**CODE :-**

using System;

using System.IO;

using System.Collections.Generic;

class Teacher

{

public int ID { get; set; }

public string Name { get; set; }

public string ClassAndSection { get; set; }

}

class Program

{

static void Main(string[] args)

{

List<Teacher> teachers = new List<Teacher>();

string filePath = "teachers.txt";

// Load existing data from file, if any

if (File.Exists(filePath))

{

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

foreach (string line in lines)

{

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

Teacher teacher = new Teacher

{

ID = int.Parse(data[0]),

Name = data[1],

ClassAndSection = data[2]

};

teachers.Add(teacher);

}

}

while (true)

{

Console.WriteLine("1. Add Teacher");

Console.WriteLine("2. Update Teacher");

Console.WriteLine("3. Display All Teachers");

Console.WriteLine("4. Exit");

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

int choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.Write("Enter Teacher ID: ");

int id = int.Parse(Console.ReadLine());

Console.Write("Enter Teacher Name: ");

string name = Console.ReadLine();

Console.Write("Enter Class and Section: ");

string classAndSection = Console.ReadLine();

Teacher newTeacher = new Teacher

{

ID = id,

Name = name,

ClassAndSection = classAndSection

};

teachers.Add(newTeacher);

Console.WriteLine("Teacher added successfully!");

break;

case 2:

Console.Write("Enter Teacher ID to update: ");

int updateID = int.Parse(Console.ReadLine());

Teacher teacherToUpdate = teachers.Find(t => t.ID == updateID);

if (teacherToUpdate != null)

{

Console.Write("Enter Updated Name: ");

teacherToUpdate.Name = Console.ReadLine();

Console.Write("Enter Updated Class and Section: ");

teacherToUpdate.ClassAndSection = Console.ReadLine();

Console.WriteLine("Teacher updated successfully!");

}

else

{

Console.WriteLine("Teacher not found!");

}

break;

case 3:

Console.WriteLine("List of Teachers:");

foreach (Teacher teacher in teachers)

{

Console.WriteLine($"ID: {teacher.ID}, Name: {teacher.Name}, Class and Section: {teacher.ClassAndSection}");

}

break;

case 4:

// Save data to file before exiting

using (StreamWriter writer = new StreamWriter(filePath))

{

foreach (Teacher teacher in teachers)

{

writer.WriteLine($"{teacher.ID},{teacher.Name},{teacher.ClassAndSection}");

}

}

Console.WriteLine("Data saved to file. Exiting program.");

Environment.Exit(0);

break;

default:

Console.WriteLine("Invalid choice. Please try again.");

break;

}

}

}

}