**BIGLEARN TRINING INSTITUTE**

**TRICHY**

****

**STUDENT MANAGEMENT SYSTEM**

**GUIDED BY: BY**

**V.AUGNES MARRY PRADEEP.E.G.**

**INTRODUCTION**

A **Student Management System (SMS)** is a software application designed to manage and streamline the administrative and academic processes associated with student data in educational institutions. The primary objective of such a system is to organize, manage, and track student-related information efficiently, reducing the administrative burden on educators and staff while ensuring that data is accurate and easily accessible.

**FEATURES:**

**1.Student Registration and Enrollment**

**2.Student Information Management**

**3.Grade and Performance Tracking**

**4.Attendance Management**

**5.Student Search and Filtering**

**6.Data Security and Access Control**

**SYSTEM CONFIGURATION:**

**PROCESSOR:** Core i3

INSTALLED RAM: 8.0GB

SYSTEM TYPE :64-BIT OS

**SYSTEM REQUIREMENTS:**

1.C++

**C++ PACKAGES USED:**

**1. Libraries**

* **Standard Libraries**: C++ comes with a rich set of standard libraries like the Standard Template Library (STL), which provides common data structures and algorithms (e.g., vector, map, sort).
* **Custom Libraries**: You can create or use third-party libraries that are packaged as .lib (static libraries) or .dll (dynamic libraries) in Windows, and .a or .so in Unix-like systems. These libraries are collections of compiled code that can be linked to your program to add functionality without including the source code directly.

**2. Namespaces**

* Namespaces in C++ are used to group related classes, functions, and variables under a single name, reducing the chance of name collisions. For example, the std namespace includes all the standard library functions and objects.

