A

PROJECT REPORT

ON

"HOSPITAL MANAGEMENT SYSTEM"

SUBMITTED BY:
Pingale Akash Vilas (2124UCEM1027)

SUBJECT:

Programming in Problem

Solving using C++

Under the guidance of

Miss. Ishwari Tirse

Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

INDEX

INTRODUCTION

A Hospital Management System (HMS) is an integrated software solution designed to streamline and enhance the operational efficiency of healthcare facilities. As the healthcare sector increasingly relies on technology, an HMS plays a crucial role in managing various functions, including patient registration, appointment scheduling, billing, and medical records management.

The primary goal of an Hospital Management System is to improve patient care by facilitating easy access to patient information and enabling efficient communication among healthcare providers. By automating routine tasks, the system reduces administrative burdens, minimizes errors, and enhances data accuracy. This results in faster service delivery, better resource management, and improved overall patient satisfaction.

The implementation of An Hospital Management System is not only enhances the efficiency of hospital operations but also significantly improves the patient experience by ensuring timely access to care and reducing waiting times. Hospital Management System is essential for delivering high-quality care, enhancing operational efficiency, and driving the future of healthcare delivery. This mini-project aims to develop a prototype of a Hospital Management System to demonstrate its essential features and benefits, ultimately contributing to better healthcare delivery.

CODE

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
class Person {
protected:
    string name;
    int age;
    string gender;
    string address;
    string contactNumber;
public:
Person(string n, int a, string g, string addr, string c)
        : name(n), age(a), gender(g), address(addr), contactNumber(c) {}
virtual void display() const {
cout << "Name: " << name << "\nAge: " << age
            << "\nGender: " << gender
            << "\nAddress: " << address
            << "\nContact Number: " << contactNumber << endl;
    }
};
class Patient : public Person {
private:
```

```
    string ailment;
    string admissionDate;
    string dischargeDate;
    string doctorAssigned;
    string insuranceProvider;
    string medicalHistory;
    string allergies;
public:
    Patient(string n, int a, string g, string addr, string c,
            string ail, string admDate, string disDate,
            string doc, string ins, string medHist, string all)
        : Person(n, a, g, addr, c), ailment(ail), admissionDate(admDate),
          dischargeDate(disDate), doctorAssigned(doc),
          insuranceProvider(ins), medicalHistory(medHist), allergies(all) {}

    void display() const override {
        Person::display();
        cout << "Ailment: " << ailment
             << "\nAdmission Date: " << admissionDate
             << "\nDischarge Date: " << dischargeDate
             << "\nDoctor Assigned: " << doctorAssigned
             << "\nInsurance Provider: " << insuranceProvider
             << "\nMedical History: " << medicalHistory
             << "\nAllergies: " << allergies << endl;
    }
```

```
};

// Derived class forDoctor
Class Doctor: public Person {
private:
    string specialization;
    int yearsOfExperience;
    string licenseNumber;
    string hospitalAffiliation;
    string consultingHours;
    string qualifications;
    string certifications;
    string contactEmail;

public:
    Doctor(string n, int a, string g, string addr, string c,
           string spec, int exp, string licNum, string hospAff,
           string hours, string qual, string cert, string email)
        : Person(n, a, g, addr, c), specialization(spec),
          yearsOfExperience(exp), licenseNumber(licNum),
          hospitalAffiliation(hospAff), consultingHours(hours),
          qualifications(qual), certifications(cert),
          contactEmail(email) {}

    void display() const override {
```

```
        Person::display();
        cout << "Specialization: " << specialization
             << "\nYears of Experience: " << yearsOfExperience
             << "\nLicense Number: " << licenseNumber
             << "\nHospital Affiliation: " << hospitalAffiliation
             << "\nConsulting Hours: " << consultingHours
             << "\nQualifications: " << qualifications
             << "\nCertifications: " << certifications
             << "\nContact Email: " << contactEmail << endl;
    }
};

// Hospital Management System
class Hospital {
private:
    vector<Patient> patients;
    vector<Doctor> doctors;
public:
    void addPatient(const Patient& patient) {
        patients.push_back(patient);
    }
    void addDoctor(const Doctor& doctor) {
        doctors.push_back(doctor);
    }
    void displayPatients() const {
```

