

# Documentatie Proiect - Prezentarea III

## 1. Structura de directoare

In proiectul WadSocialApp am organizat codul respectand cerintele MVC si Repository Pattern:

- Repositories/
  - Interfaces/: interfetele pentru accesul la date (IPostRepository.cs, IConnectionRepository.cs)
  - Implementations/: implementarile interfetelor (PostRepository.cs, ConnectionRepository.cs)
- Services/
  - Interfaces/: interfetele pentru logica de business (IPostService.cs, IConnectionService.cs)
  - Implementations/: implementarile serviciilor (PostService.cs, ConnectionService.cs)
- Controllers/: Controller-ele folosesc serviciile, nu direct DbContext.
  - PostsController.cs
  - ConnectionsController.cs
- Views/: Am creat pagini pentru fiecare entitate (Posts, Connections):
  - Index.cshtml, Create.cshtml, Edit.cshtml, Delete.cshtml

## 2. Functionalitati implementate

- Operatii CRUD complete pentru Posts si Connections.
- Separare logica de acces la date si servicii (Repository Pattern si Service Layer).
- Configurare corecta pentru Dependency Injection in Program.cs.
- Modele conectate corect la baza de date folosind Entity Framework Core.

## 3. Cum se poate testa aplicatia

Dupa rularea proiectului, paginile pot fi accesate in browser folosind link-urile:

- Posts: <http://localhost:5107/Posts>
- Connections: <http://localhost:5107/Connections>

Se pot crea, edita, sterge si vizualiza entitatile.

## Documentatie Proiect - Prezentarea III

Observatie: Pentru crearea unei conexiuni este necesar ca UserId si FriendId sa corespunda unor utilizatori existenti in baza de date.

### 4. Observatii finale

- Nu s-a implementat functionalitatea de login/register (conform cerintelor).
- Aplicatia respecta structura recomandata si separarea logica a responsabilitatilor.
- Functionalitatea speciala (search) nu este inca adaugata.