MINI PROJECT REPORT

On

“HOSPITAL DATA BASE”

A Report Submitted for A mini project for OOP and CG lab in 3rd Semester of SecondYear Computer Engineering.

SecondYear (COMPUTER ENGINEERING)
Academic Year 2023-24

Submitted by -

NO.	NAME	ROLL NO
1.	**AARYA VISHWAS DHOPE**	**2101027**
2.	**GAURAV DEVIDAS DIGHE**	**2101028**
3.	**CHINMAY MUKUND DUSE**	**2101029**
4.	**ADITI RAVINDRA GAIKWAD**	**2101030**

SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE

Narhe , Pune – 411041

Department of Computer Engineering

SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE

Narhe , Pune – 411041

Department of Computer Engineering

CERTIFICATE

Certified that the project entitled “**HOSPITAL DATA BASE**” is a bonafide work carried out by Aarya Vishwas Dhope,Gaurav Devidas Dighe, Chinmay Mukund Duse, Aditi Ravindra Gaikwad It is certified that all corrections/suggestions indicated for Internal Assignment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

Prof. Vidya Shinde **Dr.G.S.Navale**

H. O. D.

Project Guide

ACKNOWLEDGEMENT

We take this opportunity to thank our project guide **Prof. Vidya Shinde** and Head of the department **Dr.G.S.Navale** for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of Computer Engineering the Department for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities,Internet access and important books.

INDEX

ABSTRACT

We created a hospital data base using various things i.e. Class, Encapsulation, Inheritance, Polymorphism and Pointer, functions and by using the features implemented features of stack using stack header file in the C++ programing language

SOFTWARE AND HARDWARE REQUIREMENT

- ❖ Ubuntu (Linux OS)
- ❖ Programing languages: ○ C++
- ❖ Software: GNU G++
- ❖ System: Ubuntu 20.04 LTS

Introduction

Creating a hospital database using C++ is a comprehensive and vital endeavor in the realm of healthcare informatics. In this project, the aim is to design a robust system that efficiently manages and organizes crucial information within a hospital environment. Leveraging the power of C++, the programming language renowned for its versatility and performance, we embark on constructing a database that seamlessly handles patient records, medical histories, staff details, and other pertinent data. Through careful implementation of object-oriented principles, we ensure modularity and maintainability, allowing for easy scalability and adaptability to the dynamic nature of healthcare systems. This project not only involves data storage but also incorporates features such as search functionalities, update mechanisms, and security measures to safeguard sensitive information, contributing to the overall efficiency and effectiveness of hospital management.

Problem Statement: Design and implement using Object oriented Programming.

Objective: To understand the concept of OOP and the functions of vector header file and designing mini project.

Outcome: Implementing *hospital data base to store and display the data of patients.*

Implementation Code (C++) :

