using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StudentdataApp

{

class Student

{

public string Name { get; set; }

public string Class { get; set; }

}

class Program

{

static void Main()

{

List<Student> students = ReadStudentDataFromFile("C:\\Users\\pooja\\OneDrive\\Desktop\\.Net class Assignments\\student\_data.txt");

if (students == null)

{

Console.WriteLine("Error reading the student data file.");

Console.ReadLine();

return;

}

Console.WriteLine("1. Display Student Data");

Console.WriteLine("2. Search for Student by Name");

Console.WriteLine("3. Exit");

while (true)

{

Console.Write("Enter your choice: ");

if (!int.TryParse(Console.ReadLine(), out int choice))

{

Console.WriteLine("Invalid input. Please enter a valid option.");

continue;

}

switch (choice)

{

case 1:

DisplayStudentData(students);

break;

case 2:

SearchStudentByName(students);

break;

case 3:

SaveStudentDataToFile("student\_data.txt", students);

Console.WriteLine("Program exiting. Press Enter to continue...");

Console.ReadLine();

Environment.Exit(0);

break;

default:

Console.WriteLine("Invalid option. Please select a valid option.");

break;

}

}

}

static List<Student> ReadStudentDataFromFile(string fileName)

{

List<Student> students = new List<Student>();

try

{

using (StreamReader reader = new StreamReader(fileName))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] data = line.Split(',');

if (data.Length >= 2)

{

Student student = new Student

{

Name = data[0],

Class = data[1]

};

students.Add(student);

}

}

}

return students;

}

catch (Exception e)

{

Console.WriteLine("An error occurred: " + e.Message);

return null;

}

}

static void DisplayStudentData(List<Student> students)

{

Console.WriteLine("Student Data:");

foreach (var student in students)

{

Console.WriteLine($"Name: {student.Name}, Class: {student.Class}");

}

Console.WriteLine();

}

static void SearchStudentByName(List<Student> students)

{

Console.Write("Enter the name of the student to search: ");

string searchName = Console.ReadLine();

var searchResults = students.Where(s => s.Name.Equals(searchName, StringComparison.OrdinalIgnoreCase)).ToList();

if (searchResults.Any())

{

Console.WriteLine("Search Results:");

foreach (var student in searchResults)

{

Console.WriteLine($"Name: {student.Name}, Class: {student.Class}");

}

}

else

{

Console.WriteLine("No matching students found.");

}

Console.WriteLine();

}

static void SaveStudentDataToFile(string fileName, List<Student> students)

{

using (StreamWriter writer = new StreamWriter(fileName))

{

foreach (var student in students)

{

writer.WriteLine($"{student.Name},{student.Class}");

}

}

}

}

}