WRITE UP

Create a Text-file Based System For Storing and Updating Teacher Records

STEP 1: Setting up the Console Application Project
To start, we need to set up the Console Application Project in Visual Studio. We can do this by creating a new Console Application project in Visual Studio and saving it.

STEP 2: Creating the Teacher Class
The next step is to create the Teacher class, which will store the ID, Name, and Class and Section of a teacher. This class will also have a ToString method to format the teacher data for display.
Here is the code for the Teacher class:

```
 public class Teacher
 {
    public int ID { get; set; }
    public string Name { get; set; }
    public string ClassSection { get; set; }
    public Teacher(int id, string name, string classSection)
    {
       ID = id;
       Name = name;
       ClassSection = classSection;
    }
    public override string ToString()
    {
        return $"ID: {ID}, Name: {Name}, Class and Section: {ClassSection}";
    }
}
```

STEP 3: Reading Teacher Data from the Text File
To read the teacher data from the text file, we need to create a method called ReadTeachersFromFile. This method will use the File.ReadAllLines method to read all the lines from the text file and then use the split method to separate the ID, Name, and Class and Section. Each line will be converted into a Teacher object and added to a list.

```
static List<Teacher> ReadTeachersFromFile()
{
    List<Teacher> teachers = new List<Teacher>();
    if (File.Exists(filePath))
    {
        string[] lines = File.ReadAllLines(filePath);
        foreach (string line in lines)
        {
            string[] parts = line.Split(',');
            int id = int.Parse(parts[0]);
            string name = parts[1].Trim();
            string classSection = parts[2].Trim();
            teachers.Add(new Teacher(id, name, classSection));
        }
    }
    return teachers;
}
```

STEP 4: Writing Teacher Data to the Text File

To write the teacher data to the text file, we need to create a method called WriteTeachersToFile. This method will convert each Teacher object in the list into a string and write it to the text file.
Here is the code for the WriteTeachersToFile method:

```
static void WriteTeachersToFile(List<Teacher> teachers)
{
    List<string> lines = new List<string>();
    foreach (Teacher teacher in teachers)
    {
        lines.Add(teacher.ToString());
    }
    File.WriteAllLines(filePath, lines);
}
```

STEP 5: Adding Functionality to the Console Application

We can now add the functionality to the console application. We can do this by implementing methods to add a new teacher, display all teachers, update teacher information, and delete a teacher. We will also need to handle menu options for these functionalities.

STEP 6: Implementing GitHub Tracking and Sharing

Once the application is complete, we can implement GitHub tracking by creating a new repository in GitHub. We will then add the Visual Studio solution to the GitHub repository and push the changes.
After the changes have been pushed, we can share the GitHub repository link in the document.
The provided GitHub repository link should allow anyone to clone the repository and access the complete source code of the console application.