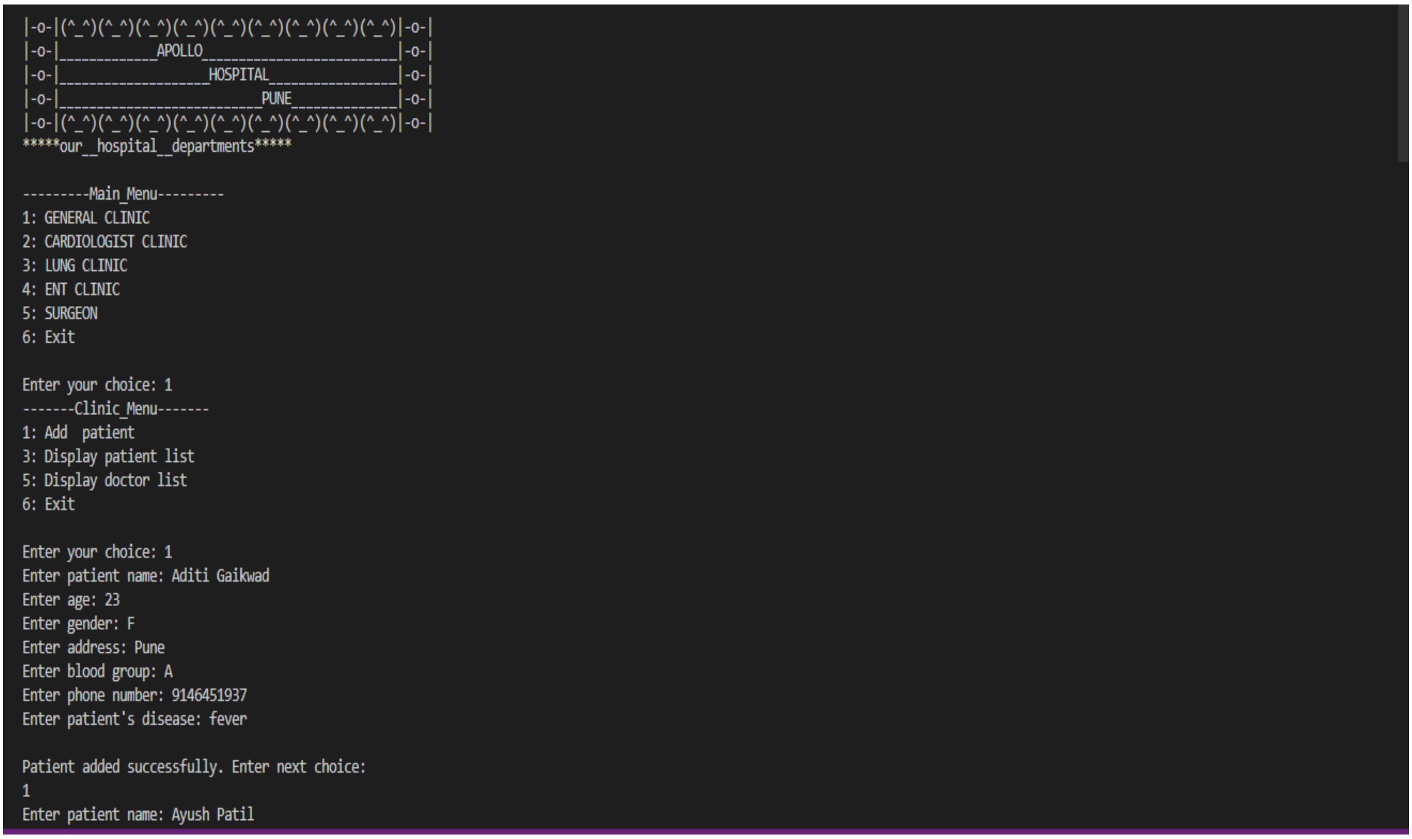
```
#include <iostream>
#include <vector>
#include <string>
#include <iomanip>
#include <limits>

using namespace std;

class Person {
public:
    string name;
    int age;
    char gender;
    double phone;
    string disease;

    // Default constructor
    Person() : name(""), age(0), gender(' '), phone(0.0), disease("") {}
};

class Doctor : public Person {
```

```
public:
    string type;

    Doctor() : type("") {}

    Doctor(const string& docName, double docPhone, const string& docType)
        : type(docType) {
        name = docName;
        phone = docPhone;
    }

    void displayDoctorInfo() {
        cout << "Doctor Name: " << name << "\nPhone: " << phone << "\nSpecialization: " << type <<
endl;
    }

    // Add a function to check if the doctor treats a specific disease
    bool treatsDisease(const string& disease) const {
        return type == disease;
    }
};

class Patient : public Person {
public:
    string address;
    string bloodgrp;
    string assignedDoctorName;  // Updated to store the assigned doctor's name
    Patient() :  address(""), bloodgrp(""), assignedDoctorName("") {}
};

class Clinic {
public:
    vector<Patient> patients;
    vector<Doctor> doctors;

    void addPatient(bool isCritical) {
        Patient p;
        cin.ignore();
        cout << "Enter patient name: "; getline(cin, p.name);
        cout << "Enter age: "; cin >> p.age;
        cout << "Enter gender: "; cin >> p.gender;
        cin.ignore();
        cout << "Enter address: "; getline(cin, p.address);
        cout << "Enter blood group: "; cin >> p.bloodgrp;

        // Validate phone number input
        do {
            cout << "Enter phone number: ";
            if (!(cin >> p.phone) || to_string(static_cast<long long>(p.phone)).length() != 10) {
                cout << "Invalid phone number. Please enter a 10-digit phone number." << endl;
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
            } else {
                break;
```

