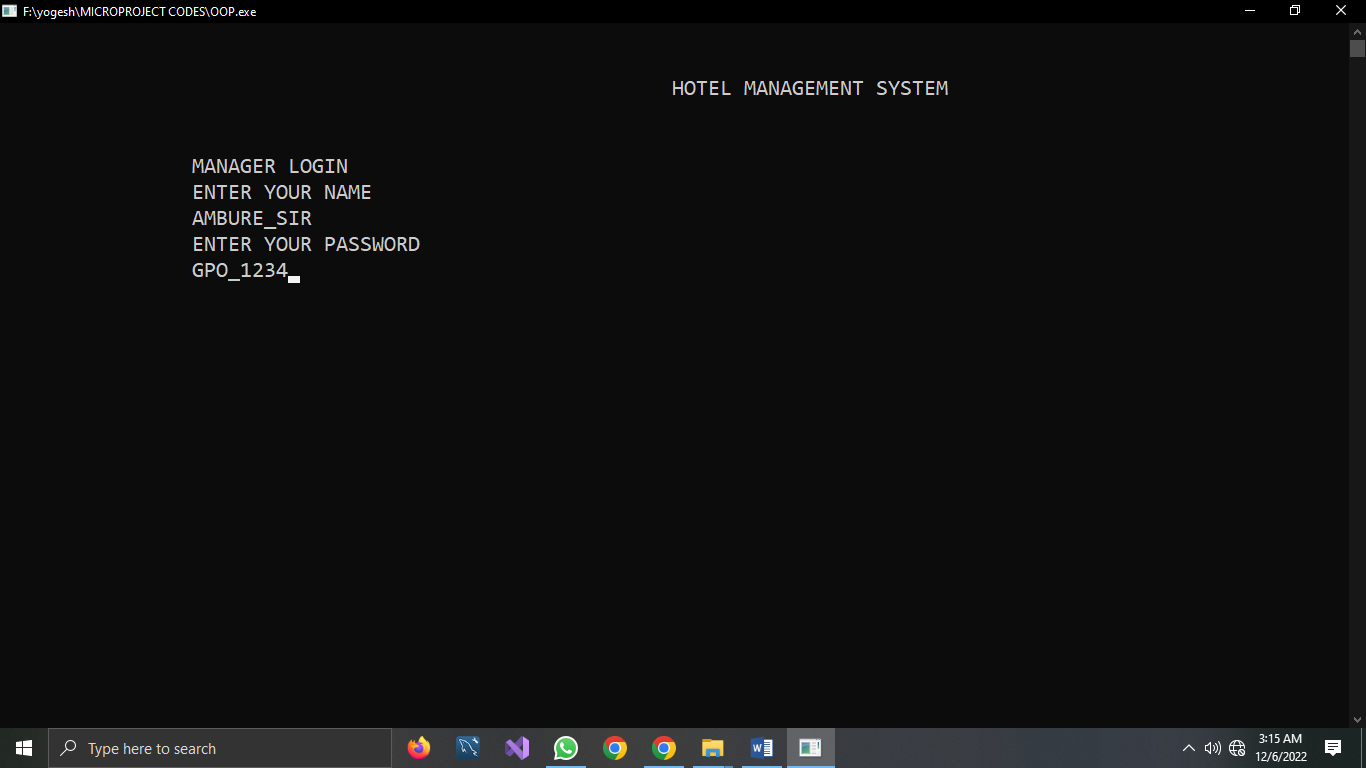
    system("cls");

    cout << "\t\tThank You \n\t\tVisit Again\n\n\t\t";

