

A
Research Project Report
ON
“HOSPITAL MANAGEMENT SYSTEM”



CHAUDHARY CHARAN SINGH UNIVERSITY MEERUT
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF
BACHELOR OF COMPUTER APPLICATIONS
BATCH 2022 – 2025

SUBMITTED TO :

MISS. PAYAL SHARMA

SUBMITTED BY:

ASRARUL HAQUE
B.C.A. (VI SEM)
ROLL NO. 220339106022

Project Guide

MISS. PAYAL SHARMA



SHANTI INSTITUTE OF TECHNOLOGY MEERUT

A
Research Project Report
ON
“HOSPITAL MANAGEMENT SYSTEM”



CHAUDHARY CHARAN SINGH UNIVERSITY
MEERUT

IN THE PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR
THE DEGREE OF
BACHELOR OF COMPUTER APPLICATIONS
BATCH 2022 – 2025

SUBMITTED TO :

MISS. PAYAL SHARMA

SUBMITTED BY:

ASRARUL HAQUE
B.C.A. (VI SEM)
ROLL NO. 220339106022

Project Guide

MISS. PAYAL SHARMA



SHANTI INSTITUTE OF TECHNOLOGY MEERUT

CONTENTS

1. CRETIFICATE
2. SELF-CERTIFICATE
3. DECLARATION
4. ACKNOWLEDGEMENT
5. SCOPE OF PROJECT
6. USER REQUIREMENT
7. THEORY ABOUT DATA CENTERED
ARCHITECTURE
8. OBJECT ORIENTED DATA CENTERED
APPROACH
9. PRODUCT RESPECTIVE
10. SYSTEM REQUIREMENT
11. DATA FLOW DIAGRAM
12. HISTORY/INTRODUCTION OF C++
13. COADING
14. OUTPUT
15. LIMITATION
16. CONCLUSION
17. BIBLIOGRAPHY

CRETIFICATE

TO WHOM IT MAY CONCERN

THIS IS TO CERTIFY THAT NAME ASRARUL HAQUE ROLL NO. 220339106022 IS A STUDENT OF BCA 6TH SEM AND HAS SUCCESSFULLY

COMPLETED HIS/HER PROJECT ON “HOSPITAL MANAGEMENT SYSTEM”.

DATE :-

MISS. PAYAL SHARMA

The External Project Viva examination has been held on

Signature of Internal Examiner

Signature of External Examiner

Date : -

Date : -

SELF – CRETIFICATE

This is to certify that the project report work "HOSPITAL MANAGEMENT SYSTEM" for ANAND HOSPITAL MEERUT is done by us is an authentic work carried out for the partial fulfillment of the requirement of the award of the Bachelor of Computer Application under the guidance of MISS. PAYAL SHARMA Victor the matter embodied in this project work has not been submitted earlier for award of any degree/diploma to the best of my knowledge and belief.

Signature of Student

DECLARATION

I ASRARUL HAQUE, student of BCA VITH Semester of SHANTI INSTITUTE OF TECHNOLOGY, MEERUT hereby that the project on “ HOSPITAL MANAGEMENT SYSTEM ” Project Guide MISS PAYAL SHARMA OF BCA DEPARTMENT SIT, MEERUT , is an original and authenticated work done by me. I further declare that it has not been submitted else where by any other person in any of the university for the award of any degree or diploma.

ASRARUL HAQUE

BCA - VITH Sem

ROLL NO. 220339106018

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contribute in the development of this work and who influenced my thinking, behaviour, and acts during the course of study.

I am thankful to Miss. Payal Sharma for this support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings.

I also extend my sincere appreciation to Guide Miss. Payal Sharma of BCA Department SIT, Meerut who provided his valuable suggestions and precious time in accomplishing my project report.

Lastly, I would like to thank the almighty, Parents, Director of the HOD of the institute for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

(ASRARUL HAQUE)

SCOPE OF PROJECT

1.1 IMPROVED SERVICES

An objective of the proposed system is to improve the existing system's services, in order to:

Increase employer and employee satisfaction

Provide the quality

Create a controlled information management environment

1.2 BETTER PERFORMANCE

It is an objective of our proposed system to provide better performance which is managed by providing

Desired flexibility

Fast response

Ability to support changes.