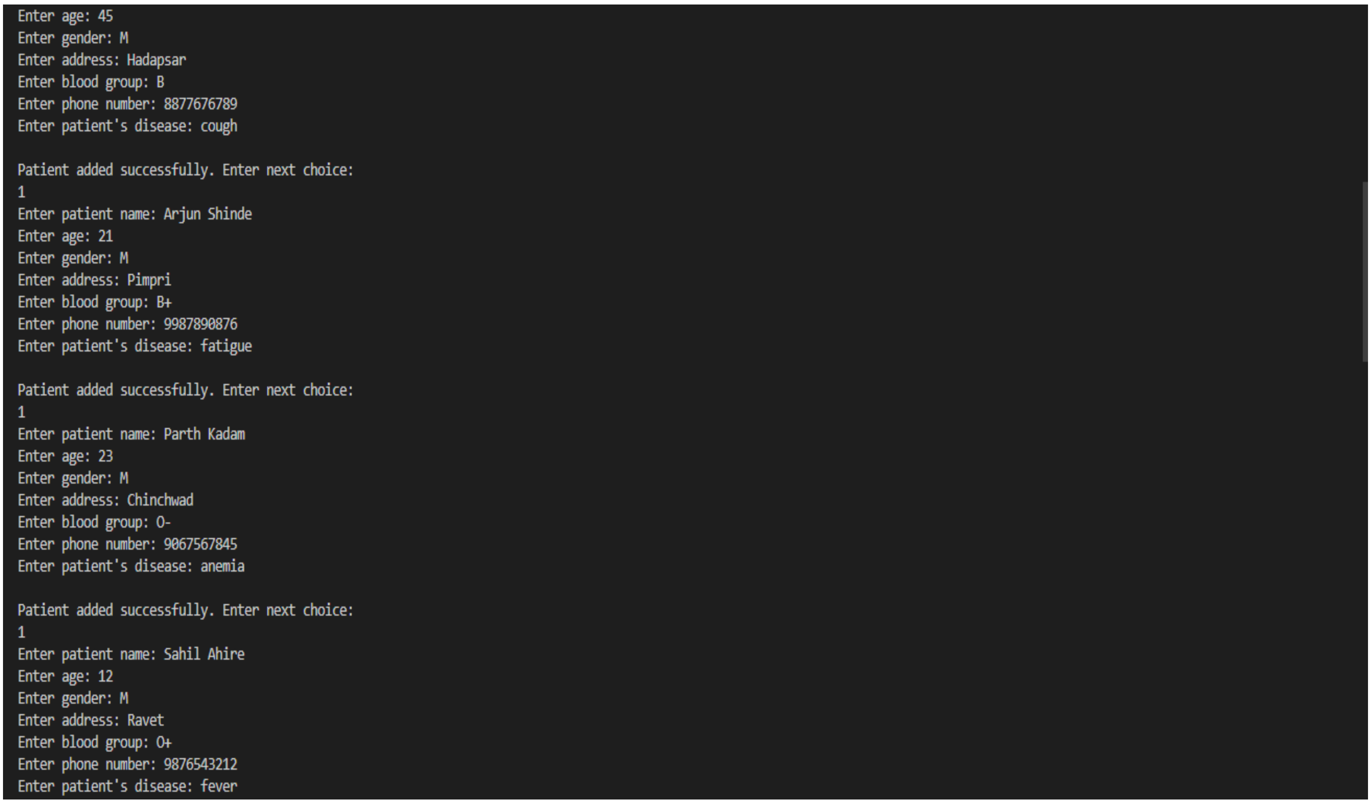
```
         }
      } while (true);

      cout << "Enter patient's disease: "; cin >> p.disease;
      patients.push_back(p);
      cout << endl << "Patient added successfully. Enter next choice:\n";
   }
   void displayPatients() {
   cout << "Patient List:\n";
   cout << left << setw(20) << "Name" << setw(5) << "Age" << setw(20) << "Address"
       << setw(8) << "Gender" << setw(15) << "Phone" << setw(15) << "Blood Group" << endl;
   cout << setfill('-') << setw(100) << "-" << setfill(' ') << endl;

   cout << fixed << setprecision(0);
   for (const auto& patient : patients) {
      cout << left << setw(20) << patient.name
          << setw(5) << patient.age << setw(20) << patient.address
          << setw(8) << patient.gender << setw(15) << patient.phone << setw(15) << patient.bloodgrp
<< endl;
   }
   cout.unsetf(ios::fixed);
   cout<<" DATA DISPLAYED IN TABLE,ENTER NEXT CHOICE\n";
}

   void displayDoctors() {
      cout << "Doctor List:\n";
      cout << left << setw(20) << "Name" << setw(15) << "Phone" << setw(20) << "Specialization" <<
endl;
      cout << setfill('-') << setw(55) << "-" << setfill(' ') << endl;

      cout << fixed << setprecision(0);
      for (const auto& doctor : doctors) {
         cout << left << setw(20) << doctor.name << setw(15) << doctor.phone << setw(20) <<
doctor.type << endl;
      }
      cout.unsetf(ios::fixed);
      cout<<endl;
      cout<<" DATA DISPLAYED IN TABLE,ENTER NEXT CHOICE\n";

   }




};

int main() {
   vector<Clinic> clinics{
      Clinic(), // General Clinic
      Clinic(), // Cardiologist Clinic
      Clinic(), // Lung Clinic
      Clinic(), // ENT Clinic
      Clinic()   // Surgeon
   };
```

```
// Directly declare 5 doctors
clinics[0].doctors.push_back(Doctor("Dr. Smith", 1234567890, "General Medicine"));
clinics[1].doctors.push_back(Doctor("Dr. Johnson", 1234567891, "Cardiology"));
clinics[2].doctors.push_back(Doctor("Dr. Wilson", 1234567892, "Pulmonology"));
