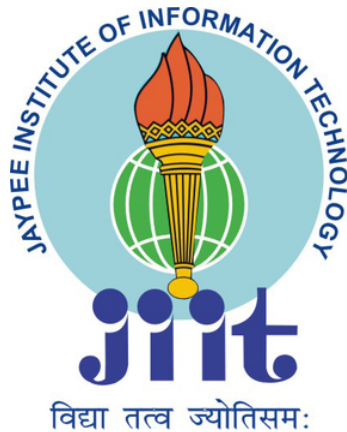


JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

– SECTOR 62, NOIDA

(B. Tech I Year, Even Semester 2023)



Software Development Fundamentals Lab– II
[15B17CI271]

Mini Project in C++ Language – Project REPORT

Project: Hospital Management System

Submitted by:

NAME

ENROLLMENT NO.

Ashwin S Deshpande

22102159 (A7)

Devansh Khare

22102166 (A7)

Sparsh Jain

22102167 (A7)

Mohak Kapoor

22102169 (A7)

INDEX

1. PROBLEM STATEMENT
2. FEATURES OF C++ USED
3. OVERVIEW OF THE CODE
4. OUTPUT AND EXPLANATION

PROBLEM STATEMENT

We are solving the problem of maintaining and keeping record of everyone and everything in a hospital. A major problem in hospitals is maintaining a record of everything and doing that in registers and files is extremely time taking and not so convenient.

Here we are designing a hospital management system to manage various aspects of a hospital's operations, such as patient data management, medical billing, inventory management, and prescribing medicine to the patient. We keep a record of everything from doctors, nurses, receptionists to store manager. The store manager can order medicines and keep the inventory full.

The purpose of this project is to develop a comprehensive and user-friendly hospital management system that streamlines hospital workflows and improves patient care. The system is capable of managing patient records, maintaining medical histories, maintaining the record of everyone working within the hospital from doctors, nurses to receptionist and store manager. We can search for a patient's record and display his/her entire details and refer to a doctor. The doctor can prescribe medicine to the patient. The project aims to improve hospital efficiency, reduce administrative overhead, and enhance patient satisfaction by providing timely and accurate information and services.

FEATURES OF C++ USED

- Libraries

```
#include <stdio.h>
#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
#include <cstdlib>
#include <ctime>
#include <string>
#include <conio.h>
#include <string.h>
#include <time.h>
#include<cstring>
```

- Object oriented programming

- Classes

```
class Person
{   protected:
    char Name[50],Gender[10];
    int Age,Date,Month,Year;
    long int PhoneNum;
    int ZipCode,PersonID;
```

- Inheritance

```
class Staff:public Person
{
    int salary;
    char time_from[30],time_to[30];

public:
```

- Function overloading(polymorphism)

```

void PersonInfo()
{
    cout<<"Enter Name :";
    cin.clear();
    cin.ignore();
    cin.getline(Name,50);
    cout<<"Enter Gender :";
    cin>>Gender;
    cout<<"Enter Age :";
    cin>>Age;
    cout<<"Enter Address :";
    cin>>Country;
    cout<<"Enter Blood group:";
    cin>>blood_g;
}

void PersonInfo(char n[], char g[],int a,char c[],char bg[])
{
    strcpy(Name,n);
    strcpy(Gender,g);
    Age = a;
    strcpy(Country,c);
    strcpy(blood_g,bg);
}

```

○ Virtual functions

```

virtual void getInfo() = 0;
void PersonInfo()

```

- **Abstract classes**

```
class Person
{
protected:
    char Name[50], Gender[10];
    int Age, Date, Month, Year;
    long int PhoneNum;
    int ZipCode, PersonID;
    char City[50], Country[50], State[50], blood_g[5];
public:
    virtual void getInfo() = 0;
    void PersonInfo()
{
```

- **Function declaration and defining**

```
    char medicine[100];
void patient_bill();
void getdata(); //function to accept data from user
void showdata(); //function to show data on screen
int retidno();
void modify_medicine(int n);
    void write_patient(); //write the record in file
    void display_all(); //read all records from file
    void display_p(int n); //accept reg_no and read record from file
    void modify_patient(int n); //accept reg_no and update record of file
    void delete_patient(int n); //accept reg_no and delete selected records from file
};
```

```

void patient::display_p(int n) //reading specific file
{
    patient p;
    ifstream inFile;
    inFile.open("patient.dat",ios::binary);
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();
        cin.get();
        return;
    }
    bool flag=false;
    while(inFile.read(reinterpret_cast<char *> (&p), sizeof(patient)))
    {
        if(p.retidno()==n)
        {
            p.showdata();
            flag=true;
        }
    }
    inFile.close();
    if(flag==false)
        cout<<"\n\nrecord not exist";
    cin.ignore();
    cin.get();
}

```

- File handling using .dat files

```
fstream File;
File.open("patient.dat",ios::binary|ios::in|ios::out);
if(!File)
{
    cout<<"File could not be open !! Press any Key...";
    cin.ignore();
    cin.get();
    return;
}
    while(!File.eof() && found==false)
{

File.read(reinterpret_cast<char *> (&p), sizeof(patient));
if(p.retidno()==n)
{
    p.showdata();
    cout<<"\n\nPlease Enter The New Details of patient"<<endl;
        p.getdata();
        int pos=(-1)*static_cast<int>(sizeof(p));
        File.seekp(pos,ios::cur);
        File.write(reinterpret_cast<char *> (&p), sizeof(patient));
        cout<<"\n\n\t Record Updated";
        found=true;
    }
```


