

Instructions

You need to create a student management system that stores student information such as Name, RollNumber, and MarksObtained in different subjects. The system should allow the user to perform the following operations:

1. Add a new student record to the system
2. Search for a student record by RollNumber
3. Edit an existing student record
4. Delete an existing student record
5. Display all student records
6. Exit the system

Each student record should be stored in a file, and the file name should be the student's RollNumber.

You can use the following guidelines to implement the system:

1. Create a class named **Student** with the following properties: **Name**, **RollNumber**, and **MarksObtained**. The **MarksObtained** property should be a dictionary where the keys are subject names and the values are the marks obtained by the student in each subject.
2. Create a class named **StudentManagementSystem** with the following methods:
 - **AddStudent()**: This method should prompt the user to enter the student's Name, RollNumber, and MarksObtained in different subjects. It should then create a **Student** object and save the object to a file with the name as the RollNumber.
 - **SearchStudent()**: This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should read the data from the file and display the student's information.
 - **EditStudent()**: This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should allow the user to edit the student's Name, RollNumber, and MarksObtained in different subjects. It should then save the updated information to the file.
 - **DeleteStudent()**: This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should delete the file.
 - **DisplayAllStudents()**: This method should display the information of all students by reading the data from all files.
 - **ExitSystem()**: This method should exit the system.
3. In the main function, create a **StudentManagementSystem** object and use its methods to perform the operations mentioned above.
4. Use file I/O functions to read and write data to files.

Example of input/out put

Student Management System

1. Add a new student record
2. Search for a student record
3. Edit an existing student record
4. Delete an existing student record
5. Display all student records
6. Exit

Enter your choice: 1

Enter student name: John

Enter student RollNumber: 1234

Enter MarksObtained in different subjects:

Maths: 90

Science: 85

English: 95

Student record added successfully!

Enter your choice: 2

Enter student RollNumber: 1234

Name: John

RollNumber: 1234

MarksObtained:

Maths: 90

Science: 85

English: 95

```
Enter your choice: 3

Enter student RollNumber: 1234

Enter updated student name: John Doe
Enter updated student RollNumber: 5678
Enter updated MarksObtained in different subjects:
Maths: 80
Science: 75
English: 85

Student record updated successfully!

Enter your choice: 4

Enter student RollNumber: 1234

Student record deleted successfully!

Enter your choice: 5

Name: Jane
RollNumber: 5678
MarksObtained:
Maths
```

Create readme file & input all the info

tự đánh giá về code mình đã làm trong project vừa rồi?, tại sao mình làm như vậy?. mình có thể làm gì để cải thiện hơn?

self-checklist for this exercise:

Create a class named Student with the following properties:

Name: string

RollNumber: string

MarksObtained: Dictionary<string, int>

Create a class named StudentManagementSystem with the following methods:

AddStudent(): This method should prompt the user to enter the student's Name, RollNumber, and MarksObtained in different subjects. It should then create a Student object and save the object to a file with the name as the RollNumber.

SearchStudent(): This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should read the data from the file and display the student's information.

EditStudent(): This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should allow the user to edit the student's Name, RollNumber, and MarksObtained in different subjects. It should then save the updated information to the file.

DeleteStudent(): This method should prompt the user to enter a RollNumber and search for the corresponding file. If the file is found, it should delete the file.

DisplayAllStudents(): This method should display the information of all students by reading the data from all files.

ExitSystem(): This method should exit the system.

In the main function, create a StudentManagementSystem object and use its methods to perform the operations mentioned above.

Use file I/O functions to read and write data to files.

Test the program thoroughly to ensure that it works correctly for all possible scenarios.