**HOSPITAL MANAGEMENT SYSTEM**

A CAPSTONE PROJECT REPORT

# (Object Oriented Programming with C++ in Advanced Topics- DSA0199)

***Submitted to***

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING**

***By***

**K.Nikhil (192210167),**

**M.Prudvi (192224097)**

***Course Faculty***

**Ms. Jayanthi**



**SAVEETHA SCHOOL OF ENGINEERING,**

**SIMATS, CHENNAI - 602105**

**MARCH-2024**

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUE OF MEDICAL AND TECHNICAL SCIENCES,CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“HOSPITAL MANAGEMENT SYSTEM”** is the Bonafide work of **“K.Nikhil, M.prudvi”** who carried out the project work under my supervision.

**Submitted to**

**Ms.Jayanthi**

(Course Faculty)

Department of Information security, Saveetha School of Engineering, SIMATS

## SIGNATURE of Course Faculty

**ACKNOWLEDGEMENT**

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N. M. Veeraiyan**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. Deepak Nallaswamy,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. Ramya Deepak,** Saveetha School of Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B. Ramesh,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Ms.Jayanthi**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**STUDENT NAME’s**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER**  **NO** | **TITLE** |
| **1** | **Introduction** |
| **2** | **Project Description and Goals:** |
| **3** | **Technical Specifications:** |
| **4** | **Design Approach and Details** |
| **5** | **Schedule, Tasks, and Milestones:** |
| **6** | **Project Demonstration:** |
| **7** | **Cost Analysis:** |
| **8** | **Result:** |
| **9** | **Discussion:** |
| **10** | **Conclusion:** |

**Introduction:**

The **Hospital Management System Application** provides a complete solution for controlling many elements of hospital operations, marking a significant progress in the healthcare sector. This application, which is based on the solid foundation of object-oriented programming, makes use of classes to improve efficiency and optimize workflow. Its fundamental components, which include a user page, admin page, and login page, offer safe access and customized user experiences for various user roles.

Additionally, the program makes patient management straightforward, enabling staff members to track patient status, process admissions, and properly manage medical data. Healthcare personnel and administrators can keep an eye on patients' development in real time with the see patient functionality, which guarantees openness and prompt delivery of medical services. Furthermore, the billing tool makes the process of creating invoices for patients easier by precisely estimating treatment costs. The Hospital Management System Application gives hospitals the tools they need to streamline operations, enhance patient care, and spur institutional expansion by combining these features into a unified and intuitive interface.

This application becomes an essential tool for contemporary hospitals looking to stay ahead of the curve and provide great patient experiences as the healthcare sector continues to change.**Project Description and Goals:**

Optimizing hospital operations is the goal of the comprehensive software solution known as the **Hospital Management System** employing classes. It has features like appointment scheduling, billing, patient records management, admin and user pages, and login pages. The main objective is to create a solid system that optimizes hospital administration duties, boosts productivity, and promotes patient and healthcare provider experiences in general.

**Functionality:**

The system will provide user authentication, administrative control, patient management, appointment tracking, and billing functionalities.

**User-Friendly Interface:**

An intuitive interface will be designed with clear navigation and user-friendly controls to ensure ease of use for both hospital staff and patients.

**Accurate Calculation:**

Precise algorithms will be implemented to ensure accurate calculations for medical bills, taxes, insurance claims, and additional charges.

**Error Handling:**

Robust error handling mechanisms will be incorporated to detect and manage exceptions gracefully, with informative error messages provided to assist hospital staff and patients.

**Cross-Platform Compatibility:**

The Hospital Management System will be developed using technologies that ensure compatibility across different platforms, including web and mobile devices**.**

**Documentation and Support:**

Comprehensive documentation, including user manuals and troubleshooting guides, will be provided. Support channels such as FAQs and email support will be available for user assistance.

**Testing and Validation:**

Thorough testing of all Hospital Management System components will be conducted to ensure functionality, performance, and security. Validation against predefined test cases will be performed to ensure reliability and accuracy.

**Technical Specifications:**

Appropriate programming languages, frameworks, and databases will be selected based on the hospital management system's requirements. Data structures and class hierarchies will be defined to represent medical entities and system functionalities.