OVERVIEW OF THE CODE

```
#include <stdio.h>
#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
#include <cstdlib>
#include <ctime>
#include <string>
#include <conio.h>
#include <string.h>
#include <time.h>
#include<cstring>

using namespace std;

class Person
{    protected:
    char Name[50],Gender[10];
    int Age,Date,Month,Year;
    long int PhoneNum;
    int ZipCode,PersonID;
    char City[50],Country[50],State[50],blood_g[5];
public:
    virtual void getInfo() = 0;
    void PersonInfo()
    {
        cout<<"Enter Name :";
        cin.clear();
        cin.ignore();
        cin.getline(Name,50);
        cout<<"Enter Gender :";
        cin>>Gender;
        cout<<"Enter Age :";
        cin>>Age;
        cout<<"Enter Address :";
        cin>>Country;
        cout<<"Enter Blood group:";
        cin>>blood_g;
```

```

    }
    void PersonInfo(char n[], char g[],int a,char c[],char bg[])
    {
        strcpy(Name,n);
        strcpy(Gender,g);
        Age = a;
        strcpy(Country,c);
        strcpy(blood_g,bg);

    }

    void Display_info()
    {
        cout<<"Name: "<<Name<<endl;
        cout<<"Gender: "<<Gender<<endl;
        cout<<"Age: "<<Age<<endl;
        cout<<"Address: "<<this->Country<<endl;
        cout<<"Blood Group:"<<blood_g<<endl;
    }
};

class Staff:public Person
{
    int salary;
    char time_from[30],time_to[30];

public:
    void getInfo(){
        cout << "Staff name: "<< Name <<endl<<"Gender:
"<<Gender<<endl<<"Salary: "<< salary <<endl<< "Working
hours"<<time_from<<":"<<time_to;
    }
    void staff_info()
    {
        PersonInfo();
        cout<<"Enter salary:"<<cin>>salary;
        cout<<"Enter Duty time:";
        cout<<"From:"<<cin>>time_from;cout<<"To:"<<cin>>time_to;
    }

    void show_staff()
    {
        cout<<"\nSalary: "<<salary;
        cout<<"\nDuty Time:
"<<time_from<<"--"<<time_to<<endl;
        Display_info();
    }
}

```

```

};

class patient: public Person
{
int reg_no;
char occupation[30];
char date_admit[30];
char date_discharge[30];
int bill,total;

public:
    void getInfo(){
        cout << "Staff name: "<< Name <<endl<<"Gender:
"<<Gender<<endl<<"Date of Admission "<< date_admit <<endl<<
"Date of Discharge"<<date_discharge<<endl<<"Bill: "<<bill;
    }
    char medicine[100];
void patient_bill();
void getdata(); //function to accept data from user
void showdata(); //function to show data on screen
int retidno();
void modify_medicine(int n);
    void write_patient(); //write the record in file
    void display_all(); //read all records from file
    void display_p(int n); //accept reg_no and read record from
file
    void modify_patient(int n); //accept reg_no and update
record of file
    void delete_patient(int n); //accept reg_no and delete
selected records from file
};

void patient::patient_bill() //using scope resolution operator
to define function outside class
{
    system("CLS");
cout <<
"*****
*****";
cout << "\n\n\t\t\tPatient Fee Charge\n\n ";
cout <<
"*****
*****";
//cout << "\nHospital Room Charges          :\t
1000";

```

```

cout << "\nHospital Care Charges           :\t 50";
cout << "\nHospital Maintaince Charges      :\t 50";
cout << "\nPatient Individual Charges       :\t " <<
bill;
total = 50 + 50 + bill;
cout << "\nTotal Fee Charge                 :\t" <<
total<<"\n";
cout <<
"*****";
}

void patient::getdata()
{
    PersonInfo();
    cout<<"\nEnter The registration number of the patient: ";
    cin>>reg_no;
    cout<<"\nEnter The occupation of the patient: ";
    cin>>occupation;
    cout<<"\nEnter The bill of the patient:";
    cin>>bill;
    cout<<"\nEnter The date admitted(dd/mm/yyyy):";
    cin>>date_admit;
    cout<<"\nEnter The date discharged(dd/mm/yyyy):";
    cin>>date_discharge;
    patient_bill();
    strncpy(medicine, "null", sizeof(medicine) - 1);
    medicine[sizeof(medicine) - 1] = 0;
}

void patient::showdata()
{
    cout<<"\nRegistration number of patient: "<<reg_no<<"\n";
    Display_info();
    cout<<"\nOccupation : "<<occupation;
    cout<<"\nAdmit date : "<<date_admit;
    cout<<"\nDischarhge date : "<<date_discharge;
    cout<<"\nBill : "<<total;
    cout<<"\nPrescribed Medicine: "<<medicine;

}

int patient::retidno()
{
    return reg_no;
}

```

```

void patient::modify_medicine(int n)
{
    bool found=false;
    patient p;
    fstream File;
    File.open("patient.dat",ios::binary|ios::in|ios::out);
    if(!File)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();
        cin.get();
        return;
    }
    while(!File.eof() && found==false)
    {

        File.read(reinterpret_cast<char *> (&p), sizeof(patient));
        if(p.retidno()==n)
        {
            p.showdata();
            cout<<"\n\nPlease Enter The Medicine of patient"<<endl;
                cin>>p.medicine;
                int pos=(-1)*static_cast<int>(sizeof(p));
                File.seekp(pos,ios::cur);
                File.write(reinterpret_cast<char *> (&p), sizeof(patient));
                cout<<"\n\n\tRecord Updated";
                found=true;
            }
        }
        File.close();
        if(found==false)
        cout<<"\n\nRecord Not Found ";
        cin.ignore();
        cin.get();
    }

}

void patient::write_patient() // writing into file
{
    patient p;
    ofstream outFile;
    outFile.open("patient.dat",ios::binary|ios::app);
    p.getdata();
    outFile.write(reinterpret_cast<char *> (&p), sizeof(patient));
    outFile.close();
    cout<<"\n\nPatient record Has Been Created ";
    cin.ignore();
}

```

```

cin.get();
}

void patient::display_all() //reading from file
{
patient p;
ifstream inFile;
inFile.open("patient.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
while(inFile.read(reinterpret_cast<char *> (&p),
sizeof(patient)))
{
p.showdata();
cout<<"\n\n=====
=====\\n";
}
inFile.close();
cin.ignore();
cin.get();
}

void patient::display_p(int n) //reading specific file
{
patient p;
ifstream inFile;
inFile.open("patient.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
bool flag=false;
while(inFile.read(reinterpret_cast<char *> (&p),
sizeof(patient)))
{
if(p.retidno()==n)
{
p.showdata();

