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

namespace CricketAcademy
{
   class CricketTeam
   {
      private List<Player> players = new List<Player>();
      static int GetPlayerIdInput()
      {
         Console.WriteLine("Enter Player ID: ");
         return int.Parse(Console.ReadLine());
      }
      static string GetPlayerNameInput()
      {
         Console.WriteLine("Enter Player Name: ");
         string playerName = Console.ReadLine();
         return playerName;
      }
      static void Main(string[] args)
      {
         CricketTeam cricketTeam = new CricketTeam();
         Console.WriteLine("FastPace Cricket Academy - One Day Team\n");
         while (true)
         {
            Console.WriteLine("Choose an option: ");
            Console.WriteLine(" 1. Add Player\n 2. Remove Player\n 3. Get Player Details by Id\n " +
               "4. Get Player Details by Name\n 5. Get All Player Details\n 6. Exit");
            Console.WriteLine("Enter your choice: ");
            int choice = int.Parse(Console.ReadLine());
            switch (choice)
            {
               case 1:
                  int playerId = GetPlayerIdInput();
                  string playerName = GetPlayerNameInput();
                  Console.Write("Enter Player Age: ");
                  int playerAge = int.Parse(Console.ReadLine());
                  cricketTeam.AddPlayer(playerId, playerName, playerAge);
                  break;
               case 2:
                  int playerIdToRemove = GetPlayerIdInput();
                  cricketTeam.RemovePlayer(playerIdToRemove);
                  break;
               case 3:
                  int playerIdToGet = GetPlayerIdInput();
```

```
            Player playerById = cricketTeam.GetPlayerDetailsById(playerIdToGet);
            if (playerById != null)
            {
                Console.WriteLine($"Player Details: ID: {playerById.PlayerId}, Name: {playerById.Name}, Age:
                {playerById.Age}");
            }
            else
            {
                Console.WriteLine($"Player with Name {playerIdToGet} not found.");
            }
            break;
        case 4:
            string playerNameToGet = GetPlayerNameInput();
            Player playerByName = cricketTeam.GetPlayerDetailsByName(playerNameToGet);
            if (playerByName != null)
            {
                Console.WriteLine($"Player Details: ID: {playerByName.PlayerId}, Name: {playerByName.Name},
                Age: {playerByName.Age}");
            }
            else
            {
                Console.WriteLine($"Player with Name {playerNameToGet} not found.");
            }
            break;
        case 5:
            cricketTeam.GetAllPlayerDetails();
            break;
        case 6:
            Environment.Exit(0);
            break;
        default:
            Console.WriteLine("Invalid choice.");
            break;
    }
    Console.WriteLine();
  }
}
public void AddPlayer(int playerId, string name, int age)
{
  if (players.Count < 11)
  {
    Player newPlayer = new Player
    {
      PlayerId = playerId,
      Name = name,
      Age = age
    };
    players.Add(newPlayer);
    Console.WriteLine($"Player {name} added successfully.");
  }
  else
```

```
        {
            Console.WriteLine("Cannot add more than 11 players to the team.");
        }
    }
    public void RemovePlayer(int playerId)
    {
        Player playerToRemove = players.Find(p => p.PlayerId == playerId);
        if (playerToRemove != null)
        {
            players.Remove(playerToRemove);
            Console.WriteLine($"Player {playerToRemove.Name} removed from the team.");
        }
        else
        {
            Console.WriteLine($"Player with ID {playerId} not found in the team.");
        }
    }
    public Player GetPlayerDetailsById(int playerId)
    {
        return players.Find(p => p.PlayerId == playerId);
    }
    public Player GetPlayerDetailsByName(string playerName)
    {
        return players.Find(p => p.Name == playerName);
    }
    public void GetAllPlayerDetails()
    {
        if (players.Count > 0)
        {
            Console.WriteLine("All Players in the team");
            foreach (var player in players)
            {
                Console.WriteLine($"Player Id: {player.PlayerId}, Name: {player.Name}, Age: {player.Age}");
            }
        }
        else
        {
            Console.WriteLine("No players in the team");
        }
    }
  }
}
```