Ability to maintain the quality of services.

1.3 EFFICIENT GUI

One of the objectives of the proposed system is to provide a user friendly GUI, that will enable the user to work on clicks

and make the information transfer possible with the most common I/O devices.

1.4 MORE INFORMATION

An objective of our proposed system is to develop a system that will eliminate the problem of insufficient and incomplete information.

1.5 DECISION SUPPORT

By providing the facility of what-if analysis, we will make a decision support system.

1.6 STRONGER CONTROLS

The proposed system will have stronger controls. Controls like passwords, various levels of user access and option will be added.

1.7 ELIMINATION OF RUNNING COST

We are going to develop the proposed system with an objective to eliminate the running cost by overcoming the continuous stationery expenses, manual storage need, and extra staff in the information resource department of the Organization.

USER REQUIREMENT

In the above proposed system single category of authenticated users are there that is the Hospital staff.

After the product has been delivered the staff will be trained by our development team. We also propose to provide a help document for the patient which will help them in using the product efficiently.

THEORY ABOUT DATA CENTERED ARCHITECTURE:

A Data store resides at the center of this architecture and is accessed frequently by other components that modify info, checks availability of room, allocate room, bill function, create bill or otherwise get function info within the Hospital Management System. A data-centered architecture promotes integrability.

OBJECT ORIENTED DATA CENTERED APPROACH

The components of a system encapsulate data and the operations that must be applied to manipulate data. Communication and coordination between components is accomplished via message passing.

PRODUCT PRESPECTIVE

This project helps to define internal boundaries achieving the appropriate balance between the general and the specific, Hospital management giving a positive and efficient workflow to the staff of the Hospital, thereby helping the business model of the Hospital to expand. In principle the product exposes a simple and general interface and hides the specific and complex implementation based. In our example, we want our Hospital design to hide detail that is too complex or unnecessary, thereby reducing paperwork and increasing efficiency.

Also through this project we aim to reduce operating costs and increase throughput of the staff.

SYSTEM REQUIREMENT

HARDWARE AND SOFTWARE REQUIREMENTS

The platform used is C++. Hence we decided to use Microsoft Visual Studio.