```

```

flag=true;
}
}
inFile.close();
if(flag==false)
cout<<"\n\nrecord not exist";
cin.ignore();
cin.get();
}

void patient:: modify_patient(int n)    //modifying patient
details
{
bool found=false;
patient p;
fstream File;
File.open("patient.dat",ios::binary|ios::in|ios::out);
if(!File)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
    while(!File.eof() && found==false)
{

File.read(reinterpret_cast<char *> (&p), sizeof(patient));
if(p.retidno()==n)
{
p.showdata();
cout<<"\n\nPlease Enter The New Details of patient"<<endl;
        p.getdata();
        int pos=(-1)*static_cast<int>(sizeof(p));
        File.seekp(pos,ios::cur);
        File.write(reinterpret_cast<char *> (&p), sizeof(patient));
        cout<<"\n\n\t Record Updated";
        found=true;
    }
}
File.close();
if(found==false)
cout<<"\n\n Record Not Found ";
cin.ignore();
cin.get();
}

```

```

void patient:: delete_patient(int n)  //deleting a particular
file
{
patient p;
ifstream inFile;
inFile.open("patient.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
ofstream outFile;
outFile.open("Temp.dat",ios::out);
inFile.seekg(0,ios::beg);
while(inFile.read(reinterpret_cast<char *> (&p),
sizeof(patient)))
{
if(p.retidno()!=n)
{
outFile.write(reinterpret_cast<char *> (&p), sizeof(patient));
}
}
outFile.close();
inFile.close();
remove("patient.dat");
rename("Temp.dat","patient.dat");
cout<<"\n\n\tRecord Deleted ..";
cin.ignore();
cin.get();
}

```

```

class Doctor: public Staff
{
    int DocID;
    patient p;
    char Spz[500];

public:
    bool CheckRegNo(int n)
    {

        bool found=false;

```



```

    Doctor d;
    fstream File;
    File.open("Doctor.dat",ios::binary|ios::in);
    if(!File)
    {
        cout<<"File could not be open !! Press any Key...";
        cin.ignore();
        cin.get();
    }
    while(!File.eof() && found==false)
    {

File.read(reinterpret_cast<char *> (&d), sizeof(Doctor));
if(d.DocID==n)
{
    found=true;
}

    }
    File.close();
    return found;
    cin.ignore();
    cin.get();

}

void DoctorInfo()
{
    staff_info();
    id:
    cout<<"Enter Doctor ID(1200-1299):";
    int m;
    cin>>m;
    if(CheckRegNo(m)==false)
    {
        DocID=m;
    }
    else
    {
        cout<<"ID already taken\n";
        goto id;
    }

    cout<<"Enter Doctor Specialize :";
    cin>>Spz;

```

```

        system("pause");
    }

    void show_doctor()
    {
        cout<<"Doctor ID: "<<DocID;
        show_staff();

        cout<<"\nDoctor Specialize: "<<Spz;
    }

    int  retDid()
    {
return DocID;
    }

    void Prescribe_Medicine()
    {
        int n;
        cout<<"\n\n\tEnter Reg Number Of patient : "; cin>>n;
        p.modify_medicine(n);
    }


    void write_doctor()
    {
Doctor d;
ofstream outFile;
outFile.open("Doctor.dat",ios::binary|ios::app);
d.DoctorInfo();
outFile.write(reinterpret_cast<char *> (&d), sizeof(Doctor));

        cout<<"\n\nDoctor record Has Been Created ";
outFile.close();
cin.ignore();
    }


    void display_all_Doctor()
    {
Doctor d;
ifstream inFile;
inFile.open("Doctor.dat",ios::binary);
if(!inFile)
    {
cout<<"File could not be open !! Press any Key...";
cin.ignore();
    }
    }

```

```

cin.get();
return;
}
cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
while(inFile.read(reinterpret_cast<char *> (&d),
sizeof(Doctor)))
{
d.show_doctor();
cout<<"\n\n=====
=====\\n";
}
inFile.close();
cin.ignore();
cin.get();
}

```

```

void display_d(int n)
{
Doctor d;
ifstream inFile;
inFile.open("Doctor.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
bool flag=false;
while(inFile.read(reinterpret_cast<char *> (&d),
sizeof(Doctor)))
{
if(d.DocID==n)
{
d.show_doctor();
flag=true;
}
}
inFile.close();
if(flag==false)
cout<<"\n\nrecord not exist";
cin.ignore();
cin.get();
}

```

```

void modify_doctor(int n)

```

```

{
bool found=false;
Doctor d;
fstream File;
File.open("Doctor.dat",ios::binary|ios::in|ios::out);
if(!File)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
    while(!File.eof() && found==false)
{

File.read(reinterpret_cast<char *> (&d), sizeof(Doctor));
if(d.DocID==n)
{
    d.show_doctor();
    cout<<"\n\nPlease Enter The New Details:"<<endl;
        d.DoctorInfo();
        int pos=(-1)*static_cast<int>(sizeof(d));
        File.seekp(pos,ios::cur);
        File.write(reinterpret_cast<char *> (&d), sizeof(Doctor));
        cout<<"\n\n\t Record Updated";
        found=true;
    }
}
File.close();
if(found==false)
cout<<"\n\n Record Not Found ";
cin.ignore();
cin.get();
}

void Doctor_choice()
{
    char ch;

    do
    {
int n;
system("cls");

        cout<<"\n\n\t01. Prescribe Medicine for patient ";
        cout<<"\n\n\t02. Search DOCTOR ";
    }
}

```

```

cout<<"\n\n\t03. To exit.....";
    cout<<"\n\n\tPlease Select Your Option (1-3) ";
    cin>>ch;

switch(ch)
{
case '1': Prescribe_Medicine();
    break;
case '2':
    cout<<"Enter Doctor ID :";cin>>n;
    display_d(n);
    break;
    case '3':break;
default :cout<<"\b";
}
    }while(ch!='3');

}

void Doctor_menu()
{
    char ch;

do
{
int n;
system("cls");

cout<<"\n\n\t01. INSERT DOCTOR DATA";
cout<<"\n\n\t02. Search DOCTOR ";
cout<<"\n\n\t03. ALL DOCTOR ";
cout<<"\n\n\t04. MODIFY DOCTOR DATA";
cout<<"\n\n\t05. TO EXIT.....";
cout<<"\n\n\tPlease Select Your Option (1-5) ";
cin>>ch;
    system("cls");
switch(ch)
{
case '1': write_doctor();
    break;
case '2':
    cout<<"Enter Doctor ID :";cin>>n;
    display_d(n);
    break;

```

```

case '3':

    display_all_Doctor();
    break;
case '4':
    cout<<"Enter Doctor ID :";cin>>n;
    modify_doctor(n);
    break;
/*case '5':break;
default :cout<<"\b";*/
}
    }while(ch!='5');

}

};

class Nurse: public Staff
{

    int NurseID;
    char ChargeRoom [50];
    patient p;
public:
    void GetNurseInfo()
    {

        cout<<"-----"<<endl;
        staff_info();
        cout<<"Enter Nurse ID:";
        cin>>NurseID;
        cout<<"Enter Nurse Duty Room :";
        cin>>ChargeRoom;
    }

    void Display_NurseInfo()
    {
        cout<<"-----"<<endl;
        cout<<"Nurse ID :"<<NurseID;
        show_staff();

        cout<<"\n"<<"Nurse in charge Room :"<<ChargeRoom;
        cout<<endl;

    }

```

```

        void Write_NurseInfo()
    {
        Nurse n;
        ofstream outFile;
        outFile.open("NurseInfo.dat",ios::binary|ios::app);
        n.GetNurseInfo();

        outFile.write(reinterpret_cast<char *> (&n), sizeof(Nurse));
        outFile.close();
        cout<<"\n\nrecord Has Been Created ";
        cin.ignore();
        cin.get();
    }

    void display_n(int m)
    {
        Nurse n;
        ifstream inFile;
        inFile.open("NurseInfo.dat",ios::binary);
        if(!inFile)
        {
            cout<<"File could not be open !! Press any Key...";
            cin.ignore();
            cin.get();
            return;
        }
        bool flag=false;
        while(inFile.read(reinterpret_cast<char *> (&n), sizeof(Nurse)))
        {
            if(n.NurseID==m)
            {
                n.Display_NurseInfo();
                flag=true;
            }
        }
        inFile.close();
        if(flag==false)
            cout<<"\n\nrecord does not exist";
        cin.ignore();
        cin.get();
    }

    void modify_Nurse(int m)
    {
        bool found=false;
        Nurse n;
        fstream File;
    }

```

```

File.open("NurseInfo.dat",ios::binary|ios::in|ios::out);
if(!File)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}

while(!File.eof() && found==false)
{

File.read(reinterpret_cast<char *> (&n), sizeof(Nurse));
if(n.NurseID==m)
{
n.Display_NurseInfo();
cout<<"\n\nPlease Enter The New Details of nurse"<<endl;
n.GetNurseInfo();
int pos=(-1)*static_cast<int>(sizeof(n));
File.seekp(pos,ios::cur);
File.write(reinterpret_cast<char *> (&n), sizeof(Nurse));
cout<<"\n\n\t Record Updated";
found=true;
}
}
File.close();
if(found==false)
cout<<"\n\n Record does Not Found ";
cin.ignore();
cin.get();
}

void display_all_Nurse()
{
Nurse n;
ifstream inFile;
inFile.open("NurseInfo.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
while(inFile.read(reinterpret_cast<char *> (&n), sizeof(Nurse)))
{
n.Display_NurseInfo();

```



```

cout<<"\n\n=====
=====\\n";
}
inFile.close();
cin.ignore();
cin.get();
}

void Nurse_menu()
{
    char ch;

do
{
    int n;
    system("cls");

    cout<<"\n\n\t01. INSERT NURSE DATA";
    cout<<"\n\n\t02. Search NURSE ";
    cout<<"\n\n\t03. ALL NURSE ";
    cout<<"\n\n\t04. MODIFY NURSE DATA";
    cout<<"\n\n\t05. To EXIT.....";
    cout<<"\n\n\tPlease Select Your Option (1-5) ";
    cin>>ch;

    switch(ch)
    {
    case '1': Write_NurseInfo();
        break;
    case '2':
        cout<<"Enter NURSE ID :";cin>>n;
        display_n(n);
        break;

    case '3':

        display_all_Nurse();
        break;
    case '4':
        cout<<"Enter Nurse ID :";cin>>n;
        modify_Nurse(n);
        break;

    }

    }while(ch!='5');

}

```

```
};
```

```
class Receptionist:public Staff  
{
```

```
    patient p;  
    int RecID;
```

```
public:
```

```
    void RecInfo()
```

```
    {  
        staff_info();  
        cout<<"Enter Receptionist ID :";  
        cin>>RecID;
```

```
    }
```

```
    void show_receptionist()
```

```
    {  
  
        show_staff();
```

```
        cout<<"Receptionist ID: "<<RecID;
```

```
    }
```

```
void entry_Patient()
```

```
{
```

```
char ch;
```

```
int num;
```

```
do{
```

```
system("cls");
```

```
cout<<"\n\n\n\tENTRY MENU";
```

```
cout<<"\n\n\t1.CREATE PATIENT RECORD";
```

```
cout<<"\n\n\t2.DISPLAY ALL PATIENTS RECORDS";
```

```
cout<<"\n\n\t3.SEARCH PATIENT RECORD ";
```

```
cout<<"\n\n\t4.MODIFY PATIENT RECORD";
```

```
cout<<"\n\n\t5.DELETE PATIENT RECORD";
```

```

cout<<"\n\n\t6.TO EXIT";
cout<<"\n\n\tPlease Enter Your Choice (1-6) ";
cin>>ch;
system("cls");

switch(ch)
{
case '1': p.write_patient();break;
case '2': p.display_all(); break;
case '3': cout<<"\n\n\tPlease Enter The reg  number "; cin>>num;
        p.display_p(num);break;
case '4': cout<<"\n\n\tPlease Enter The reg number "; cin>>num;
        p.modify_patient(num);break;
case '5': cout<<"\n\n\tPlease Enter The reg number "; cin>>num;
        p.delete_patient(num);break;

}
}while(ch!='6');
}

void write_receptionist()
{
Receptionist r;
ofstream outFile;
outFile.open("Receptionist.dat",ios::binary|ios::app);
r.RecInfo();
outFile.write(reinterpret_cast<char *> (&r),
sizeof(Receptionist));
outFile.close();
        cout<<"\n\nReceptionist record Has Been Created ";
cin.ignore();
cin.get();
}

void display_all_Receptionist()
{
Receptionist r;
ifstream inFile;
inFile.open("Receptionist.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
}

