# Practical 9 Source Code:-

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

struct Student { int rollNumber; string name; string division;

string address;

};

// Function to add a student record void addStudent(const string& filename) { ofstream outFile(filename, ios::app);

if (!outFile) {

cerr << "Error opening file!" << endl; return;

}

Student student; cout << "Enter roll number: "; cin >> student.rollNumber;

cin.ignore(); // Ignore the newline character left in the buffer cout << "Enter name: "; getline(cin, student.name); cout << "Enter division: "; getline(cin, student.division); cout << "Enter address: ";

getline(cin, student.address);

outFile << student.rollNumber << endl; outFile << student.name << endl; outFile << student.division << endl; outFile << student.address << endl;

outFile.close();

cout << "Student record added successfully!" << endl;

}

// Function to delete a student record void deleteStudent(const string& filename) {

int rollNumber;

cout << "Enter roll number of student to delete: "; cin >> rollNumber;

ifstream inFile(filename); ofstream outFile("temp.txt"); if (!inFile || !outFile) {

cerr << "Error opening file!" << endl; return;

}

bool found = false;

Student student;

while (inFile >> student.rollNumber) { inFile.ignore(); // Ignore the newline character getline(inFile, student.name); getline(inFile, student.division);

getline(inFile, student.address);

if (student.rollNumber == rollNumber) { found = true;

cout << "Student record deleted: " << student.name << endl; continue; // Skip writing this record to the temp file

}

// Write the record to the temp file outFile << student.rollNumber << endl; outFile << student.name << endl; outFile << student.division << endl;

outFile << student.address << endl;

}

inFile.close();

outFile.close();

// Replace old file with the new file remove(filename.c\_str()); rename("temp.txt", filename.c\_str());

if (!found) {

cout << "No student record found with roll number " << rollNumber << endl;

} else {

cout << "Student record deleted successfully!" << endl;

}

}

// Function to display student details void displayStudent(const string& filename) {

int rollNumber;

cout << "Enter roll number of student to display: "; cin >> rollNumber;

ifstream inFile(filename);

if (!inFile) { cerr << "Error opening file!" << endl;

return;

}

Student student; bool found = false;

while (inFile >> student.rollNumber) { inFile.ignore(); // Ignore the newline character getline(inFile, student.name);

getline(inFile, student.division);

getline(inFile, student.address);

if (student.rollNumber == rollNumber) { found = true;

cout << "Student Details:\n";

cout << "Roll Number: " << student.rollNumber << endl; cout << "Name: " << student.name << endl; cout << "Division: " << student.division << endl;

cout << "Address: " << student.address << endl; break;

}

}

inFile.close();

if (!found) {

cout << "No student record found with roll number " << rollNumber << endl;

}

}

// Main function int main() {

string filename = "students.txt";

int choice;

do {

cout << "\nStudent Database Menu:\n";

cout << "1. Add Student\n"; cout << "2. Delete Student\n"; cout << "3. Display Student\n"; cout << "4. Exit\n"; cout << "Enter your choice: ";

cin >> choice;

switch (choice) { case 1:

addStudent(filename);

break; case 2:

deleteStudent(filename);

break; case 3:

displayStudent(filename); break; case 4:

cout << "Exiting program." << endl; break; default:

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

}

} while (choice != 4);

return 0;

}

# Output:-