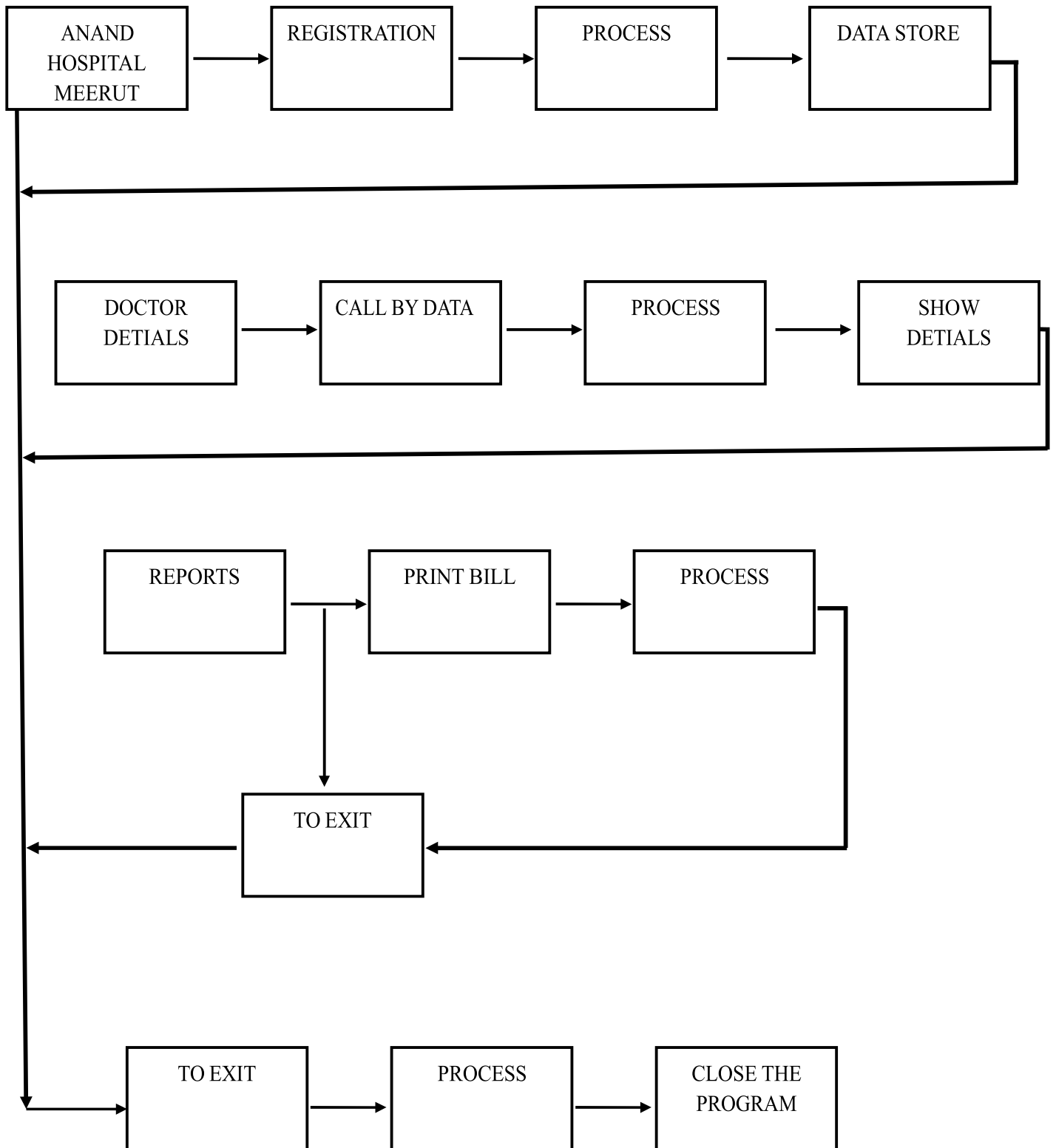
Since BGI graphics aren't supported in the Visual studio so for actually running the software we used Borland C++ compiler 3.0 which was recently made free by Borland Inc.

For optional coding and easy debugging we used Microsoft Visual Studio and made explicit .cpp files.

Henceforth for optimal usage of such a software a windows based operating system preferably Windows vista must be there.

Also on the hardware part any system having Windows Vista installed will suffice.

DATA FLOW DIAGRAM



HISTORY / INTRODUCTION OF C++

C++ is a general-purpose programming language that was developed by Bjarne Stroustrup as an enhancement of the C language to add object-oriented paradigm. It is a highlevel programming language that was first released in 1985 and since then has become the foundation of many modern technologies like game engines, web browsers, operating systems, financial systems, etc.

Features of C++

The main features C++ programming language are as follows:

- **Simple:** It is a simple language in the sense that programs can be broken down into logical units and parts, and has a rich library support and a variety of datatypes.
- **Machine Independent:** C++ code can be run on any machine as long as a suitable compiler is provided.
- **Low-level Access:** C++ provides low-level access to system resources, which makes it a suitable choice for system programming and writing efficient code.
- **Fast Execution Speed:** C++ is one of the fastest high-level languages. There is no additional processing overhead in C++, it is blazing fast.
- **Object-Oriented:** One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. large-scale applications can be built.