```

```

cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
while(inFile.read(reinterpret_cast<char *> (&r),
sizeof(Receptionist)))
{
r.show_receptionist();
cout<<"\n\n=====
=====\\n";
}
inFile.close();
cin.ignore();
cin.get();
}

```

```

void display_r(int n)
{
Receptionist r;
ifstream inFile;
inFile.open("Receptionist.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
bool flag=false;
while(inFile.read(reinterpret_cast<char *> (&r),
sizeof(Receptionist)))
{
if(r.RecID==n)
{
r.show_receptionist();
flag=true;
}
}
inFile.close();
if(flag==false)
cout<<"\n\nrecord not exist";
cin.ignore();
cin.get();
}

```

```

void modify_receptionist(int n)
{
bool found=false;
Receptionist r;

```

```

fstream File;
File.open("Receptionist.dat",ios::binary|ios::in|ios::out);
if(!File)
{
    cout<<"File could not be open !! Press any Key...";
    cin.ignore();
    cin.get();
    return;
}

    while(!File.eof() && found==false)
    {

File.read(reinterpret_cast<char *> (&r), sizeof(Receptionist));
if(r.RecID==n)
{
    r.show_receptionist();
    cout<<"\n\nPlease Enter The New Details "<<endl;
        r.RecInfo();
        int pos=(-1)*static_cast<int>(sizeof(r));
        File.seekp(pos,ios::cur);
        File.write(reinterpret_cast<char *> (&r),
sizeof(Receptionist));
        cout<<"\n\n\t Record Updated";
        found=true;
    }
}
File.close();
if(found==false)
cout<<"\n\n Record Not Found ";
cin.ignore();
cin.get();
}

void Receptionist_menu()
{
    char ch;

do
{
    int n;
    system("cls");

    cout<<"\n\n\t01. INSERT Receptionist DATA";
    cout<<"\n\n\t02. Search Receptionist ";
    cout<<"\n\n\t03. ALL Receptionist ";
    cout<<"\n\n\t04. MODIFY Receptionist DATA";

```

```

cout<<"\n\n\t05. To EXIT.....";
cout<<"\n\n\tPlease Select Your Option (1-5) ";
cin>>ch;

switch(ch)
{
case '1': write_receptionist();
        break;
case '2':
        cout<<"Enter Receptionist ID :";cin>>n;
        display_r(n);
        break;

case '3':

        display_all_Receptionist();
        break;
case '4':
        cout<<"Enter Receptionist ID :";cin>>n;
        modify_receptionist(n);
        break;

}

}while(ch!='5');

}

};

```

```

class store_manager:public Staff
{

private:
    string productName;
    string supplierName;
    int unit;
    int bill;
    int SID;

public:

    void SInfo()
    {
        staff_info();
        cout<<"Enter Receptionist ID :";
        cin>>SID;
    }
}

```

```

    }

    void show_StoreManager()
    {
        show_staff();
        cout<<"Store Manager ID: "<<SID;

    }

    void orderproducts()
    {
        cout<<"\n\n|----- ORDER Products & SUPPLIES
-----|"<<endl<<endl;

        cout<<"Enter Product Name: ";
        cin>>productName;
        cout<<"\nEnter Supplier's Name: ";
        cin>>supplierName;
        cout<<"\nEnter Number of Unit: ";
        cin>>unit;
        cout<<"\nEnter Total bill: ";
        cin>>bill;
        ofstream fout;
        fout.open("Products&Supplies.txt",ios::app);
        fout<<"Product Name: "<< productName<<endl;
        fout<<"Supplier's Name: "<< supplierName<<endl;
        fout<<"Number of Unit: "<< unit<<endl;
        fout<<"Total Bill: "<< bill<<endl<<endl;
        fout.close();
        cout<<"\n\nPRODUCT ORDERED SUCCESSFULLY!!"<<endl<<endl;
        int choice;
        cout<<"Do you want to order more?"<<endl;
        cout<<"(1) Yes"<<endl;
        cout<<"(2) No"<<endl<<endl<<endl;
        cin>>choice;

        if(choice==1)
        {
            system("cls");
            orderproducts();
        }
        else
        {
            return;
        }
    }
}

```

```

void readFromFile()
{
    ifstream fin;
    fin.open("Products&Supplies.txt");

    string str;
    if(!fin)
    {
        cout<<"\nERROR LOADING FILE!!"<<endl;
        return;
    }
    else
    {
        while(!fin.eof())
        {
            getline(fin,str);
            cout<<str;
            cout<<endl;
        }
    }
    cout<<endl<<endl<<endl;
    fin.close();
    cin.get();
    system("pause");
}

```

```

void write_store_manager()
{
    store_manager s;
    ofstream outFile;
    outFile.open("StoreManager.dat",ios::binary|ios::app);
    s.SInfo();
    outFile.write(reinterpret_cast<char *> (&s),
    sizeof(store_manager));
    outFile.close();
    cout<<"\n\nrecord Has Been Created ";
    cin.ignore();
    cin.get();
}

```

```

void display_all_storeManager()
{
    store_manager s;
    ifstream inFile;
    inFile.open("StoreManager.dat",ios::binary);

```



```

if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
while(inFile.read(reinterpret_cast<char *> (&s),
sizeof(store_manager)))
{
s.show_StoreManager();
cout<<"\n\n=====
=====\\n";
}
inFile.close();
cin.ignore();
cin.get();
}

void display_s(int n)
{
store_manager s;
ifstream inFile;
inFile.open("StoreManager.dat",ios::binary);
if(!inFile)
{
cout<<"File could not be open !! Press any Key...";
cin.ignore();
cin.get();
return;
}
bool flag=false;
while(inFile.read(reinterpret_cast<char *> (&s),
sizeof(store_manager)))
{
if(s.SID==n)
{
s.show_StoreManager();
flag=true;
}
}
inFile.close();
if(flag==false)
cout<<"\n\nrecord not exist";
cin.ignore();
cin.get();
}

