using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace RetrieveDataUsingSortingAndSearching

{

class Student

{

public string Name { get; set; }

public string Class { get; set; }

}

internal class Program

{

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

static void Main(string[] args)

{

LoadStudentData();

Console.WriteLine("\n Students Data before Sorting:");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

DisplayStudentData(students);

students.Sort((s1, s2) => string.Compare(s1.Name, s2.Name, StringComparison.OrdinalIgnoreCase));

Console.WriteLine("\nStudent Data After Sorting");

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

DisplaySortedStudentData();

Console.Write("\nEnter a student's name to search: ");

string searchName = Console.ReadLine();

SearchStudentByName(searchName);

Console.ReadKey();

}

static void LoadStudentData()

{

try

{

string filePath = "D:\\simplilearn\\project-3\\Student.txt";

if (File.Exists(filePath))

{

string[] lines = File.ReadAllLines(filePath);

foreach (string line in lines)

{

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

if (data.Length == 2)

{

string name = data[0].Trim();

string studentClass = data[1].Trim();

students.Add(new Student { Name = name, Class = studentClass });

}

}

}

else

{

Console.WriteLine("File not found: students.txt");

}

}

catch (Exception ex)

{

Console.WriteLine("Error loading student data: " + ex.Message);

}

}

static void DisplayStudentData(List<Student> students)

{

foreach (var student in students)

{

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

}

}

static void DisplaySortedStudentData()

{

Console.WriteLine("\nSorted Student Data:");

Console.WriteLine("====================");

foreach (var student in students)

{

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

}

}

static void SearchStudentByName(string searchName)

{

bool found = false;

foreach (var student in students)

{

if (student.Name.Equals(searchName, StringComparison.OrdinalIgnoreCase))

{

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

found = true;

break;

}

}

if (!found)

{

Console.WriteLine("\nStudent not found.");

}

}

}

}