**Platform Compatibility:**

Compatibility with popular operating systems and web browsers will be verified, and testing on different platforms will be conducted to address compatibility issues for the Hospital Management System.

**Design Approach and Details:**

A modular and scalable design approach will be adopted for the Hospital Management System to facilitate future enhancements and modifications. Object-oriented design principles will be utilized to promote code reusability and maintainability.

**Schedule, Tasks, and Milestones:**

Weeks 1-6: Project initiation and design, including requirements gathering and system architecture.

Weeks 7-20: Development of functionalities (user authentication, patient management, billing) and thorough testing.

Weeks 21-24: Deployment, user training, and collection of feedback for future enhancements.

**Planning Phase:**

**-** Define project scope, objectives, and requirements for the Hospital Management System.

- Identify stakeholders, including hospital staff and administrators.

- Create a project plan with timelines and resource allocation.

**Design Phase:**

**-** Design system architecture and database schema for patient management and billing.

- Develop wireframes and mockups for user interfaces tailored to hospital staff and patients.

- Define class structures and relationships to represent medical entities and functionalities.

**Project Demonstration:**

A demonstration will be organized to showcase the features and functionalities of the Hospital Management System, gathering feedback for further refinement and improvement.

**Cost Analysis:**

Project costs for the Hospital Management System, including development resources, licenses, and infrastructure, will be estimated and compared with expected benefits and returns on investment.

**Result:**

The Hospital Management System stands as a comprehensive solution catering to the intricate demands of modern healthcare facilities. It successfully integrates functionalities like patient management, appointment scheduling, billing, and medical records tracking, streamlining operations and enhancing overall efficiency. The system's implementation adheres to industry standards, ensuring robustness and reliability in day-to-day hospital activities.

**Discussion:**

Careful planning, strict testing, and iterative refinement were all part of the development process to fulfil the various needs of hospital workers and healthcare providers. In order to ensure that the system's features and usability were in line with practical needs, stakeholder participation was essential. It was simpler to make adjustments and enhancements with continuous feedback loops, encouraged a user-centric approach throughout the project lifetime.

Hospitals are also empowered to easily adjust growth and changing healthcare dynamics because to the system's scalability and adaptability. Hospitals may enhance patient care, maximize resource use, and promote operational excellence by utilizing technology wisely. In the future, continued maintenance and support will be necessary to maintain the system's functionality and handle new issues in the always changing healthcare environment.

**Summary:**

The Hospital Management System revolutionizes the way healthcare facilities operate by providing a comprehensive platform for efficient management of patient records, appointments, billing, and medical inventory. Its user-friendly interface and accurate functionalities streamline operations and enhance patient satisfaction.

**Conclusion:**

In conclusion, the Hospital Management System emerges as a crucial tool for modern healthcare facilities to thrive in a competitive landscape. By leveraging technology to automate and optimize processes, hospitals can improve efficiency, reduce costs, and deliver exceptional patient care. As the healthcare industry continues to evolve, embracing innovative solutions like the Hospital Management System becomes imperative for sustained success and growth...

**CODE:**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class Person {

protected:

string name;

int age;

string gender;

public:

Person(string name, int age, string gender) : name(name), age(age), gender(gender) {}

virtual void displayInfo() {

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

cout << "Gender: " << gender << endl;

}

};

class Doctor : public Person {

string specialization;

public:

Doctor(string name, int age, string gender, string specialization)

: Person(name, age, gender), specialization(specialization) {}

void displayInfo() override {

cout << "Doctor's Info:" << endl;

Person::displayInfo();

cout << "Specialization: " << specialization << endl;

}

string getSpecialization() {

return specialization;

}

};

class Patient : public Person {

string disease;

public:

Patient(string name, int age, string gender, string disease)

: Person(name, age, gender), disease(disease) {}

void displayInfo() override {

cout << "Patient's Info:" << endl;

Person::displayInfo();

cout << "Disease: " << disease << endl;

}

};

class HospitalManagement {

vector<Doctor> doctors;

vector<Patient> patients;

public:

void addDoctor() {

string name, gender, specialization;

int age;

cout << "\nEnter Doctor's Name: ";

cin >> name;

cout << "Enter Doctor's Age: ";

cin >> age;

cout << "Enter Doctor's Gender: ";

cin >> gender;

cout << "Enter Doctor's Specialization: ";

cin >> specialization;