```
        cout << "\nPatients List:\n";
        for (const auto& patient : patients) {
            patient.display();
            cout << "--------------------\n";
        }
    }
    void displayDoctors() const {
        cout << "\nDoctors List:\n";
        for (const auto& doctor : doctors) {
            doctor.display();
            cout << "--------------------\n";
        }
    }
};
int main() {
Hospital hospital;
hospital.addPatient(Patient("Sujit more", 19,"Male", "123 Elm St",
"9356672433",
                            "Flu", "2024-01-01", "2024-01-10",
                            "Dr. Smith", "XYZ Insurance",
                            "No major issues", "None"));
    hospital.addPatient(Patient("Gaurav Bodkhe", 25, "male", "456 Oak St",
"7709223249",
                            "Cough", "2024-01-05", "2024-01-15",
                            "Dr. Brown", "ABC Insurance",
                            "Asthma", "Peanuts"));
```

```
    // Adding doctors
    hospital.addDoctor(Doctor("Dr. saushmita sen", 45, "Female", "789 Pine St", "9356672433",
                              "Cardiology", 20, "LIC123456", "City Hospital",
                              "9 AM - 5 PM", "MD, PhD", "ACLS Certified",
                              "saushmita12@gmail.com"));
    hospital.addDoctor(Doctor("Dr. Ramesh", 50, "Male", "101 Maple St", "555-4321",
                              "Neurology", 25, "LIC654321", "County Hospital",
                              "10 AM - 6 PM", "MD", "Neuro Board Certified",
                              "Ramesh1@gmail.com"));
    // Displaying all patients and doctors
    hospital.displayPatients();
    hospital.displayDoctors();
    return 0;
}
```

OUTPUT

```
Patients List:
Name: Sujit more
Age: 19
Gender: Male
Address: 123 Elm St
Contact Number: 9356672433
Ailment: Flu
Admission Date: 2024-01-01
Discharge Date: 2024-01-10
Doctor Assigned: Dr. Smith
Insurance Provider: XYZ Insurance
Medical History: No major issues
Allergies: None
----------------------
Name: Gaurav Bodkhe
Age: 25
Gender: male
Address: 456 Oak St
Contact Number: 7709223249
Ailment: Cough
Admission Date: 2024-01-05
Discharge Date: 2024-01-15
Doctor Assigned: Dr. Brown
Insurance Provider: ABC Insurance
Medical History: Asthma
```

```
Doctors List:
Name: Dr. saushmita sen
Age: 45
Gender: Female
Address: 789 Pine St
Contact Number: 9356672433
Specialization: Cardiology
Years of Experience: 20
License Number: LIC123456
Hospital Affiliation: City Hospital
Consulting Hours: 9 AM - 5 PM
Qualifications: MD, PhD
Certifications: ACLS Certified
Contact Email: saushmita12@gmail.com
----------------------
Name: Dr. Ramesh
Age: 50
Gender: Male
Address: 101 Maple St
Contact Number: 555-4321
Specialization: Neurology
Years of Experience: 25
License Number: LIC654321
Hospital Affiliation: County Hospital
Consulting Hours: 10 AM - 6 PM
```

```
Consulting Hours: 10 AM - 6 PM
Qualifications: MD
Certifications: Neuro Board Certified
Contact Email: Ramesh1@gmail.com
```

CONCLUSION

"This project demonstrates a hospital management system using OOP concepts in C++. We designed classes for patients, doctors, and and implemented methods for adding, deleting, and searching records. The system provides a user-friendly interface for hospital staff to manage patient care and administrative tasks efficiently.

We utilized encapsulation to hide data and ensure data integrity, inheritance to create specialized classes for different medical specialties, and polymorphism to enable flexible searching and sorting of records.

The system allows for easy management of patient records, doctor schedules, and appointments. It also enables searching for specific patient or doctor information, and generates reports for administrative purposes.

Overall, our hospital management system demonstrates a comprehensive and efficient solution for managing patient care and administrative tasks in a hospital setting. Its scalability and flexibility make it an ideal starting point for further development and integration with other healthcare systems."