OBJECT ORIENTED PROGRAMMING IN C++

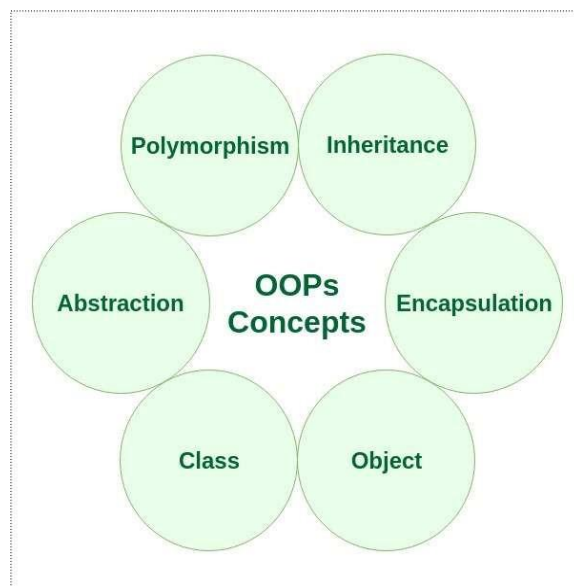
Object Oriented Programming – As the name suggests uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc. in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Characteristics of an Object-Oriented Programming Language

There are some basic concepts that act as the building blocks of OOPs i.e.

Table of Content

- Class
- Object
- Encapsulation
- Abstraction
- Polymorphism
- Inheritance



CLASS

The building
block of

Object-Oriented programming in C++ is a Class. It is a user-defined data type that act as a blueprint representing a group of objects which shares some common properties and behaviours. These properties are stored as data members, and the behaviour is represented by member functions.

For example:- consider the class of Animals. An Animal class could have common properties like name, age, and species as data members, and behaviors like eat, sleep, and makeSound as member functions.

OBJECT

An Object is an identifiable actual entity with some characteristics and behaviour. In C++, it is an instance of a class.

For Example:- , the Animal class is just a concept or category, not an actual entity. But a black cat named VoidShadowDarkFangReaper is actual animal that exists. Similarly, classes are just the concepts and objects are the actual entity that belongs to that concept.

ENCAPSULATION

In simple terms, encapsulation is defined as wrapping up data and information under a single unit. In Object-Oriented Programming, encapsulation is defined as binding together the data and the functions that manipulate them together in a class.

For example:- the Animal class, the data members species, age and name are encapsulated with the member functions like eat(), sleep, etc. They can be protected by the access specifier protected, hides the data of the class from outside.

ABSTRACTION

Abstraction is one of the most essential and important features of object-oriented programming in C++.

Abstraction means displaying only essential information and ignoring the other details.

Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. In our case, when we call the makeSound() method, we don't need to know how the sound is produced internally, only that the method makes the animal sound.

POLYMORPHISM

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of an entity to behave different in different scenarios. person at the same time can have different characteristics.

For example:- the same makeSound() method, the output will vary depending on the type of animal. So, this is an example of polymorphism where the makeSound() method behaves differently depending on the Animal type (Dog or Cat).

In C++, polymorphism can be of three types:

- Operator Overloading: The process of making an operator exhibit different behaviours in different instances is known as operator overloading.
- Function Overloading: Function overloading is using a single function name to perform different types of tasks.
- Function Overriding: Function overriding is changing the behaviour of a function that is derived from the base class using inheritance.

INHERITANCE

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming.

- Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.
- Super Class: The class whose properties are inherited by a sub-class is called Base Class or Superclass.

Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

ADVANTAGE OF OOPS

- We can build the programs from standard working modules that communicate with one another, rather than having to start writing the code from scratch which leads to saving of development time and higher productivity,
- OOP language allows to break the program into the bitsized problems that can be solved easily (one object at a time).
- The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost.
- OOP systems can be easily upgraded from small to large systems.
- It is possible that multiple instances of objects co-exist without any interference,

- It is very easy to partition the work in a project based on objects.
- It is possible to map the objects in problem domain to those in the program.
- The principle of data hiding helps the programmer to build secure programs which cannot be invaded by the code in other parts of the program.
- By using inheritance, we can eliminate redundant code and extend the use of existing classes.
- Message passing techniques is used for communication between objects which makes the interface descriptions with external systems much simpler.
- The data-centered design approach enables us to capture more details of model in an implementable form.

