**A**

**PROJECT REPORT**

**ON**

**“Health Record Management System”**

**SUBMITTED BY:**

**Vedant Rajendra Borawake**

**PRN:- 2124UCEM1112**

**Subject:-**

**Programming and 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**

|  |  |  |
| --- | --- | --- |
| **SR.**  **NO** | **CONTENT** | **PAGE NO.** |
| **1.** | **INTRODUCTION** | **3** |
| **2.** | **CODE** | **4-8** |
| **3.** | **OUTPUT** | **9-10** |
| **4.** | **CONCLUSION** | **11** |

**Introduction**

This mini-project is a simple **Health Record Management System** that is implemented in C++. It can be useful to the user in managing patient information. This is made possible through the following three operations: patient records addition, viewing of a patient record, and deletion of a patient record. The object-oriented programming principle that applies is the tracking of the list of patients with regard to the name, age, and disease by each patient. The basic data structures, classes, and standard I/O operations combined together in the small-scale application.

**Code**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class Patient {

public:

string name;

int age;

string disease;

Patient(string n, int a, string d) : name(n), age(a), disease(d) {}

};

class HealthRecordSystem {

private:

vector<Patient> patients;

public:

void addPatient(string name, int age, string disease) {

Patient newPatient(name, age, disease);

patients.push\_back(newPatient);

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

}

void viewPatients() {

if (patients.empty()) {

cout << "No patient records available." << endl;

return;

}

cout << "Patient Records:" << endl;

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

cout << "Patient " << (i + 1) << ": " << patients[i].name

<< ", Age: " << patients[i].age

<< ", Disease: " << patients[i].disease << endl;

}

}

void deletePatient(int index) {

if (index < 0 || index >= patients.size()) {

cout << "Invalid patient index." << endl;

return;

}

patients.erase(patients.begin() + index);

cout << "Patient deleted successfully." << endl;

}

};

int main() {

HealthRecordSystem system;

int choice;

do {

cout << "\nHealth Record Management System" << endl;

cout << "1. Add Patient" << endl;

cout << "2. View Patients" << endl;

cout << "3. Delete Patient" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

string name;

int age;

string disease;

cout << "Enter patient name: ";

cin >> name;

cout << "Enter patient age: ";

cin >> age;

cout << "Enter disease: ";

cin >> disease;

system.addPatient(name, age, disease);

break;

}

case 2:

system.viewPatients();

break;

case 3: {

int index;

cout << "Enter patient index to delete: ";

cin >> index;

system.deletePatient(index - 1);

break;

}

case 4:

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

break;

default:

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

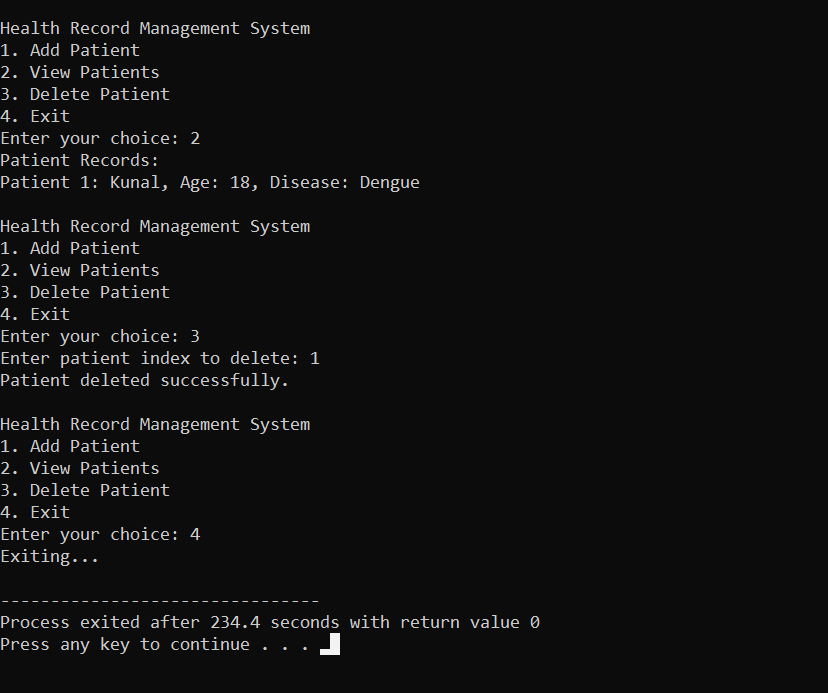
}

} while (choice != 4);

    return 0;

}

**Output**



**Conclusion**

The Health Record Management System is a good example of a simple, interactive program to work with patient records. It demonstrates some important concepts of C++ such as class design, encapsulation, and vector-based data handling. This mini-project will further allow the inclusion of feature enhancements such as data validation, error handling, and persistent storage to provide a more complete healthcare management application