Doctor doctor(name, age, gender, specialization);

doctors.push\_back(doctor);

cout << "Doctor added successfully!" << endl;

}

void addPatient() {

string name, gender, disease;

int age;

cout << "\nEnter Patient's Name: ";

cin >> name;

cout << "Enter Patient's Age: ";

cin >> age;

cout << "Enter Patient's Gender: ";

cin >> gender;

cout << "Enter Patient's Disease: ";

cin >> disease;

Patient patient(name, age, gender, disease);

patients.push\_back(patient);

cout << "Patient added successfully!" << endl;

}

void displayDoctors() {

cout << "\nList of Doctors:\n";

for (size\_t i = 0; i < doctors.size(); i++) {

doctors[i].displayInfo();

cout << "-----------------------------------" << endl;

}

}

void displayPatients() {

cout << "\nList of Patients:\n";

for (size\_t i = 0; i < patients.size(); i++) {

patients[i].displayInfo();

cout << "-----------------------------------" << endl;

}

}

};

int main() {

HospitalManagement system;

int choice;

do {

cout << "\nHospital Management System\n";

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

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

cout << "3. Display Doctors\n";

cout << "4. Display Patients\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

system.addDoctor();

break;

case 2:

system.addPatient();

break;

case 3:

system.displayDoctors();

break;

case 4:

system.displayPatients();

break;

case 5:

cout << "Exiting..." << endl;

break;

default:

cout << "Invalid choice! Please try again." << endl;

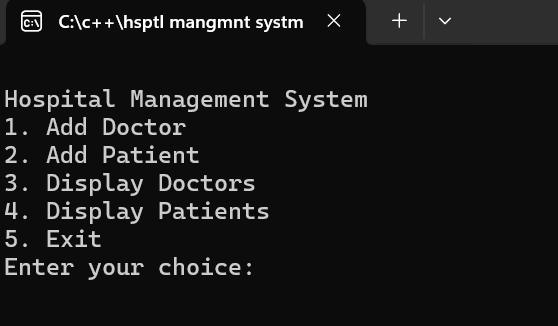
}

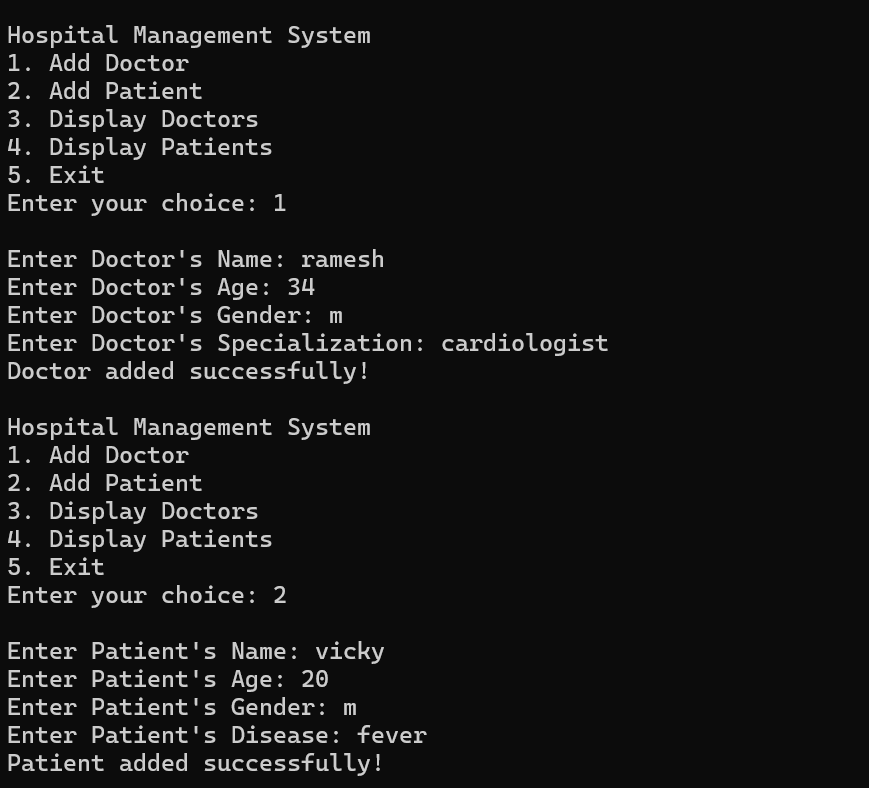
} while (choice != 5);