clinics[3].doctors.push_back(Doctor("Dr. Brown", 1234567893, "ENT"));
clinics[4].doctors.push_back(Doctor("Dr. Davis", 1234567894, "Surgery"));

int mainChoice;
int clinicChoice;

cout << endl << "|-o-|(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)|-o-|\n";
cout << "|-o-|______________APOLLO____________________________|-o-|\n";
cout << "|-o-|______________________HOSPITAL__________________|-o-|\n";
cout << "|-o-|_____________________________PUNE_______________|-o-|\n";
cout << "|-o-|(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)|-o-|\n";

do {
    cout << "*****our__hospital__departments*****\n";
    cout << endl << "---------Main_Menu---------\n";
    cout << "1: GENERAL CLINIC\n";
    cout << "2: CARDIOLOGIST CLINIC\n";
    cout << "3: LUNG CLINIC\n";
    cout << "4: ENT CLINIC\n";
    cout << "5: SURGEON\n";
    cout << "6: Exit\n\n";
    cout << "Enter your choice: "; cin >> mainChoice;

    if (mainChoice >= 1 && mainChoice <= 5) {
        Clinic& selectedClinic = clinics[mainChoice - 1];

        cout << "-------Clinic_Menu-------\n";
            cout << "1: Add  patient\n";
            cout << "3: Display patient list\n";
            cout << "5: Display doctor list\n";
            cout << "6: Exit\n\n";
            cout << "Enter your choice: ";

        do {
            cin >> clinicChoice;

            switch (clinicChoice) {
            case 1: selectedClinic.addPatient(false); break;

            case 3: selectedClinic.displayPatients(); break;
            case 5: selectedClinic.displayDoctors(); break;
            case 6: break; // Exit clinic menu
            default: cout << "Invalid choice. Please enter a valid option.\n";
            }

        } while (clinicChoice != 6);
    }
    else if (mainChoice != 6) {
        cout << "Invalid choice. Please enter a valid option.\n";
        }

    } while (mainChoice != 6);

    cout << "Exiting Hospital Management System. Goodbye!\n";

    return 0;
}
```