While it is possible to incorporate all these features in an OOP, their importance depends upon the type of project and preference of the programmer. These technology is still developing and current products may be superseded quickly.

Developing a software is easy to use makes it hard to build.

DISADVANTAGES OF OOPS

- The length of the programmes developed using OOP language is much larger than the procedural approach. Since the programme becomes larger in size, it requires more time to be executed that leads to slower execution of the programme.
- We can not apply OOP everywhere as it is not a universal language. It is applied only when it is required. It is not suitable for all types of problems.

- Programmers need to have brilliant designing skill and programming skill along with proper planning because using OOP is little bit tricky.
- OOPs take time to get used to it. The thought process involved in object-oriented programming may not be natural for some people.
- Everything is treated as object in OOP so before applying it we need to have excellent thinking in terms of objects.

COADING

```
#include <iostream>
```

```
#include <vector>
```

```
#include <string>
```

```
using namespace std;
```

```
class Person {
```

```
protected:
```

```
string name;
```

```
int age;    string
```

```
gender; public:
```

```
    Person(string n, int a, string g) : name(n), age(a),
```

```
    gender(g) {}    virtual void display() const = 0;
```

```
};
```

```
class Patient : public Person {
```

```
int patientID;    static int
```

```
idCounter; public:
```

```

    Patient(string n, int a, string g) : Person(n, a, g) {
patientID = ++idCounter;

    }

    void display() const override {    cout << "Patient
ID: " << patientID << ", Name: " << name
        << ", Age: " << age << ", Gender: " << gender <<
endl;

    }

};

```

```

int Patient::idCounter = 0;

```

```

class Doctor : public Person {    string
specialty;

```

```

    int doctorID;    static
int idCounter; public:

    Doctor(string n, int a, string g, string s) : Person(n, a,
g), specialty(s) {        doctorID = ++idCounter;

    }

```

```

        void display() const override {            cout << "Doctor
ID: " << doctorID << ", Name: " << name
        << ", Age: " << age << ", Gender: " << gender
        << ", Specialty: " << specialty << endl;

    }

};

```

```

int Doctor::idCounter = 0;

```

```

class Appointment {

    int patientID;

    int doctorID;

    string date; public:

        Appointment(int pid, int did, string d) : patientID(pid),
        doctorID(did), date(d) {}    void display() const {            cout
        << "Appointment - Patient ID: " << patientID
        << ", Doctor ID: " << doctorID << ", Date: " << date
        << endl;

    }
}

```

```
};
```

```
vector<Patient> patients; vector<Doctor>  
doctors; vector<Appointment>  
appointments;
```

```
void addPatient() {    string  
name, gender;  
    int age;  
    cout << "Enter patient name: ";    cin >>  
name;    cout << "Enter age: ";    cin >> age;  
    cout << "Enter gender: ";    cin >> gender;  
    patients.emplace_back(name, age, gender);  
    cout << "Patient added successfully.\n";  
}
```

```
void addDoctor() {    string name,  
gender, specialty;    int age;  
    cout << "Enter doctor name: ";
```

```
cin >> name;    cout << "Enter  
age: ";    cin >> age;    cout <<  
"Enter gender: ";    cin >> gender;  
  
    cout << "Enter specialty: ";    cin >> specialty;  
doctors.emplace_back(name, age, gender, specialty);  
cout << "Doctor added successfully.\n";  
}
```

```
void viewPatients() {    cout << "\n---  
List of Patients ---\n";    for (const  
auto& p : patients)  
    p.display();  
}
```

```
void viewDoctors() {    cout << "\n---  
List of Doctors ---\n";    for (const  
auto& d : doctors)  
    d.display();  
}
```



```

void bookAppointment() {
    int pid, did;    string date;    cout << "Enter patient ID
(1 to " << patients.size() <<
"): ";    cin >> pid;    cout << "Enter doctor ID (1 to " <<
doctors.size() << "):
";    cin >> did;    cout << "Enter date
(YYYY-MM-DD): ";    cin >> date;

    if (pid > 0 && pid <= patients.size() && did > 0 && did
<= doctors.size()) {
        appointments.emplace_back(pid, did, date);        cout
<< "Appointment booked successfully.\n";

        } else {            cout << "Invalid IDs
entered.\n";

        }
    }
}