```

```

}

void store_manager_choice()
{
    char ch;

    do
    {
        int n;
        system("cls");

        cout<<"\n\n\t01. Order Products ";
        cout<<"\n\n\t02. Show Order Products & Bills ";
        cout<<"\n\n\t03. To exit.....";
        cout<<"\n\n\tPlease Select Your Option (1-3) ";
        cin>>ch;

        switch(ch)
        {
            case '1':
                orderproducts();
                break;
            case '2':
                readFromFile();
                break;
            case '3':break;
            default :cout<<"\b";
        }
        }while(ch!='3');

    }

void store_manager_menu()
{
    char ch;

    do
    {
        int n;
        system("cls");

        cout<<"\n\n\t01. INSERT STORE MANAGER DATA";
        cout<<"\n\n\t02. Search STORE MANAGER ";
        cout<<"\n\n\t03. ALL STORE MANAGER ";
    }

```

```

cout<<"\n\n\t04. To exit this menu";
cout<<"\n\n\tPlease Select Your Option (1-4) ";
cin>>ch;

switch(ch)
{
case '1': write_store_manager();
        break;
case '2':
        cout<<"Enter STORE MANAGER ID :";cin>>n;
        display_s(n);
        break;

case '3':

        display_all_storeManager();
        break;

case '4':break;
default :cout<<"\b";
}
        }while(ch!='4');

}

};

```

```

class admin
{
    Doctor d;
    patient p;
    Receptionist r;
    store_manager s;
    Nurse n;

    public:
        void staff_menu()
        {
            char ch;
            do{
system("cls");
cout<<"\n\n\tStaff Menu";
cout<<"\n\n\t1.Doctor";
cout<<"\n\n\t2.Nurse";
cout<<"\n\n\t3.Receptionist";
cout<<"\n\n\t4.Store Manger";

```

```

cout<<"\n\n\t5.TO EXIT";
cout<<"\n\n\tPlease Enter Your Choice (1-4) ";
cin>>ch;
system("cls");

switch(ch)
{
    case '1': d.Doctor_menu();break;
    case '2': n.Nurse_menu(); break;
    case '3':    r.Receptionist_menu();break;
    case '4':    s.store_manager_menu();break;

}
}while(ch!='5');

    }

};
void fordelay(int j)
{    int i,k;
    for(i=0;i<j;i++)
        k=i;
}

int main()
{
    system("color f5");
    admin a;
    Receptionist r;
    Doctor d;
    store_manager s;
    int ch;
    char pass[15];
    int n;
    cout << "\n\n\n\n\n\n\t\t\t\tWELCOME TO OUR HOSPITAL
MANAGEMENT SYSTEM";
    for(int i=0;i<6;i++)
    {
        fordelay(100000000);
        printf(".");
    }
    do{

        fomo:
            system("cls");
        cout<<"\t\t\n\n\n";
        cout<<"\t\t\t\t\tSELECT AN OPTION\n\n";

```

```

cout<<"\t\t\t\t1.Admin\n";
cout<<"\t\t\t\t2.Receptionist\n";
cout<<"\t\t\t\t3.Doctor\n";
cout<<"\t\t\t\t4.Store Manager\n";
cout<<"\t\t\t\t5.Exit From The SYSTEM\n";
cin>>ch;
switch(ch){
    case 1:
        login1:
        cout<<"Enter ADMIN password: ";

        cin>>pass;

        if(strcmp(pass,"admin")==0)
        {
            a.staff_menu();
        }
        else
        {
            cout<<"WRONG PASSWORD!!!\n";
            cout<<"PRESS 1 TO TRY AGAIN OR 5 TO GO BACK
TO MENU\n";

            cin>>n;
            if(n==1)
                goto login1;

            else
                goto fomo;
        }
        break;
    case 2:
        login2:
        cout<<"Enter Receptionist password: ";

        cin>>pass;

        if(strcmp(pass,"receptionist")==0)
        {
            r.entry_Patient();
        }
        else
        {
            cout<<"WRONG PASSWORD!!!\n";
            cout<<"PRESS 1 TO TRY AGAIN OR 5 TO GO BACK
TO MENU\n";

```

```

        cin>>n;
        if(n==1)
            goto login2;

        else
            goto fomo;
    }
    break;
case 3:

    login3:
    cout<<"Enter Doctor password: ";

    cin>>pass;

    if(strcmp(pass,"doctor")==0)
    {
        d.Doctor_choice();
    }
    else
    {
        cout<<"WRONG PASSWORD!!!\n";
        cout<<"PRESS 1 TO TRY AGAIN OR 5 TO GO BACK
TO MENU\n";

        cin>>n;
        if(n==1)
            goto login3;

        else
            goto fomo;
    }
    break;
case 4:

    login4:
    cout<<"Enter STORE MANAGER password: ";

    cin>>pass;
    if(strcmp(pass,"store")==0)
    {
        s.store_manager_choice();
    }
    else
    {
        cout<<"WRONG PASSWORD!!!\n";
        cout<<"PRESS 1 TO TRY AGAIN OR 5 TO GO BACK
TO MENU\n";

```

```
        cin>>n;
        if(n==1)
            goto login4;

        else
            goto fomo;
    }
    break;
}
}while(ch!=5);
}
```

OUTPUT AND EXPLANATION

Welcome Screen:



WELCOME TO OUR HOSPITAL MANAGEMENT SYSTEM..

Main menu:



SELECT AN OPTION

- 1.Admin
- 2.Receptionist
- 3.Doctor
- 4.Store Manager
- 5.Exit From The SYSTEM

Log-In as Admin: Ensuring the correct password is entered!

```

1
1.Admin
2.Receptionist
3.Doctor
4.Store Manager
5.Exit From The SYSTEM
1
Enter ADMIN password: password
WRONG PASSWORD!!!
PRESS 1 TO TRY AGAIN OR 5 TO GO BACK TO MENU
1
Enter ADMIN password: admin
```

ADMIN MENU: The options visible to the Admin of the Management System

```

Staff Menu

1.Doctor

2.Nurse

3.Receptionist

4.Store Manger

5.TO EXIT

Please Enter Your Choice (1-4) |
```

ADMIN MENU: STAFF MENU - 1. DOCTOR

```

01. INSERT DOCTOR DATA

02. Search DOCTOR

03. ALL DOCTOR

04. MODIFY DOCTOR DATA

05. TO EXIT.....

Please Select Your Option (1-5) |
```

