#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

const string FILENAME = "students.dat";

class StudentSystem {

struct Student {

int id;

char name[50];

int age;

float gpa;

};

public:

void addStudent();

void updateStudent();

void deleteStudent();

void searchStudent();

void displayStudent(const Student& student);

private:

fstream file;

};

void StudentSystem::addStudent() {

Student student;

cout << "Enter Student ID: ";

cin >> student.id;

cout << "Enter Student Name: ";

cin.ignore();

cin.getline(student.name, 50);

cout << "Enter Student Age: ";

cin >> student.age;

cout << "Enter Student GPA: ";

cin >> student.gpa;

file.open(FILENAME, ios::in | ios::out | ios::binary);

if (!file) {

file.open(FILENAME, ios::out | ios::binary);

}

file.seekp(student.id \* sizeof(Student));

file.write(reinterpret\_cast<char\*>(&student), sizeof(Student));

file.close();

cout << "Student added successfully.\n";

}

void StudentSystem::updateStudent() {

Student student;

cout << "Enter Student ID to update: ";

cin >> student.id;

file.open(FILENAME, ios::in | ios::out | ios::binary);

if (!file) {

cout << "No records found.\n";

return;

}

file.seekg(student.id \* sizeof(Student));

file.read(reinterpret\_cast<char\*>(&student), sizeof(Student));

if (file.gcount() == 0) {

cout << "No record found for ID " << student.id << ".\n";

file.close();

return;

}

cout << "Updating record for ID " << student.id << ":\n";

cout << "Enter new Student Name: ";

cin.ignore();

cin.getline(student.name, 50);

cout << "Enter new Student Age: ";

cin >> student.age;

cout << "Enter new Student GPA: ";

cin >> student.gpa;

file.seekp(student.id \* sizeof(Student));

file.write(reinterpret\_cast<char\*>(&student), sizeof(Student));

file.close();

cout << "Student updated successfully.\n";

}

void StudentSystem::deleteStudent() {

int id;

Student student = {0, "", 0, 0.0};

cout << "Enter Student ID to delete: ";

cin >> id;

file.open(FILENAME, ios::in | ios::out | ios::binary);

if (!file) {

cout << "No records found.\n";

return;

}

file.seekg(id \* sizeof(Student));

file.read(reinterpret\_cast<char\*>(&student), sizeof(Student));

if (file.gcount() == 0) {

cout << "No record found for ID " << id << ".\n";

file.close();

return;

}

file.seekp(id \* sizeof(Student));

file.write(reinterpret\_cast<char\*>(&student), sizeof(Student));

file.close();

cout << "Student deleted successfully.\n";

}

void StudentSystem::searchStudent() {

int id;

Student student;

cout << "Enter Student ID to search: ";

cin >> id;

file.open(FILENAME, ios::in | ios::binary);

if (!file) {

cout << "No records found.\n";

return;

}

file.seekg(id \* sizeof(Student));

file.read(reinterpret\_cast<char\*>(&student), sizeof(Student));

if (file.gcount() == 0) {

cout << "No record found for ID " << id << ".\n";

file.close();

return;

}

cout << "Student Record Found:\n";

displayStudent(student);

file.close();

}

void StudentSystem::displayStudent(const Student& student) {

cout << "ID: " << student.id << "\n";

cout << "Name: " << student.name << "\n";

cout << "Age: " << student.age << "\n";

cout << "GPA: " << fixed << setprecision(2) << student.gpa << "\n";

}

int main() {

StudentSystem system;

int choice;

do {

cout << "\nStudent Information System\n";

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

cout << "2. Update Student\n";

cout << "3. Delete Student\n";

cout << "4. Search Student\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: system.addStudent(); break;

case 2: system.updateStudent(); break;

case 3: system.deleteStudent(); break;

case 4: system.searchStudent(); break;

case 5: cout << "Exiting program.\n"; break;

default: cout << "Invalid choice. Please try again.\n";

}

} while (choice != 5);

return 0;

}