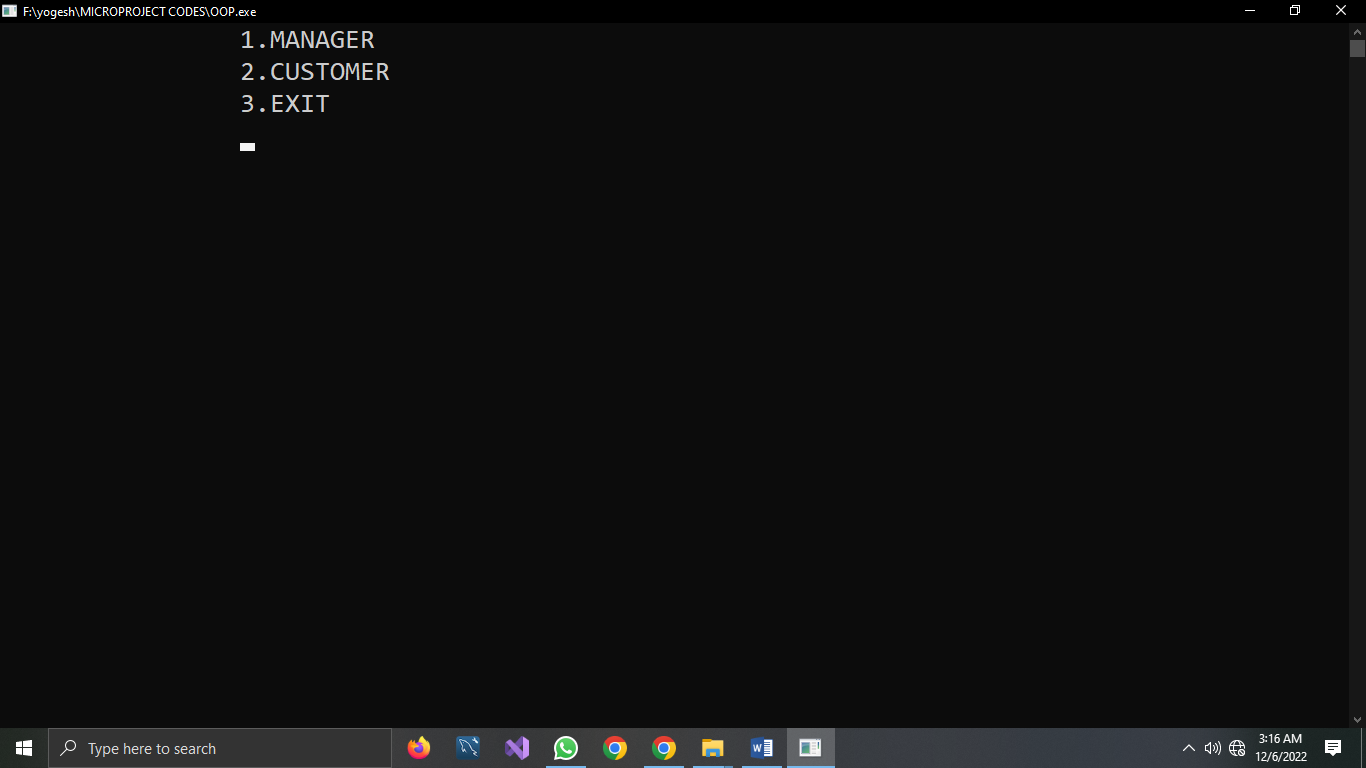
    return 0;

