WRITE UP

Player and Team Requ.irements.

Course-end Project 1

Description

Pre-Requisites: Should have good knowledge of Classes, Interfaces, Generic Collections, and LINQ (Lambda expressions, Extension methods, and Anonymous Function)

Application Environment: Console Application using C#

Case Study:

FastPace Cricket Academy has decided to create a solution to maintain information about the teams' players for one day game with the below functionalities:

User will be able to add a player to the team with details Player Id, Name, and Age.

User will be able to remove a player from the team by passing the player's Id.

User will be able to get player details by passing the player's Id.

User will be able to get player details by passing the player's name.

User will be able to get all player details.

User will not be able to add more than 11 players to the tea

STEP BY STEP PROCESS:

1. Player Class:

   - Defines a Player class with properties PlayerId, Name, and Age.

2. ITeam Interface:

   - Declares an interface ITeam with methods for adding, removing, getting player details by ID and name, and getting all players.

3. OneDayTeam Class (Implements ITeam):

   - Implements the ITeam interface.

   - Contains a static List of players (oneDayTeam) to store player objects.

   - Initializes the team capacity to 11 in the constructor.

4. Program Class:

- Contains the Main method where the program execution starts.
- Creates an instance of OneDayTeam to interact with the team functionalities.

5. User Interaction Loop:

- Uses a while (true) loop for continuous user interaction until the user decides to exit.

6. Switch-Case Menu:

- Presents a menu to the user with options:
  - 1: Add Player
  - 2: Remove Player by Id
  - 3: Get Player By Id
  - 4: Get Player by Name
  - 5: Get All Players

7. User Input and Function Calls:

- Reads the user's choice and performs the corresponding action.
  - Case 1: Adds a player to the team by taking player details as input.
  - Case 2: Removes a player from the team by taking the player's ID as input.
  - Case 3: Gets player details by ID and displays them.
  - Case 4: Gets player details by name and displays them.
  - Case 5: Gets and displays all players in the team.
  - Case 6: Not be able to add more than 11 players to the team.

8. Continue or Exit:

- Asks the user if they want to continue. If the response is not "yes," the loop breaks, and the program exits.

9. Output:

- Displays appropriate messages for successful actions or informs the user if a player is not found or the team is full.

10. Error Handling:

- Minimal error handling is included (e.g., converting user input to integers). You might want to enhance this based on specific requirements.

11. Continuous Interaction:

- The program allows the user to perform multiple actions in sequence, creating, removing, and retrieving player details until the user decides to exit.

12. Documents , write up and Create GitHub Repository.

GITHUB LINK: