



Student Data Management System with Voting feature

By Doha Fekouss



Requirements

A system which stores student data and allows you to add new data or access, modify and delete current student data.

The system has an embedded voting feature which allows admins to vote for students to receive an award.

The admin portal where student data can be accessed as well as manipulated and votes can be cast are only accessible by a username and password otherwise the system quits.

```

class Student{
public:
    Student();
    Student (int id,string firstname,string surname, string dob,string attendance);
    Student (int id, string firstname, string surname, int votes);
    int getId() const;
    string getFirstName() const;
    string getSurname() const;
    string getDOB() const;
    string getAttendance() const;
    int getVotes() const;
    void setVotes(int vote);
    void setFirstName(string fname);
    void setSurname(string sname);
    void setDOB(string dob);
    void setAttendance(string attendance);
private:
    int studentId;
    string fname;
    string sname;
    string date_of_birth;
    string studentAttendance;
    int noOfVotes;
};

```

The student class contains public methods to get and set the private variables also declared here.

There are variables for the student id, first name, surname, date of birth, attendance percentage and number of votes which is used for the embedded voting feature.

```

void add_student() {
    string firstname, surname, dob, attendance;
    int id;
    id = cnt;
    cout<<"Enter student's first name: ";
    cin>>firstname;
    cout<<"Enter student's surname: ";
    cin>>surname;
    cout<<"Enter student's date of birth: ";
    cin>>dob;
    cout<<"Enter student's attendance percentage: ";
    cin>>attendance;
    studentList.push_back(Student(id,firstname,surname,dob,attendance));
}

```

The add student function simply asks the user for the first name, surname, dob and attendance for the new student they want to add.

The student information is stored in a student list vector.

The display all function loops through each student within the student list vector and prints their id, name, dob and attendance.

```

void displayAll(){
    cout << "\nStudents:" << endl;
    for (const Student& student : studentList) {
        cout <<"Student Id: "<<student.getId() << "\tFirst Name: "<<student.getFirstName()<<
        "\tSurname: "<<student.getSurname()<<"\tDate Of Birth: "<<student.getDOB()
        <<"\tAttendance: "<<student.getAttendance()<<endl;
    }
}

```

```

void getStudentById(){
    int id;
    cout<<"\nEnter Student Id: ";
    cin>>id;
    for (Student& student : studentList) {
        if (student.getId() == id) {
            cout << "Student Id: " << student.getId() << "\tFirst Name: " << student.getFirstName() <<
                "\tSurname: " << student.getSurname() << "\tDate Of Birth: " << student.getDOB()
                << "\tAttendance: " << student.getAttendance() << endl;
        }
    }
}

void deleteById(){
    int id;
    cout<<"\nEnter Student Id: ";
    cin>>id;
    for (int i=0;i<studentList.size();i++){
        if (studentList[i].getId()==id){
            studentList.erase(studentList.begin()+i);
        }
    }
}

```

The get student by id function takes an id input and loops through the student list vector to find the student. It prints the students information.

The delete by id function essentially does the same except it erases said student from the vector list instead of outputting it.

```

void editDetails() {
    int id;
    cout<<"Enter Student Id: ";
    cin>>id;
    for (Student& student : studentList){
        if (student.getId()==id){
            int ch;
            string ch2;
            cout<<"\n1. First Name\n2. Surname\n3. Date of birth\n4. Attendance\n";
            cout<<"Select a choice: \n";
            cin>>ch;
            cout<<"What do you want to change it to?\n";
            cin>>ch2;
            switch(ch) {
                case 1:
                    student.setFirstName(ch2);
                    cout << "Change Made!" << endl;
                    break;
                case 2:
                    student.setSurname(ch2);
                    cout << "Change Made!<<" << endl;
                    break;
                case 3:
                    student.setDOB(ch2);
                    cout << "Change Made!" << endl;
                    break;
                case 4:
                    student.setAttendance(ch2);

```

The edit details function uses a for loop to find the student whose id they entered.

A switch case is used to edit student information based on what they want to change.

```

bool login() {
    string username, password;
    string choice;

    cout << "Are you an admin(yes/no) : ";
    cin >> choice;

    if (choice == "yes") {
        cout << "\n\t\tEnter login credentials for ADMIN PORTAL" << endl;
        cout << "\nEnter username : ";
        cin >> username;
        cout << "Enter password : ";
        cin >> password;

        if (username == "admin" && password == "admin") {
            cout << "\nWELCOME TO ADMIN PORTAL" << endl;
            return true;
        } else {
            cout<<"\nUsername or Password Incorrect!";
            return false;
        }
    } else{
        cout<<"\n Sorry, you do not have access to this system.";
        return 0;
    }
}

```

The login function asks the user if there are an admin and if answered yes, it prompts them to enter their username and password.

It returns a boolean; true if the username and password are correct and false if incorrect.

If details entered are incorrect, program ends else the initial menu is shown.

```

if (login() == true) {
    do {
        cout << "\n-----";
        cout << "\n      STUDENT MANAGEMENT SYSTEM      ";
        cout << "\n-----";
        cout << "\n      1. Add New Student      ";
        cout << "\n      2. Edit Student Details ";
        cout << "\n      3. Delete Student      ";
        cout << "\n      4. Vote For Student Awards ";
        cout << "\n      5. Search For Student  ";
        cout << "\n      6. Return All Students ";
        cout << "\n      7. Exit                ";
        cout << "\n-----\n";

        cout << "Choose An Option: \n";
        cin>>ch;

        switch (ch) {
            case 1:

```

This is the initial menu shown if login() is true.

A switch case is used within a do while loop to allow users to select a option.

The relevant function is called based on the input.


```

case 4:
    do {
        cout << "\n-----";
        cout << "\n      STUDENT VOTING SYSTEM      ";
        cout << "\n-----";
        cout << "\n      1. Academic Excellence      ";
        cout << "\n      2. Best Effort              ";
        cout << "\n      3. Most Improved           ";
        cout << "\n      4. Perfect Attendance      ";
        cout << "\n      5. Friendship              ";
        cout << "\n      6. Back to Main Menu      ";
        cout << "\n-----\n";

        cout << "Choose An Option: \n";
        cin >> ch2;
        switch (ch2) {
            case 1:

```

When the user chooses to vote for student awards, a list of awards is presented.

A switch case within a do while loop is used to provide function for each user choice.

An option is given to return to main menu.

```

void print(vector <Student> list) {
    int ch3,vote,votes,id,highestVotes;
    string firstName, surname;
    bool x=false;
    do {
        cout << "\n-----";
        cout << "\n      STUDENT VOTING SYSTEM      ";
        cout << "\n-----";
        cout << "\n      1. Vote With Student ID      ";
        cout << "\n      2. View Current Nominees     ";
        cout << "\n      3. Nominate Student         ";
        cout << "\n      4. View Winner              ";
        cout << "\n      5. Back To Awards List      ";
        cout << "\n-----\n";
        cout<<"Enter Your Choice: ";
        cin >> ch3;
        switch (ch3) {
            case 1:
                cout << "\nEnter your votes student id: ";
                cin >> vote;
                for (Student &student: list) {
                    if (student.getId() == vote) {
                        x=true;
                        int a = student.getVotes();
                        student.setVotes(a + 1);
                    }
                }
                if (x==false){
                    cout<<"\nInvalid Student Id!";

```

The print function allows user to choose whether they want to vote from current nominees, nominate another student, view current nominees, view winner or return back to awards list.