Course End Project 1

**Player and Team Requirements.**

GitHub Link: https://github.com/saraswathy15/PlayerAndTeamPE1

**Overview:**

The FastPace Cricket Academy Player and Team Management System is a console application developed in C# to assist in maintaining information about players participating in one-day cricket games. This documentation provides a simple guide on how to use the system.

**Prerequisites:**

Ensure you have a good understanding of the following concepts in C#:

* Classes
* Interfaces
* Generic Collections
* LINQ (Lambda expressions, Extension methods, and Anonymous Function)

**Functionality:**

**1. Add a Player to the Team**

Users can add a player to the team by providing details such as Player Id, Name, and Age.

**2. Remove a Player from the Team**

Users can remove a player from the team by specifying the player’s Id.

**3. Get Player Details by Id**

Retrieve player details by providing the player’s Id.

**4. Get Player Details by Name**

Retrieve player details by providing the player’s name.

**5. Get All Player Details**

Retrieve details of all players in the team.

**6. Player Limitation**

Users are restricted from adding more than 11 players to the team.

**Class Structure:**

**Player Class**

Represents an individual player with properties for Player Id, Name, and Age.

**Team Class**

Represents the cricket team and provides methods for managing players within the team.

**Limitations:**

* Designed for one-day cricket games; may not be suitable for longer formats.
* Limited error handling for invalid inputs in this version.

**Conclusion:**

The FastPace Cricket Academy Player and Team Management System provides a straightforward solution for managing player information in one-day cricket games. Users are encouraged to explore and adapt the system to meet their specific needs.

**The step-by-step process involved in completing this task.**

1. Created a new project using the "Console App (.NET Framework)" template.

2. Created a Player class with auto-implemented properties.

3. Created an interface ITeam and declared functions as given.

4.Created a derived class OneDayTeam to implement ITeam interface functionalities, and written then Main function inside the Program.cs.

5. Program executed successfully.

6. Then go to top left > view > solution explorer > right click your project > open in file folder > right click > open gitbash here.

7. Initialized a Git repository, added my files using git add . command.

8. And then committed my files

9. Created new repository and finally pushed my code to GitHub.