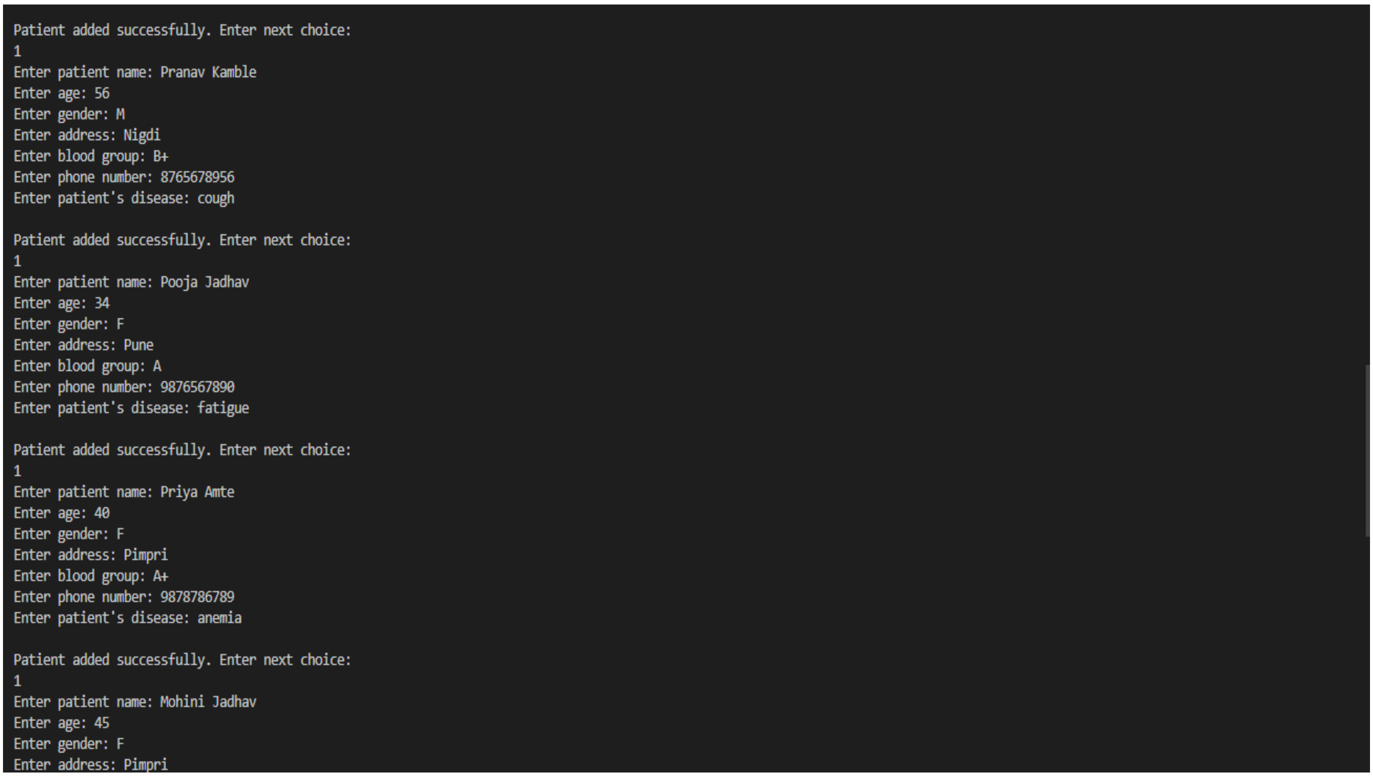
Output:



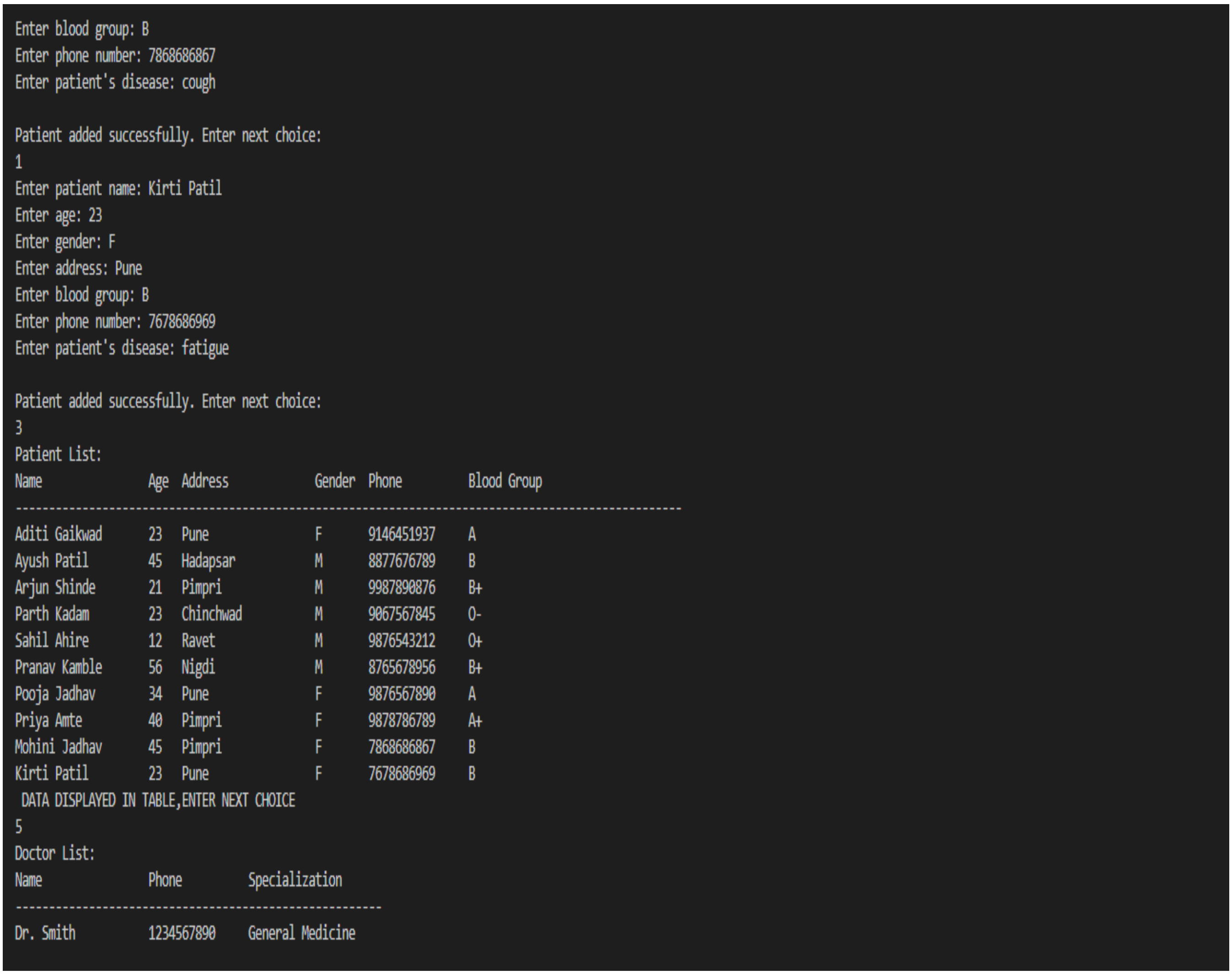
```
|-o-|(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)|-o-|
|-o-|_____________APOLLO__________________________|-o-|
|-o-|____________________HOSPITAL_________________|-o-|
|-o-|___________________________PUNE______________|-o-|
|-o-|(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)(^_^)|-o-|
*****our__hospital__departments*****

---------Main_Menu---------
1: GENERAL CLINIC
2: CARDIOLOGIST CLINIC
3: LUNG CLINIC
4: ENT CLINIC
5: SURGEON
6: Exit

Enter your choice: 1
-------Clinic_Menu-------
1: Add  patient
3: Display patient list
5: Display doctor list
6: Exit

Enter your choice: 1
Enter patient name: Aditi Gaikwad
Enter age: 23
Enter gender: F
Enter address: Pune
Enter blood group: A
Enter phone number: 9146451937
Enter patient's disease: fever

Patient added successfully. Enter next choice:
1
Enter patient name: Ayush Patil
```



```
Enter age: 45
Enter gender: M
Enter address: Hadapsar
Enter blood group: B
Enter phone number: 8877676789
Enter patient's disease: cough

Patient added successfully. Enter next choice:
1
Enter patient name: Arjun Shinde
Enter age: 21
Enter gender: M
Enter address: Pimpri
Enter blood group: B+
Enter phone number: 9987890876
Enter patient's disease: fatigue

Patient added successfully. Enter next choice:
1
Enter patient name: Parth Kadam
Enter age: 23
Enter gender: M
Enter address: Chinchwad
Enter blood group: O-
Enter phone number: 9067567845
Enter patient's disease: anemia

Patient added successfully. Enter next choice:
1
Enter patient name: Sahil Ahire
Enter age: 12
Enter gender: M
Enter address: Ravet
Enter blood group: O+
Enter phone number: 9876543212
Enter patient's disease: fever
```



```
Patient added successfully. Enter next choice:
1
Enter patient name: Pranav Kamble
Enter age: 56
Enter gender: M
Enter address: Nigdi
Enter blood group: B+
Enter phone number: 8765678956
Enter patient's disease: cough

Patient added successfully. Enter next choice:
1
Enter patient name: Pooja Jadhav
Enter age: 34
Enter gender: F
Enter address: Pune
Enter blood group: A
Enter phone number: 9876567890
Enter patient's disease: fatigue

Patient added successfully. Enter next choice:
1
Enter patient name: Priya Amte
Enter age: 40
Enter gender: F
Enter address: Pimpri
Enter blood group: A+