}

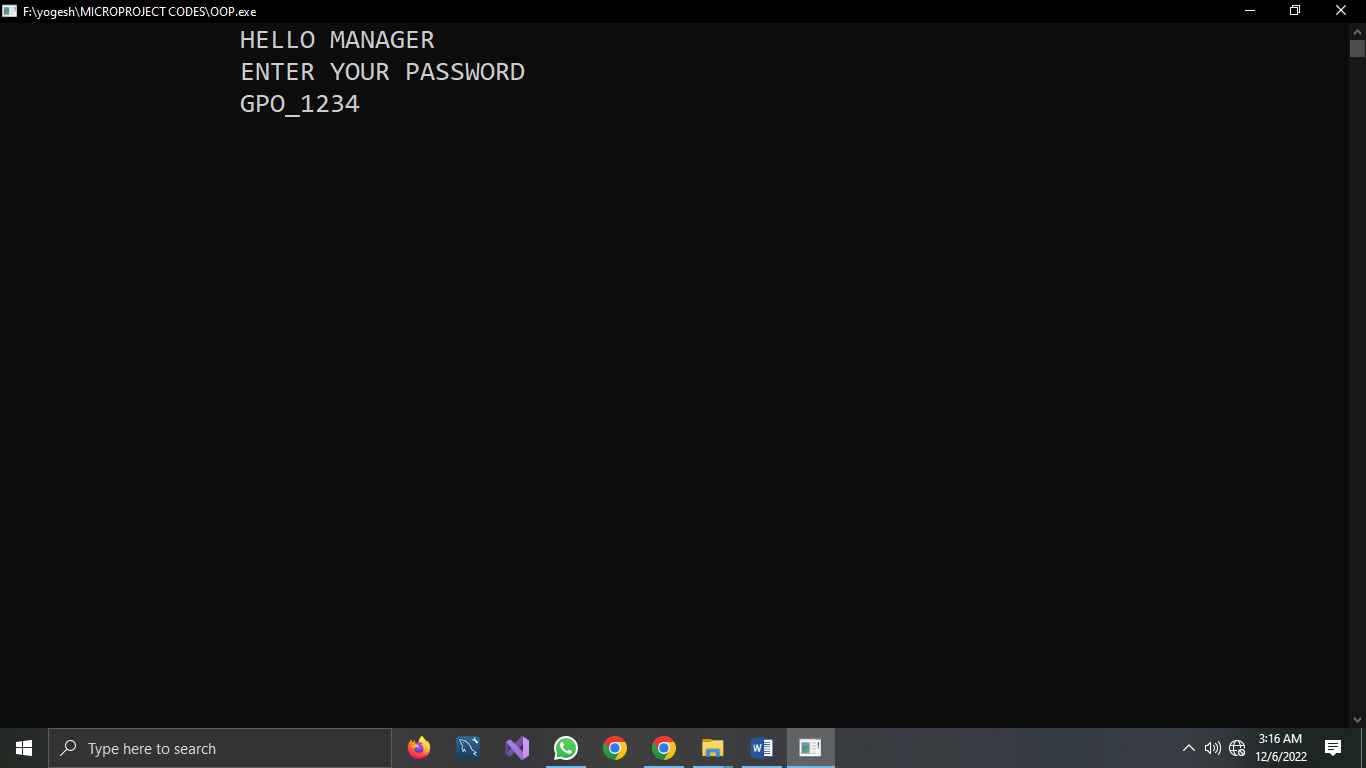
**OUTPUT**



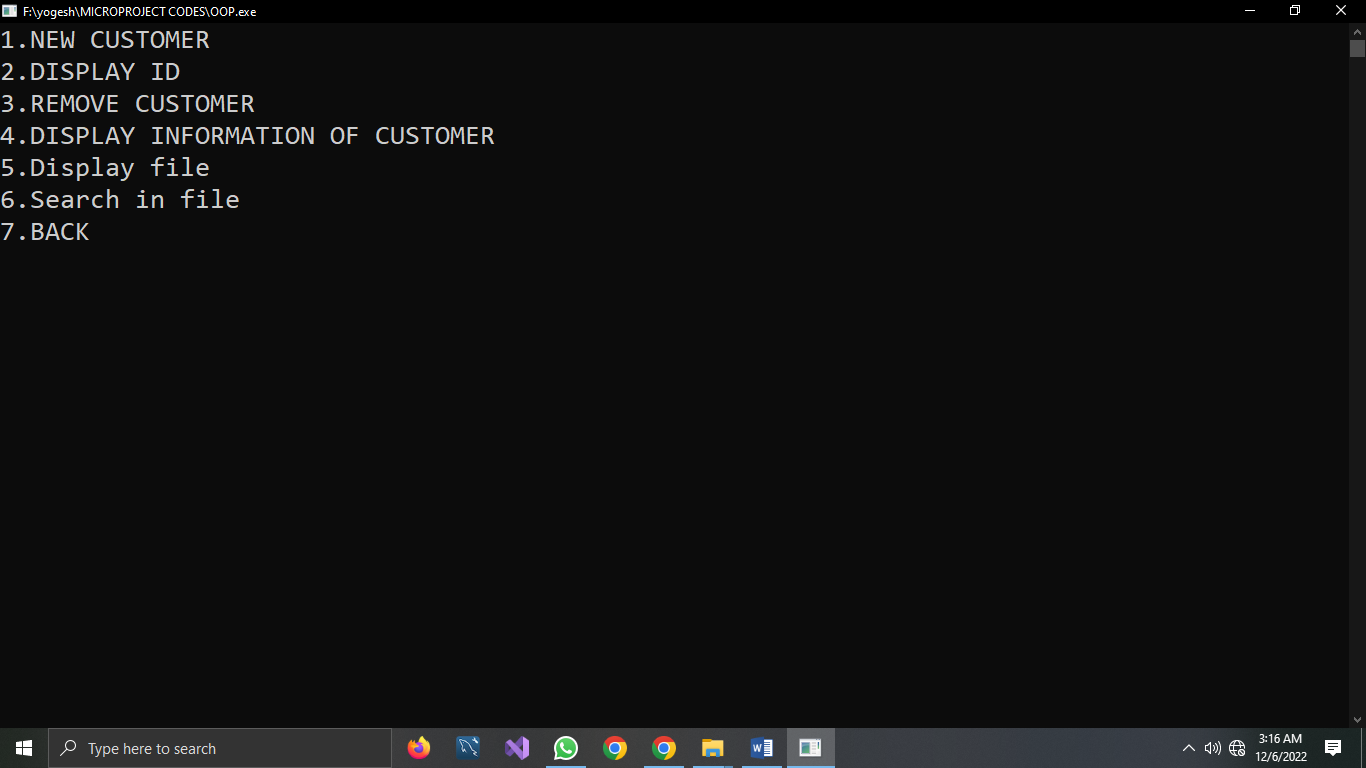
**MANAGER ACCOUNT CRATING**



**OPTIONS**

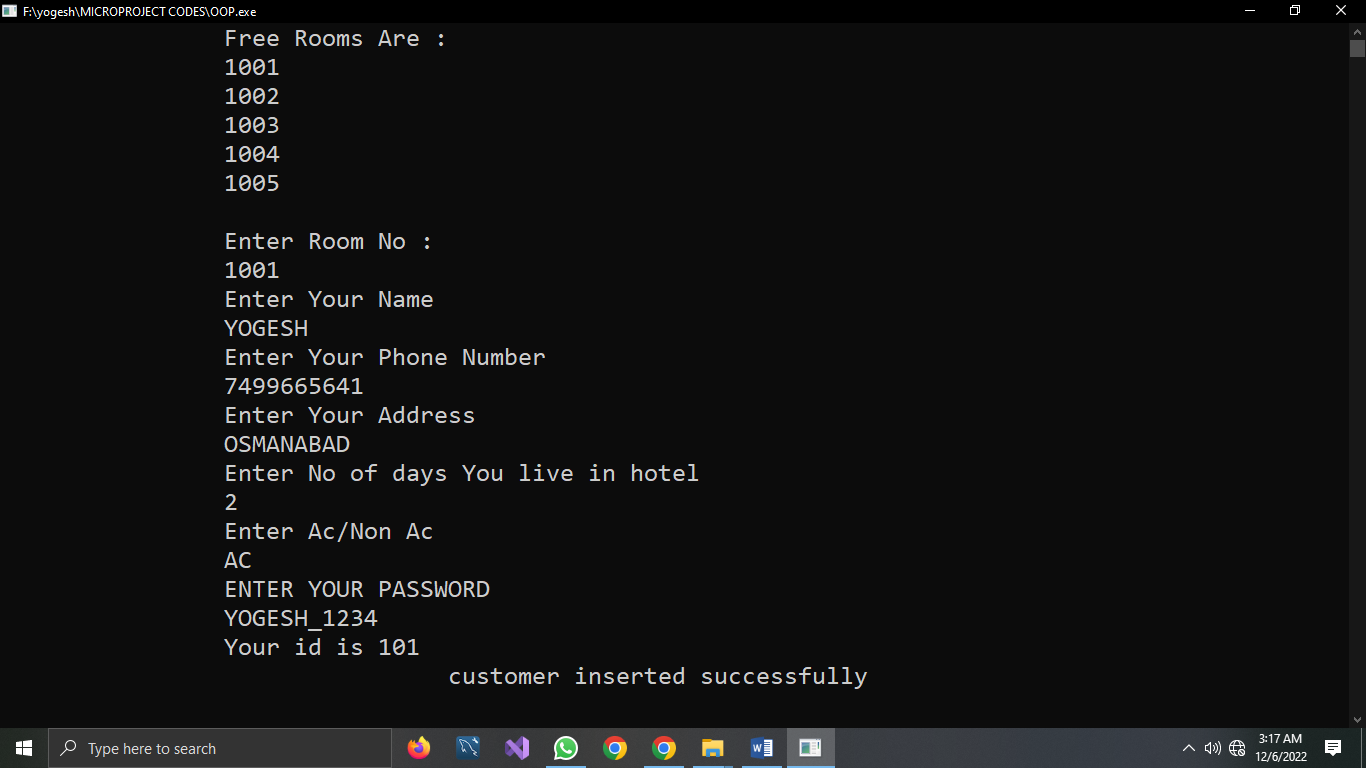


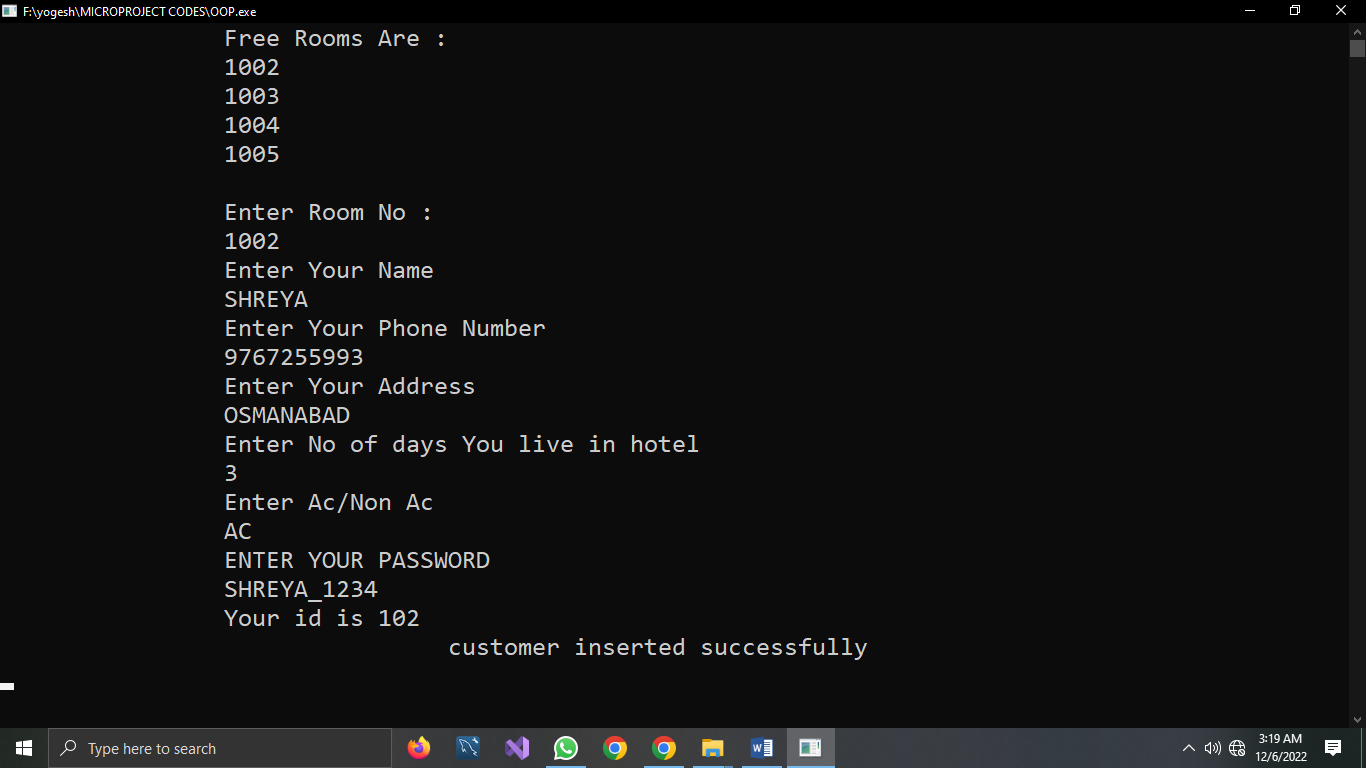
