Team.cs

```
using Domain.Interfaces;
using ErrorOr;

namespace Domain;

public class Team
{
    public int Id { get; private set; }
    public string Name { get; private set; }
    public int TrainerId { get; private set; }

    private readonly List<ConfiguredPokemon> _pokemon = new();
    public IReadOnlyCollection<ConfiguredPokemon> Pokemon =>
_pokemon.AsReadOnly();

    // Private constructor for factory method
    private Team() { }

    // Domain behavior with proper validations
    public ErrorOr<Success> AddPokemon(ConfiguredPokemon pokemon)
    {
        if (_pokemon.Count >= 6)
            return Error.Conflict(description: "A team cannot have more than 6
Pokémon");

        _pokemon.Add(pokemon);
        return Result.Success;
    }

    // Factory method for creating from TeamRequest
    public static ErrorOr<Team> CreateFromRequest(TeamRequest request)
    {
        if (string.IsNullOrWhiteSpace(request.Name))
            return Error.Validation(description: "Team name cannot be empty");

        if (request.User <= 0)
            return Error.Validation(description: "User ID must be valid");

        var team = new Team
        {
            Name = request.Name,
            TrainerId = request.User
        };

        foreach (var pokemonRequest in request.Pokemon)
        {
            var pokemonResult =
ConfiguredPokemon.CreateFromRequest(pokemonRequest);
            if (pokemonResult.IsError)
                return pokemonResult.Errors;
            var addResult = team.AddPokemon(pokemonResult.Value);
            if (addResult.IsError)
                return addResult.Errors;
        }

        return team;
    }
}
```

Path: `./Backend/Domain/TeamAggregate/Team.cs`