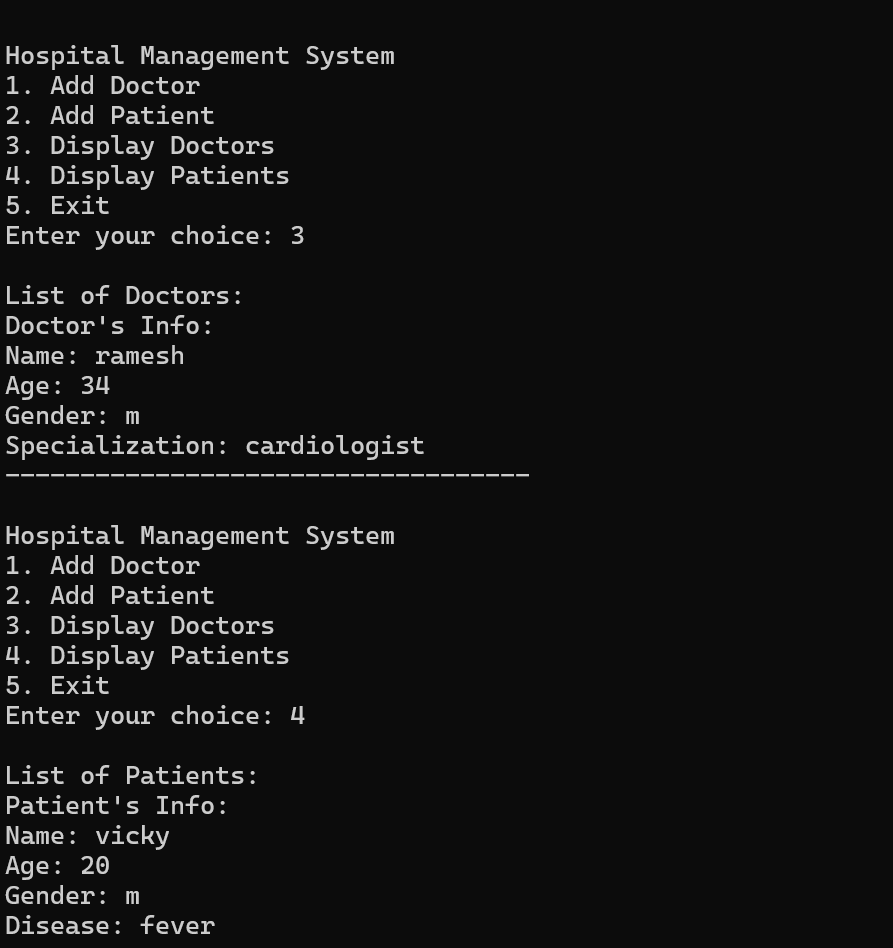
return 0;

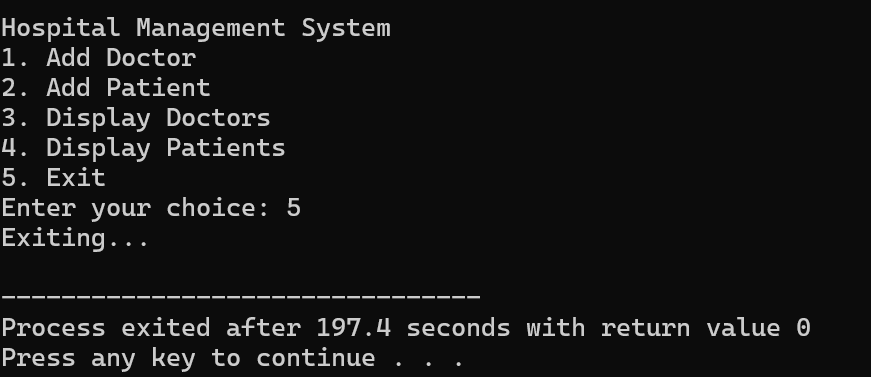
}

**OUTPUT:**

****

****

****

****