```cpp
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <algorithm>
using namespace std;

// ------------------------------------------------------
// Auto ID Generator Helper
// ------------------------------------------------------
int generateID() {
    static int counter = 1;
    return counter++;
}

// ------------------------------------------------------
// Student Class
// ------------------------------------------------------
class Student {
public:
    int id;
    string name, dept;
    int year;
    float cgpa;

    Student(int i, string n, string d, int y, float c) {
        id = i;
        name = n;
        dept = d;
        year = y;
        cgpa = c;
    }

    void display() {
        cout << "\nID     : SRM2025" << id;
        cout << "\nName   : " << name;
        cout << "\nDept   : " << dept;
        cout << "\nYear   : " << year;
        cout << "\nCGPA   : " << cgpa;
        cout << "\n----------------------------------------\n";
    }
};

// ------------------------------------------------------
// SRMS System Class
// ------------------------------------------------------
class SRMS {
private:
    vector<Student> students;

public:

    // ----------------------------------------
    // Load Data From File
    // ----------------------------------------
    void loadFromFile() {
        ifstream file("students.txt");
        if (!file) return;

        int id, year;
        float cgpa;
        string name, dept;

        while (file >> id) {
            file.ignore();
            getline(file, name);
            getline(file, dept);
            file >> year >> cgpa;
            students.push_back(Student(id, name, dept, year, cgpa));
        }
        file.close();
    }

    // ----------------------------------------
    // Save Data to File
    // ----------------------------------------
    void saveToFile() {
        ofstream file("students.txt");

        for (auto &s : students) {
            file << s.id << endl;
            file << s.name << endl;
            file << s.dept << endl;
            file << s.year << endl;
            file << s.cgpa << endl;
        }

        file.close();
    }

    // ----------------------------------------
    // Add Student
    // ----------------------------------------
    void addStudent() {
        int id = generateID();
        string name, dept;
    int year;
    float cgpa;

    cin.ignore();
    cout << "\nEnter Name: ";
    getline(cin, name);

    cout << "Enter Department: ";
    getline(cin, dept);

    cout << "Enter Year: ";
    cin >> year;

    cout << "Enter CGPA: ";
    cin >> cgpa;

    students.push_back(Student(id, name, dept, year, cgpa));
    cout << "\nStudent Added Successfully With ID: SRM2025" << id << "\n";

    saveToFile();
}

// ----------------------------------------
// Display All Students
// ----------------------------------------
void displayAll() {
    if (students.empty()) {
        cout << "\nNo students available.\n";
        return;
    }
    cout << "\n---------- STUDENT LIST ----------\n";
    for (auto &s : students)
        s.display();
}

// ----------------------------------------
// Search by ID
// ----------------------------------------
void searchStudent() {
    int id;
    cout << "\nEnter Student ID (Only number after SRM2025): ";
    cin >> id;

    for (auto &s : students) {
        if (s.id == id) {
            cout << "\nStudent Found!\n";
            s.display();
            return;
        }
    }
    cout << "\nStudent Not Found!\n";
}

// ----------------------------------------
// Delete Student
// ----------------------------------------
void deleteStudent() {
    int id;
    cout << "\nEnter ID to delete: ";
    cin >> id;

    for (int i = 0; i < students.size(); i++) {
        if (students[i].id == id) {
            students.erase(students.begin() + i);
            cout << "\nStudent Deleted Successfully!\n";
            saveToFile();
            return;
        }
    }
    cout << "\nStudent Not Found!\n";
}

// ----------------------------------------
// Update Student
// ----------------------------------------
void updateStudent() {
    int id;
    cout << "\nEnter ID to update: ";
    cin >> id;

    for (auto &s : students) {
        if (s.id == id) {
            cin.ignore();
            cout << "\nEnter New Name: ";
            getline(cin, s.name);

            cout << "Enter New Department: ";
            getline(cin, s.dept);

            cout << "Enter New Year: ";
            cin >> s.year;

            cout << "Enter New CGPA: ";
            cin >> s.cgpa;

            cout << "\nUpdated Successfully!\n";
```

```
                saveToFile();
                return;
            }
        }
        cout << "\nStudent Not Found!\n";
    }

    // ----------------------------------------
    // Analytics Dashboard
    // ----------------------------------------
    void showStats() {
        if (students.empty()) {
            cout << "\nNo data available.\n";
            return;
        }

        cout << "\n----- ANALYTICS DASHBOARD -----\n";
        cout << "Total Students: " << students.size() << endl;

        float totalCGPA = 0, maxCGPA = 0;
        string topStudent = "";

        for (auto &s : students) {
            totalCGPA += s.cgpa;
            if (s.cgpa > maxCGPA) {
                maxCGPA = s.cgpa;
                topStudent = s.name;
            }
        }

        cout << "Highest CGPA: " << maxCGPA << " (" << topStudent << ")\n";
        cout << "Average CGPA: " << totalCGPA / students.size() << endl;
    }

    // ----------------------------------------
    // Sorting Feature
    // ----------------------------------------
    void sortMenu() {
        int choice;
        cout << "\n1. Sort by Name";
        cout << "\n2. Sort by CGPA";
        cout << "\n3. Sort by Year\n";
        cout << "Choose: ";
        cin >> choice;

        if (choice == 1)
            sort(students.begin(), students.end(), [](Student a, Student b){ return a.name < b.name;
});
```

```
        else if (choice == 2)
            sort(students.begin(), students.end(), [](Student a, Student b){ return a.cgpa > b.cgpa; });

        else if (choice == 3)
            sort(students.begin(), students.end(), [](Student a, Student b){ return a.year < b.year; });

        cout << "\nSorted Successfully!\n";
        saveToFile();
    }
};

// ------------------------------------------------------------------
// MAIN FUNCTION
// ------------------------------------------------------------------
int main() {
    SRMS system;
    system.loadFromFile();

    int choice;

    while (true) {
        cout << "\n\n========== ADVANCED SRMS ==========";
        cout << "\n1. Add Student";
        cout << "\n2. Display All Students";
        cout << "\n3. Search Student";
        cout << "\n4. Delete Student";
        cout << "\n5. Update Student";
        cout << "\n6. Analytics Dashboard";
        cout << "\n7. Sorting Options";
        cout << "\n8. Exit";
        cout << "\nEnter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: system.addStudent(); break;
            case 2: system.displayAll(); break;
            case 3: system.searchStudent(); break;
            case 4: system.deleteStudent(); break;
            case 5: system.updateStudent(); break;
            case 6: system.showStats(); break;
            case 7: system.sortMenu(); break;
            case 8: cout << "Exiting SRMS..."; return 0;
            default: cout << "Invalid choice!\n";
        }
    }
}
```