```

```

void viewAppointments() {    cout << "\n---
Appointments List ---\n";    for (const auto&
a : appointments)

        a.display();

}

int main() {

int    choice;

do {

        cout << "\n===== Hospital Management System
=====\\n";

        cout << "1. Add Patient\\n";

cout << "2. Add Doctor\\n";        cout <<

"3. View Patients\\n";        cout << "4.

View Doctors\\n";        cout << "5. Book

Appointment\\n";        cout << "6. View

Appointments\\n";        cout << "0.

Exit\\n";

```

```

        cout << "Enter choice: ";

cin >> choice;


        switch (choice) {
            case 1: addPatient();
break;
            case 2: addDoctor(); break;
case 3: viewPatients(); break;
            case 4:
viewDoctors(); break;
            case 5:
bookAppointment(); break;
            case 6:
viewAppointments(); break;
            case 0: cout
<< "Exiting system.\n"; break;
            default: cout
<< "Invalid choice.\n";

        }

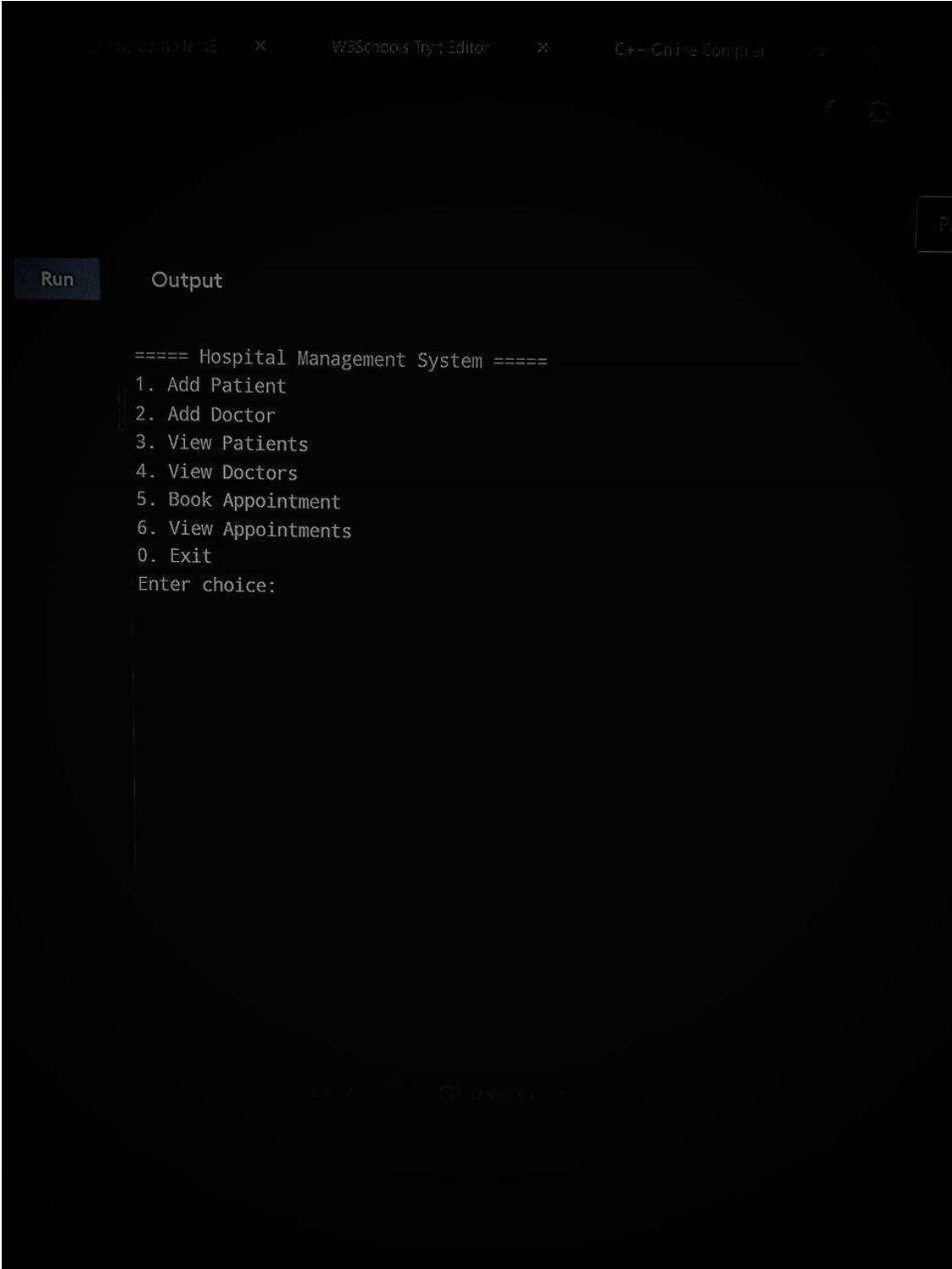
    } while (choice != 0);

    return 0;

}