**PROGRAM:**

**#include <iostream>**

**#include <vector>**

**#include <string>**

**using namespace std;**

**class Student {**

**public:**

**string name;**

**int id;**

**float grade;**

**Student(string name, int id, float grade) {**

**this->name = name;**

**this->id = id;**

**this->grade = grade;**

**}**

**void display() {**

**cout << "ID: " << id << ", Name: " << name << ", Grade: " << grade << endl;**

**}**

**};**

**class StudentManagementSystem {**

**private:**

**vector<Student> students;**

**public:**

**void addStudent(string name, int id, float grade) {**

**students.push\_back(Student(name, id, grade));**

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

**}**

**void deleteStudent(int id) {**

**bool found = false;**

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

**if (students[i].id == id) {**

**students.erase(students.begin() + i);**

**cout << "Student with ID " << id << " deleted successfully.\n";**

**found = true;**

**break;**

**}**

**}**

**if (!found) {**

**cout << "Student with ID " << id << " not found.\n";**

**}**

**}**

**void viewStudents() {**

**if (students.empty()) {**

**cout << "No students in the system.\n";**

**return;**

**}**

**cout << "Student Records:\n";**

**for (const auto& student : students) {**

**student.display();**

**}**

**}**

**};**

**int main() {**

**StudentManagementSystem sms;**

**int choice;**

**string name;**

**int id;**

**float grade;**

**do {**

**cout << "\nStudent Management System Menu:\n";**

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

**cout << "2. Delete Student\n";**

**cout << "3. View Students\n";**

**cout << "4. Exit\n";**

**cout << "Enter your choice: ";**

**cin >> choice;**

**switch (choice) {**

**case 1:**

**cout << "Enter Student Name: ";**

**cin.ignore(); // To ignore the newline character left by previous input**

**getline(cin, name);**

**cout << "Enter Student ID: ";**

**cin >> id;**

**cout << "Enter Student Grade: ";**

**cin >> grade;**

**sms.addStudent(name, id, grade);**

**break;**

**case 2:**

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

**cin >> id;**

**sms.deleteStudent(id);**

**break;**

**case 3:**

**sms.viewStudents();**

**break;**

**case 4:**

**cout << "Exiting the system.\n";**

**break;**

**default:**

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

**}**

**} while (choice != 4);**

**return 0;**

**}**

**OUTPUT:**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

**3. View Students**

**4. Exit**

**Enter your choice: 1**

**Enter Student Name: Alice**

**Enter Student ID: 101**

**Enter Student Grade: 85.5**

**Student added successfully.**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

**3. View Students**

**4. Exit**

**Enter your choice: 1**

**Enter Student Name: Bob**

**Enter Student ID: 102**

**Enter Student Grade: 92.3**

**Student added successfully.**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

**3. View Students**

**4. Exit**

**Enter your choice: 3**

**Student Records:**

**ID: 101, Name: Alice, Grade: 85.5**

**ID: 102, Name: Bob, Grade: 92.3**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

**3. View Students**

**4. Exit**

**Enter your choice: 2**

**Enter Student ID to delete: 101**

**Student with ID 101 deleted successfully.**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

**3. View Students**

**4. Exit**

**Enter your choice: 3**

**Student Records:**

**ID: 102, Name: Bob, Grade: 92.3**

**Student Management System Menu:**

**1. Add Student**

**2. Delete Student**

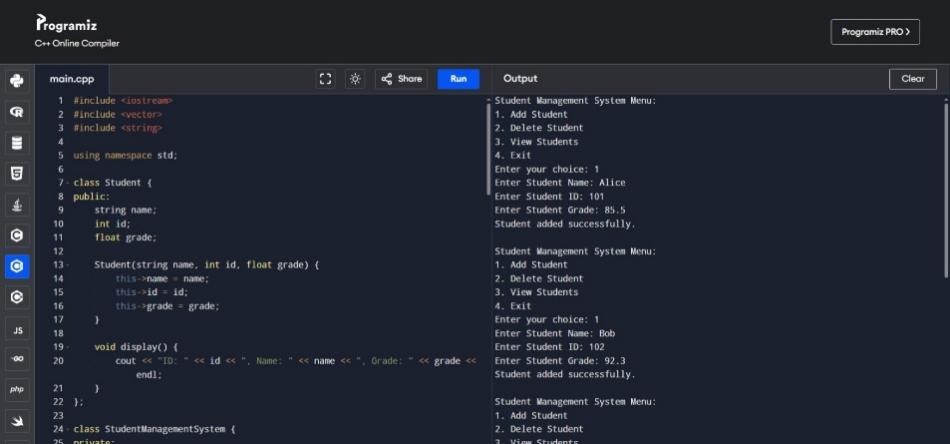
**3. View Students**

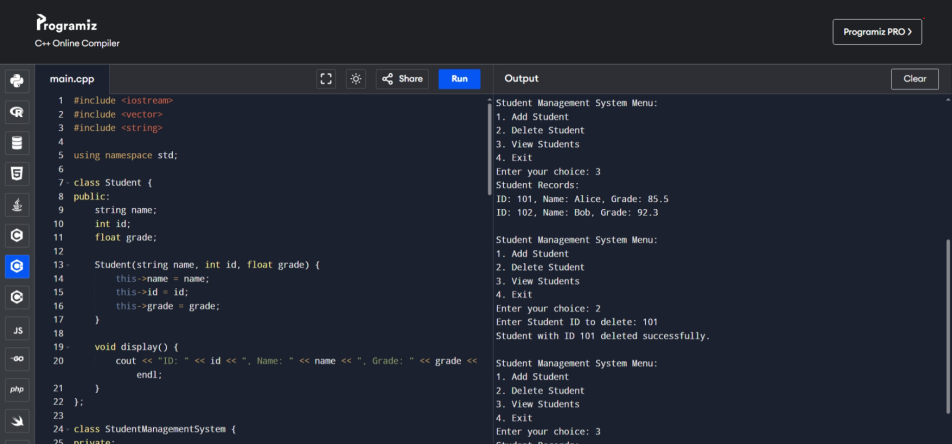
**4. Exit**

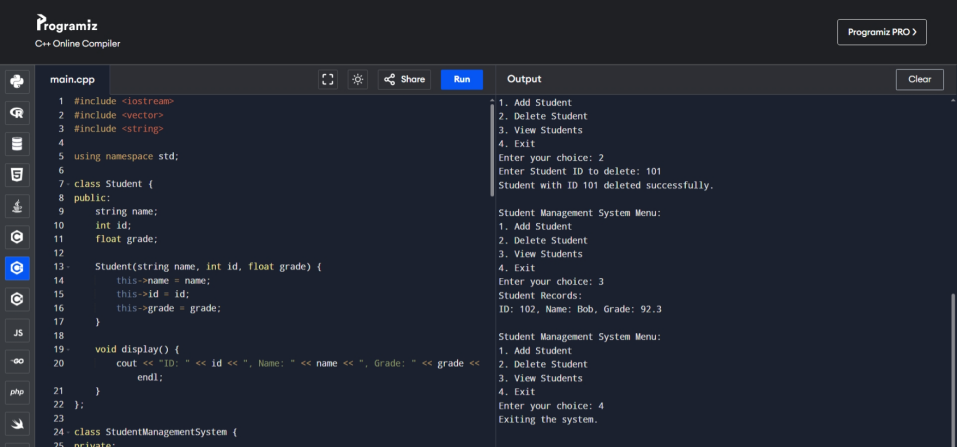
**Enter your choice: 4**

**Exiting the system.**

**OUTPUT:**

****

****

****

### Conclusion

The Student Management System is an essential tool for modern educational institutions, providing a comprehensive solution for managing student data and administrative tasks. By leveraging the features of an SMS, schools, colleges, and universities can improve their operational efficiency, enhance data accuracy, and ultimately deliver a better educational experience to students. This system is not only a practical application for students learning C++ but also a stepping stone towards building more sophisticated educational management systems.