Enter phone number: 9878786789
Enter patient's disease: anemia

Patient added successfully. Enter next choice:
1
Enter patient name: Mohini Jadhav
Enter age: 45
Enter gender: F
Enter address: Pimpri
```



```
Enter blood group: B
Enter phone number: 7868686867
Enter patient's disease: cough

Patient added successfully. Enter next choice:
1
Enter patient name: Kirti Patil
Enter age: 23
Enter gender: F
Enter address: Pune
Enter blood group: B
Enter phone number: 7678686969
Enter patient's disease: fatigue

Patient added successfully. Enter next choice:
3
Patient List:
Name                Age  Address             Gender  Phone          Blood Group
----------------------------------------------------------------------------------------------------
Aditi Gaikwad       23   Pune                F       9146451937     A
Ayush Patil         45   Hadapsar            M       8877676789     B
Arjun Shinde        21   Pimpri              M       9987890876     B+
Parth Kadam         23   Chinchwad           M       9067567845     O-
Sahil Ahire         12   Ravet               M       9876543212     O+
Pranav Kamble       56   Nigdi               M       8765678956     B+
Pooja Jadhav        34   Pune                F       9876567890     A
Priya Amte          40   Pimpri              F       9878786789     A+
Mohini Jadhav       45   Pimpri              F       7868686867     B
Kirti Patil         23   Pune                F       7678686969     B
 DATA DISPLAYED IN TABLE,ENTER NEXT CHOICE
5
Doctor List:
Name                Phone          Specialization
-------------------------------------------------------
Dr. Smith           1234567890     General Medicine
```

Conclusion:

While developing a system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that will generate the proper system cases.

While making the system an eye has been kept on making it as user friendly as such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in case of any system development process where are number of short comings, there have been some short comings in the development of this system also.

References:

Website: ***1.GeeksforGeeks C++,***
2.Tutorialspoint C++
3.Codecademy

Books***:*** ***1.Object-Oriented Programming with C++" by E. Balagurusamy:***
2.Programming in C++" by Sachin Malhotra, Saurabh Malhotra:.