ADMIN MENU: DOCTOR - Insert Doctor Data

```
Enter Name :Ashwin
Enter Gender :Male
Enter Age :20
Enter Address :Mysore
Enter Blood group:A+
Enter salary:10000000
Enter Duty time:From:10AM
To:3PM
Enter Doctor ID(1200-1299):1201
Enter Doctor Specialize :GeneralPhysic
Press any key to continue . . .
```

Doctor ID Can't be repeated:

```
Enter Doctor ID(1200-1299):1203
ID already taken
Enter Doctor ID(1200-1299):1204
```

ADMIN MENU: DOCTOR == 2. SEARCH DOCTOR

```
Enter Doctor ID :1207
Doctor ID: 1207
Salary: 1200000
Duty Time: 9pm--12pm
Name: Janvi
Gender: Female
Age: 19
Address: Gujarat
Blood Group:O+

Doctor Specialize: Surgeon
```

ADMIN MENU: DOCTOR == 3. Display All Doctor Data:

```

                DISPLAY ALL RECORD !!!

Doctor ID: 1201
Salary: 10000000
Duty Time: 10AM--3PM
Name: Ashwin
Gender: Male
Age: 20
Address: Mysore
Blood Group:A+

Doctor Specialize: GeneralPhysic

=====
Doctor ID: 1202
Salary: 2000000
Duty Time: 3pm--7pm
Name: Devansh
Gender: Male
Age: 21
Address: Delhi
Blood Group:B+

Doctor Specialize: GeneralPhysc

=====
Doctor ID: 1203
Salary: 3000000
Duty Time: 7pm--9pm
Name: Sparsh
Gender: Male
Age: 21
Address: Noida
Blood Group:AB+

Doctor Specialize: Surgeon

=====
Doctor ID: 1203
Salary: 3000000
Duty Time: 7pm--9pm
Name: Sparsh
Gender: Male
Age: 21
Address: Noida
Blood Group:AB+

Doctor Specialize: Surgeon

=====
Doctor ID: 1204
Salary: 1200000
Duty Time: 7am--9am
Name: Mohak
Gender: male
Age: 21
Address: Rithala
Blood Group:O+

Doctor Specialize: GeneralPhysc

=====
Doctor ID: 1207
Salary: 1200000
Duty Time: 9pm--12pm
Name: Janvi
Gender: Female
Age: 19
Address: Gujarat
Blood Group:O+

Doctor Specialize: Surgeon

=====
```

ADMIN MENU: DOCTOR == 4. Modify Doctor Data

```
Enter Doctor ID :1207
Doctor ID: 1207
Salary: 1200000
Duty Time: 9pm--12pm
Name: Janvi
Gender: Female
Age: 19
Address: Gujarat
Blood Group:O+

Doctor Specialize: Surgeon

Please Enter The New Details:
Enter Name :Janvi
Enter Gender :Female
Enter Age :19
Enter Address :Bharuch
Enter Blood group:o+
Enter salary:125890
Enter Duty time:From:4pm
To:9pm
Enter Doctor ID(1200-1299):1208
Enter Doctor Specialize :HeartSurgeon
Press any key to continue . . .
```

Record Updated

ADMIN MENU: 2. NURSE

```
01. INSERT NURSE DATA

02. Search NURSE

03. ALL NURSE

04. MODIFY NURSE DATA

05. To EXIT.....

Please Select Your Option (1-5) 1
```

ADMIN MENU: NURSE ==1.INSERT NURSE DATA

```
-----  
Enter Name :Ujjawal  
Enter Gender :Male  
Enter Age :21  
Enter Address :Delhi  
Enter Blood group:O+  
Enter salary:1900000  
Enter Duty time:From:9am  
To:9pm  
Enter Nurse ID:1001  
Enter Nurse Duty Room :B12
```

```
record Has Been Created
```

ADMIN MENU: NURSE == 2. SEARCH NURSE

```
                Please Select Your Option (1-5) 2  
Enter NURSE ID :1002  
-----  
Nurse ID :1002  
Salary: 12000  
Duty Time: 3am--3pm  
Name: Deepanshu  
Gender: male  
Age: 21  
Address: gurgaon  
Blood Group:a+  
  
Nurse in charge Room :b12
```

ADMIN MENU: NURSE == 3. DISPLAY ALL NURSE DATA

DISPLAY ALL RECORD !!!

Nurse ID :1001
Salary: 1900000
Duty Time: 9am--9pm
Name: Ujjawal
Gender: Male
Age: 21
Address: Delhi
Blood Group:O+

Nurse in charge Room :B12

=====

Nurse ID :1003
Salary: 133333
Duty Time: 1am--1pm
Name: Archit
Gender: Male
Age: 22
Address: Muzzafarnagar
Blood Group:a+

Nurse in charge Room :b15

=====

Nurse ID :1002
Salary: 12000
Duty Time: 3am--3pm
Name: Deepanshu
Gender: male
Age: 21
Address: gurgaon
Blood Group:a+

Nurse in charge Room :b12

ADMIN MENU: NURSE == 4. Modify Nurse Data

```

Please Select Your Option (1-5) 4
Enter Nurse ID :5
-----
Nurse ID :5
Salary: 5
Duty Time: 5--5
Name: Deepanshu
Gender: 130000
Age: 3
Address: am
Blood Group:5

Nurse in charge Room :5

Please Enter The New Details of nurse
-----
Enter Name :Deepanshu
Enter Gender :Male
Enter Age :21
Enter Address :Gurgaon
Enter Blood group:A+
Enter salary:1230000
Enter Duty time:From:1am
To:9pm
Enter Nurse ID:1002
Enter Nurse Duty Room :b12

Record Updated
```

ADMIN MENU: RECEPTIONIST == 1. ADD DATA

```

Please Select Your Option (1-5) 1
Enter Name :Anita
Enter Gender :Female
Enter Age :20
Enter Address :Mysore
Enter Blood group:A+
Enter salary:10000
Enter Duty time:From:9am
To:9pm
Enter Receptionist ID :3001

Receptionist record Has Been Created
```

ADMIN MENU: RECEPTIONIST == 2. SEARCH RECEPTIONIST

```
      Please Select Your Option (1-5) 2
Enter Receptionist ID :3001

Salary: 10000
Duty Time: 9am--9pm
Name: Anita
Gender: Female
Age: 20
Address: Mysore
Blood Group:A+
Receptionist ID: 3001
```