**MANAGER LOGIN**

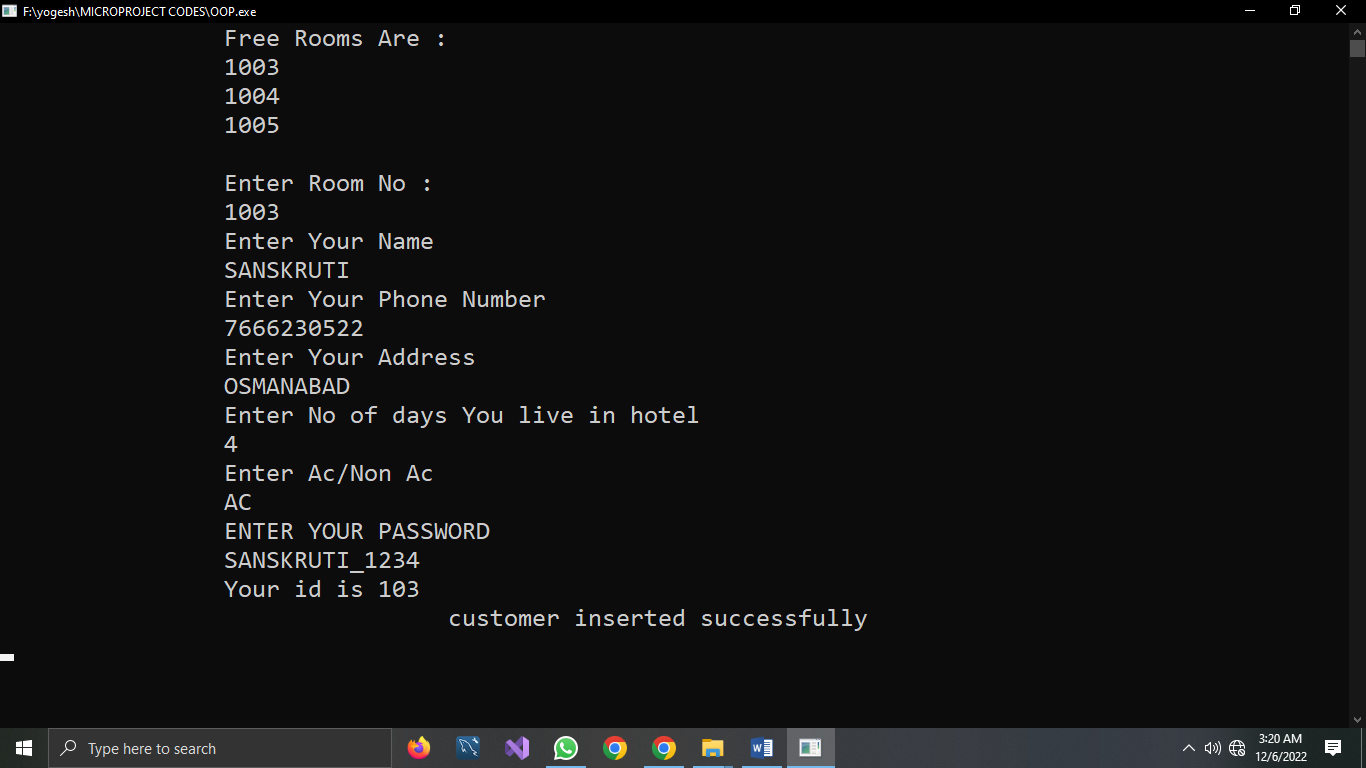


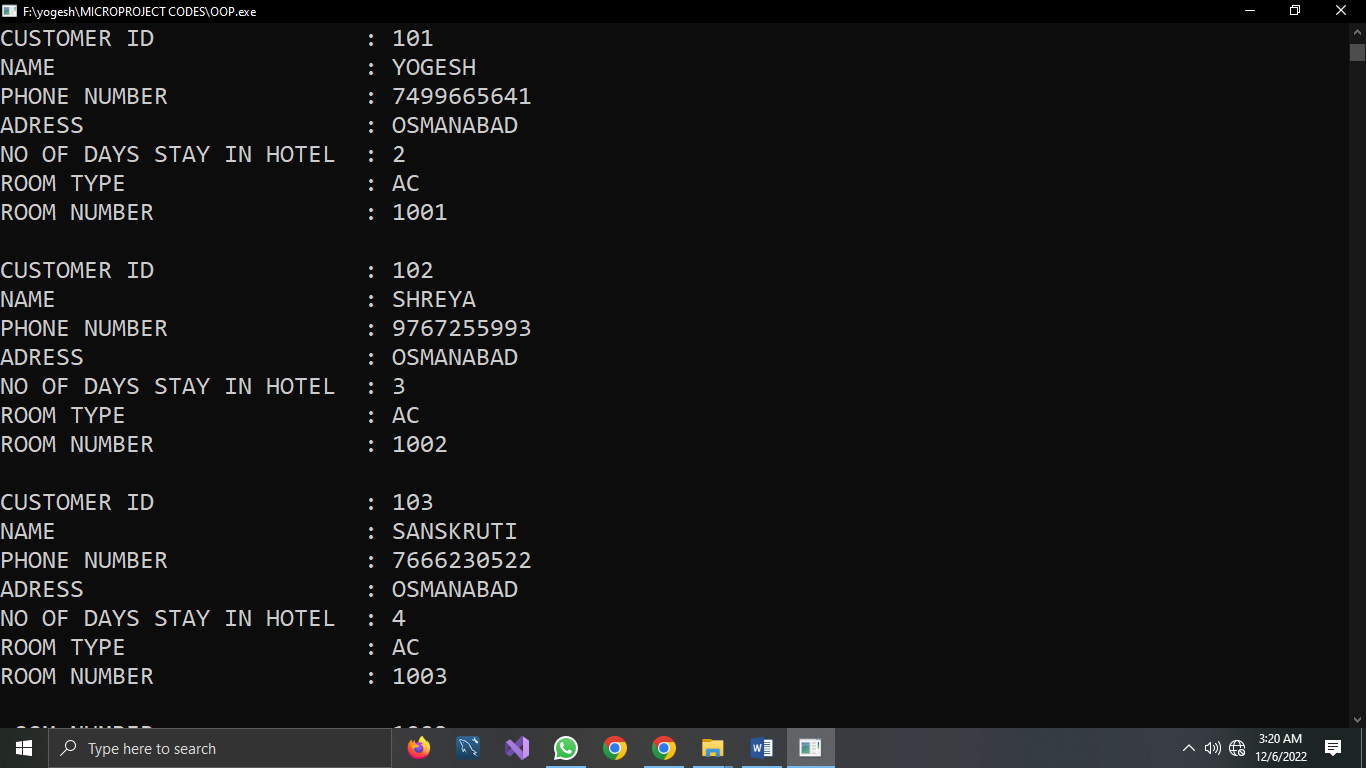
**OPTIONS**

**CUSTOMER ENTRY**

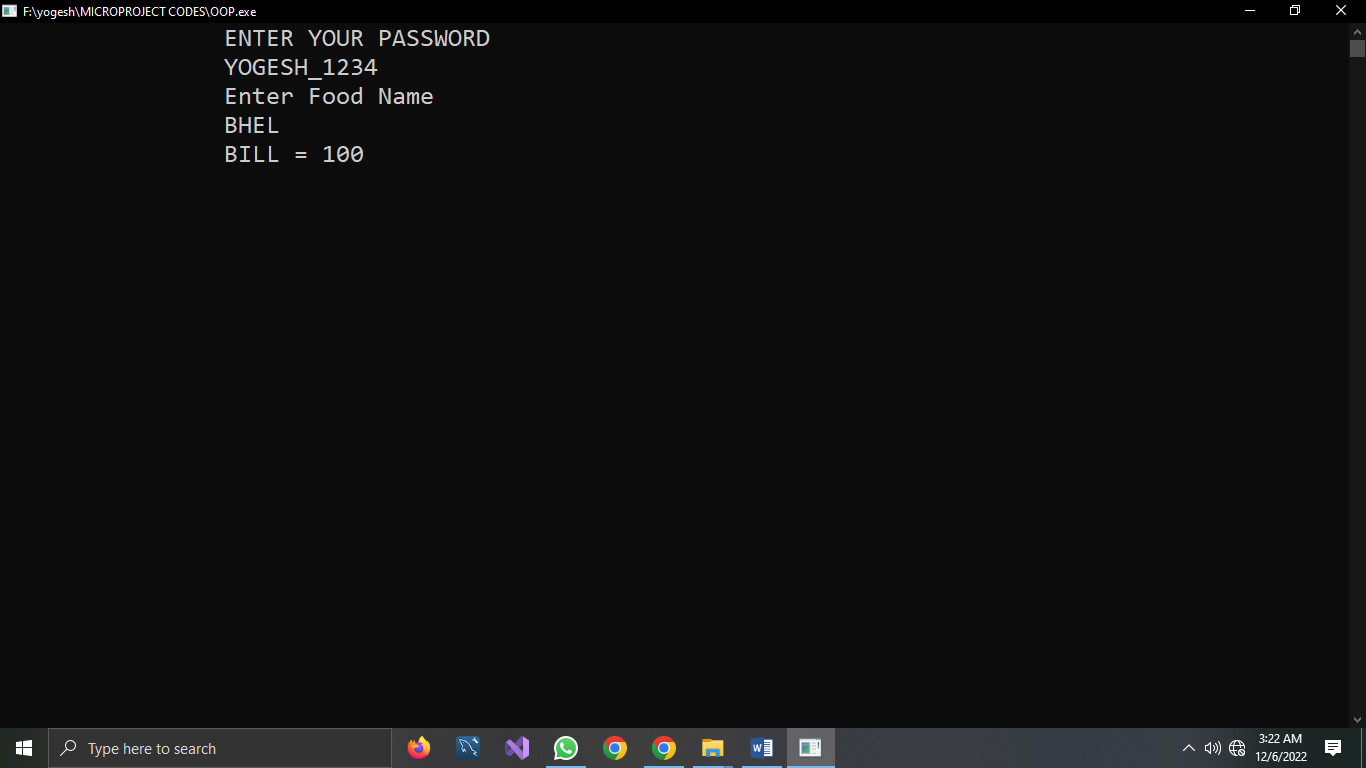
****



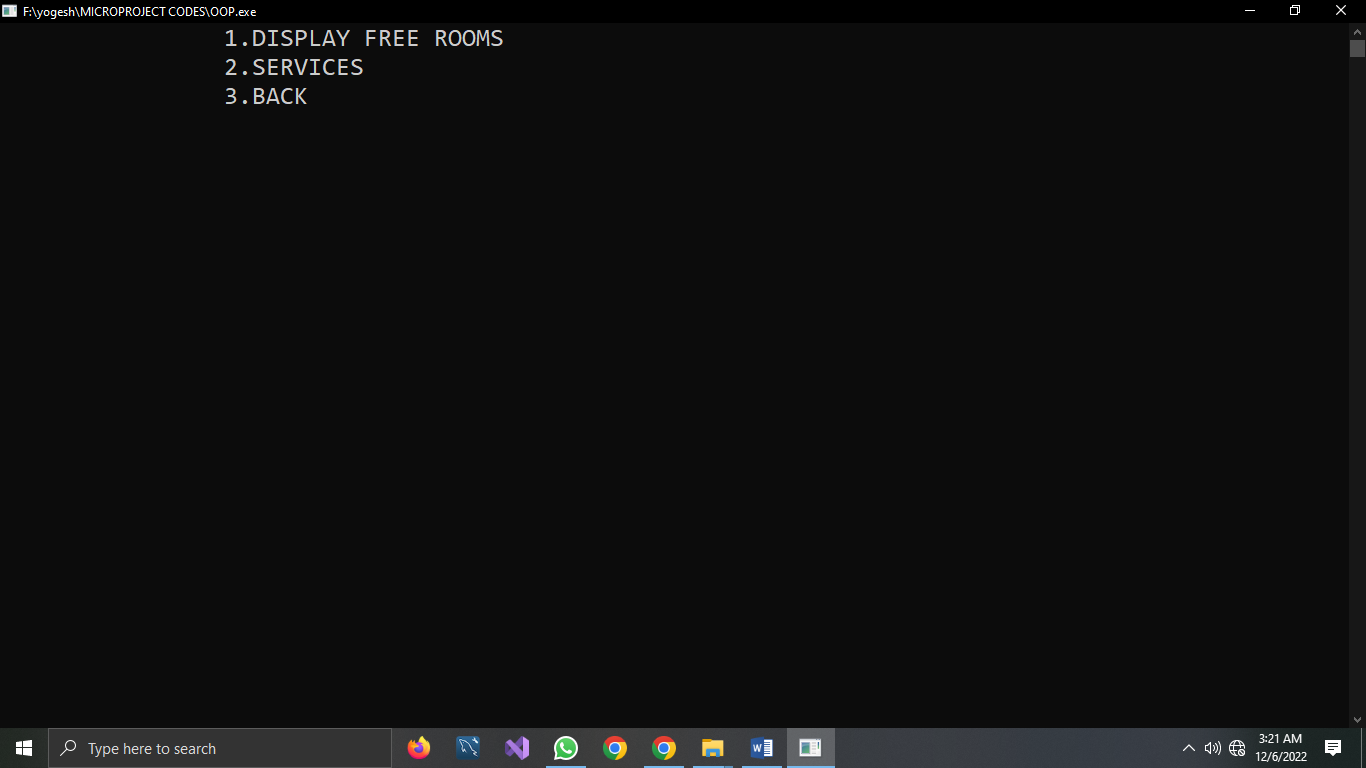
****

****

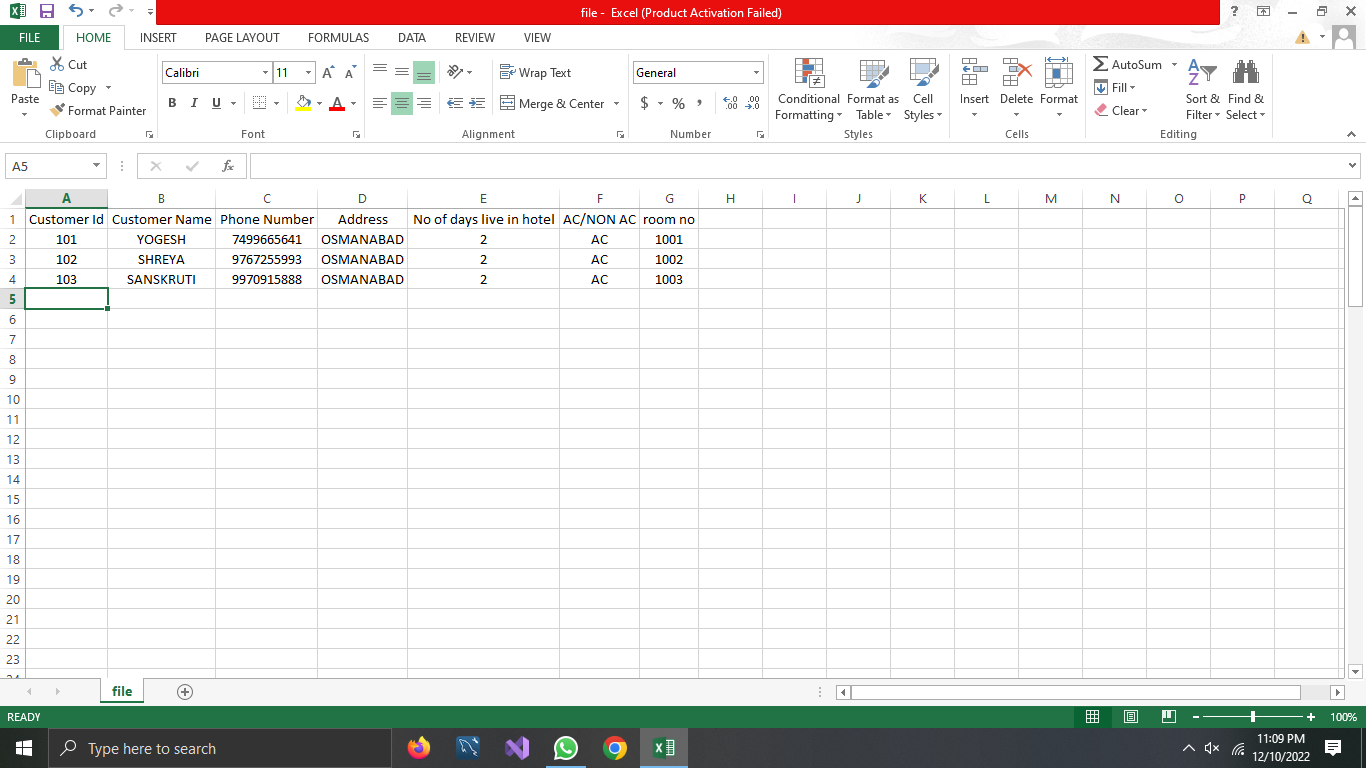
**DISPLAYING DATA FROM FILE**



**FOOD ORDER**

****

**OPTIONS**

****

**FILE**

**RESOURCES USED**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr N.** | **RESOURCES** | **SPECS** | **Qty** | **REMARKS** |
| **1.** | Computer system | RAM : 8 GB  ROM : 512 SSD  OS : WINDOW’S | 1 |  |
| **2.** | Software | Visual Studio , TurboC, VS code | 1 |  |
| **3.** | Any other resources used | Keyboard , Mouse | 1 |  |

**REFERENCE & OUTCOMES**

**REFERENCE**

I take the reference of following ,

SITES

1. [www.computerknowledge.com](http://www.computerknowledge.com)

2. [www.javapoint.com](http://www.javapoint.com)

3. [www.Wikipedia.com](http://www.Wikipedia.com)

4. [www.computer-hope.com](http://www.computer-hope.com)

5. [www.cppinfo.com](http://www.cppinfo.com)

**BOOKS**

1. Introduction to object oriented programming - Budd, Timothy

**SKILL DEVELOPED / LEARNING OUTCOMES**

1. we get proper information about the syntaxes and the proper use of OOP

2. Due to this project handling of the IDE’s become easy and handy