CODE:

Program.cs

```
using phase1Endproject2;
using System;

class Program
{
    static void Main()
    {
        string dataFilePath = "teachers.txt";
        TeacherDataRepository teacherDataRepo = new
TeacherDataRepository(dataFilePath);

        while (true)
        {
            Console.WriteLine("Options:");
            Console.WriteLine("1. Add Teacher");
            Console.WriteLine("2. Display All Teachers");
            Console.WriteLine("3. Update Teacher");
            Console.WriteLine("4. Exit");
            Console.Write("Enter your choice: ");
            string choice = 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();
                    var teacher = new Teacher { ID = id, Name = name, Class =
classAndSection.Split(' ')[0], Section = classAndSection.Split(' ')[1] };
                    teacherDataRepo.AddTeacher(teacher);
                    Console.WriteLine("Teacher added successfully!");
                    break;

                case "2":
                    var allTeachers = teacherDataRepo.GetAllTeachers();
                    if (allTeachers.Count == 0)
                    {
                        Console.WriteLine("No teachers found.");
                    }
                    else
                    {
                        Console.WriteLine("Teachers:");
                        foreach (var t in allTeachers)
                        {
                            Console.WriteLine($"ID: {t.ID}, Name: {t.Name},
Class: {t.Class}, Section: {t.Section}");
                        }
                    }
                    break;

                case "3":
                    Console.Write("Enter Teacher ID to update: ");
                    int updateId = int.Parse(Console.ReadLine());
                    Console.Write("Enter new Teacher Name: ");
                    string newName = Console.ReadLine();
```

```
                    Console.Write("Enter new Class and Section: ");
                    string newClassAndSection = Console.ReadLine();
                    var updatedTeacher = new Teacher { ID = updateId, Name =
newName, Class = newClassAndSection.Split(' ')[0], Section =
newClassAndSection.Split(' ')[1] };
                    teacherDataRepo.UpdateTeacher(updatedTeacher);
                    Console.WriteLine("Teacher updated successfully!");
                    break;

                case "4":
                    Console.WriteLine("Exiting the program.");
                    Environment.Exit(0);
                    break;

                default:
                    Console.WriteLine("Invalid choice. Please try again.");
                    break;
            }

            Console.WriteLine();
        }
    }
}
```

Teacher.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace phase1Endproject2
{
    public class Teacher
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string Class { get; set; }
        public string Section { get; set; }

    }
}
```

TeacherDataRepository.cs

```
using phase1Endproject2;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace phase1Endproject2
{

    public class TeacherDataRepository
    {
        private string dataFilePath;
```

```
        public TeacherDataRepository(string filePath)
        {
            this.dataFilePath = filePath;
        }

        public void AddTeacher(Teacher teacher)
        {
            List<string> lines = new List<string>();
            if (File.Exists(dataFilePath))
            {
                lines = File.ReadAllLines(dataFilePath).ToList();
            }


lines.Add($"{teacher.ID},{teacher.Name},{teacher.Class},{teacher.Section}");
            File.WriteAllLines(dataFilePath, lines);
        }

        public List<Teacher> GetAllTeachers()
        {
            List<Teacher> teachers = new List<Teacher>();
            if (File.Exists(dataFilePath))
            {
                string[] lines = File.ReadAllLines(dataFilePath);
                foreach (string line in lines)
                {
                    string[] parts = line.Split(',');
                    if (parts.Length == 4)
                    {
                        int id = int.Parse(parts[0]);
                        string name = parts[1];
                        string className = parts[2];
                        string section = parts[3];
                        teachers.Add(new Teacher { ID = id, Name = name, Class =
classNam, Section = section });
                    }
                }
            }
            return teachers;
        }

        public void UpdateTeacher(Teacher updatedTeacher)
        {
            List<string> lines = new List<string>();
            if (File.Exists(dataFilePath))
            {
                lines = File.ReadAllLines(dataFilePath).ToList();
            }

            for (int i = 0; i < lines.Count; i++)
            {
                string[] parts = lines[i].Split(',');
                int id = int.Parse(parts[0]);
                if (id == updatedTeacher.ID)
                {
                    lines[i] =
$"{updatedTeacher.ID},{updatedTeacher.Name},{updatedTeacher.Class},{updatedTeache
r.Section}";
                    break;
                }
            }

            File.WriteAllLines(dataFilePath, lines);
```

```
        }
    }
}
```

GIT LINK : https://github.com/sundar2568223/Phase1EndProject-2.git