ADMIN MENU: RECEPTIONIST == 3. DISPLAY ALL RECORDS

```
      DISPLAY ALL RECORD !!!

Salary: 10000
Duty Time: 9am--9pm
Name: Anita
Gender: Female
Age: 20
Address: Mysore
Blood Group:A+
Receptionist ID: 3001

=====
```


ADMIN MENU: RECEPTIONIST == 4. MODIFY RECORDS

```
Enter Receptionist ID :3001

Salary: 10000
Duty Time: 9am--9pm
Name: Anita
Gender: Female
Age: 20
Address: Mysore
Blood Group:A+
Receptionist ID: 3001

Please Enter The New Details
Enter Name :AnitaJoshi
Enter Gender :Female
Enter Age :21
Enter Address :Mysore
Enter Blood group:A+
Enter salary:12000
Enter Duty time:From:9am
To:9pm
Enter Receptionist ID :3002

Record Updated
```

ADMIN MENU: STORE MANAGER == 1. ADD RECORDS

```
Enter Name :dubey
Enter Gender :male
Enter Age :22
Enter Address :jhansi
Enter Blood group:a+
Enter salary:120000
Enter Duty time:From:9am
To:9pm
Enter Receptionist ID :4001
```

```
record Has Been Created
```

ADMIN MENU: STORE MANAGER == 2. SEARCH BY RECORD

```
Enter STORE MANAGER ID :4001
```

```
Salary: 120000
Duty Time: 9am--9pm
Name: dubey
Gender: male
Age: 22
Address: jhansi
Blood Group:a+
Store Manager ID: 4001
```

ADMIN MENU: STORE MANAGER == 3. DISPLAY ALL RECORDS

```
DISPLAY ALL RECORD !!!
```

```
Salary: 120000
Duty Time: 9am--9pm
Name: dubey
Gender: male
Age: 22
Address: jhansi
Blood Group:a+
Store Manager ID: 4001
```

```
=====
```

RECEPTIONIST MENU::

ENTRY MENU

- 1.CREATE PATIENT RECORD
- 2.DISPLAY ALL PATIENTS RECORDS
- 3.SEARCH PATIENT RECORD
- 4.MODIFY PATIENT RECORD
- 5.DELETE PATIENT RECORD
- 6.TO EXIT

Please Enter Your Choice (1-6)

RECEPTIONIST :: MENU OF RECEPTIONIST

Enter Name :Abhi

Enter Gender :male

Enter Age :21

Enter Address :mysore

Enter Blood group:a+

Enter The registration number of the patient: 5001

Enter The occupation of the patient: business

Enter The bill of the patient:10000

Enter The date admitted(dd/mm/yyyy):12/02/2023

Enter The date discharged(dd/mm/yyyy):13/03/2023

RECEPTIONIST MENU :: BILL OF PATIENT

Patient Fee Charge

Hospital Care Charges	:	50
Hospital Maintaince Charges	:	50
Patient Individual Charges	:	10000
Total Fee Charge	:	10100

Patient record Has Been Created

RECEPTIONIST MENU :: SEARCH PATIENT BY RECORD

Please Enter The reg number 5001

Registration number of patient: 5001

Name: Abhishek

Gender: male

Age: 19

Address: delhi

Blood Group:a+

Occupation : business

Admit date : 12/02/2003

Discharhge date :12/02/2003

Bill :1300

Prescribed Medicine: null

Registration number of patient: 5001

Name: Abhi

Gender: male

Age: 21

Address: mysore

Blood Group:a+

Occupation : business

Admit date : 12/02/2023

Discharhge date :13/03/2023

Bill :10100

Prescribed Medicine: null

RECEPTIONIST MENU :: MODIFY PATIENT BY RECORD

```

Please Enter The reg number 5001
Registration number of patient: 5001
Name: Abhishek
Gender: male
Age: 19
Address: delhi
Blood Group:a+

Occupation : business
Admit date : 12/02/2003
Discharhge date :12/02/2003
Bill :1300
Prescribed Medicine: null

Please Enter The New Details of patient
Enter Name :
```

RECEPTIONIST MENU :: DELETE PATIENT BY RECORD

```

Please Enter The reg number 5001

Record Deleted ..
```

DOCTOR MENU :: MAIN MENU

```

01. Prescribe Medicine for patient
02. Search DOCTOR
03. To exit.....

Please Select Your Option (1-3)
```

DOCTOR MENU :: 1. PRESCRIBE MEDICINE FOR THE PATIENT

```

Please Select Your Option (1-3)
1

Enter Reg Number Of patient : 5001

Registration number of patient: 5001
Name: Abhi
Gender: male
Age: 21
Address: mysore
Blood Group:a+

Occupation : business
Admit date : 23/02/2003
Discharhge date :21/03/2004
Bill :12100
Prescribed Medicine: null

Please Enter The Medicine of patient
ASPIRINE

Record Updated
```

DOCTOR MENU :: SEARCH DOCTOR

```

01. Prescribe Medicine for patient
02. Search DOCTOR
03. To exit.....

Please Select Your Option (1-3) 2
Enter Doctor ID :
1208
Doctor ID: 1208
Salary: 125890
Duty Time: 4pm--9pm
Name: Janvi
Gender: Female
Age: 19
Address: Bharuch
Blood Group:o+

Doctor Specialize: HeartSurgeon
```

STORE MANAGER :: MAIN MENU

```
01. Order Products
02. Show Order Products & Bills
03. To exit.....
Please Select Your Option (1-3)
```

STORE MANAGER :: ORDER PRODUCTS

```
01. Order Products
02. Show Order Products & Bills
03. To exit.....
Please Select Your Option (1-3) 1

|----- ORDER Products & SUPPLIES -----|
Enter Product Name: aspirine
Enter Supplier's Name: vexe
Enter Number of Unit: 24
Enter Total bill: 1200

PRODUCT ORDERED SUCCESSFULLY!!
Do you want to order more?
(1) Yes
(2) No
```

STORE MANAGER :: SHOW ORDERED PRODUCTS AND BILLS

```
Please Select Your Option (1-3) 2
Product Name: aspirine
Supplier's Name: vexe
Number of Unit: 24
Total Bill: 1200

Press any key to continue . . .
```