```

OUTPUT



The screenshot shows a web browser window with three tabs: "C++ Compiler", "W3Schools Tryit Editor", and "C++ Online Compiler". The "C++ Online Compiler" tab is active. The compiler interface includes a "Run" button and an "Output" section. The output displays the following text:

```
==== Hospital Management System ====  
1. Add Patient  
2. Add Doctor  
3. View Patients  
4. View Doctors  
5. Book Appointment  
6. View Appointments  
0. Exit  
Enter choice:
```

Below the output, there is a large, empty text input area for providing input to the program. At the bottom of the page, there is a footer with the text "C++ Online Compiler" and a logo.

Run

Output

```
===== Hospital Management System =====
```

1. Add Patient
2. Add Doctor
3. View Patients
4. View Doctors
5. Book Appointment
6. View Appointments
0. Exit

Enter choice: 1

Enter patient name: Anuj

Enter age: 35

Enter gender: Male

Patient added successfully.

```
===== Hospital Management System =====
```

1. Add Patient
2. Add Doctor
3. View Patients
4. View Doctors
5. Book Appointment
6. View Appointments
0. Exit

Enter choice: |

LIMITATION

Though we have completed this project with all our effort but has certain limitation like it cannot be operated in networking. the working of project is slow in windows operating system.

CONCLUSION

This project is very useful in preparing hospital management. this project meets our requirements and complete in all sense. this software is really of great use for Hospital Management, who want their hospital Mnagement to be computerized.

BIBLIOGRAPHY

Anand Hospital in Meerut is a comprehensive, multispecialty hospital known for its advanced diagnostic and therapeutic services. It was established in 2007 and has become a significant healthcare provider in western Uttar Pradesh. The hospital offers a wide range of medical services, including round-the-clock emergency care, critical care, and specialized departments like cardiology, oncology, and orthopedics. Anand Hospital is also recognized for its NABH accreditation and has pioneered the first liver transplant in Meerut.

Key Features and Services:

- Comprehensive Care:

Anand Hospital provides a wide range of medical services, catering to various specialties and subspecialties.

- Advanced Technology:

The hospital is equipped with state-of-the-art technology and equipment for both diagnostic and therapeutic purposes.

- Specialized Departments:

Offers specialized departments like cardiology, oncology, orthopedics, and more.

- Emergency Services:

Provides 24/7 emergency care, critical care, and accident care.

- NABH Accreditation:

The hospital has achieved NABH Full Accreditation across 35 specialties, demonstrating its commitment to patient safety and healthcare quality.

- Liver Transplant Pioneer:

Anand Hospital has successfully performed the first liver transplant in Meerut.

- Blood Bank:

Equipped with advanced technology for collecting, processing, and supplying blood, including plasmapheresis.

- Health Plans:

Offers various health plans for comprehensive body check-ups at discounted rates.