

Dezvoltarea Aplicațiilor Web utilizând ASP.NET Core MVC

Laborator 5

OBIECTIVE:

- Crearea modelului de date în cadrul stratului business (Model);
- Configurarea Entity Framework Core și conectarea la baza de date SQL Server;
- Utilizarea Dependency Injection pentru gestionarea serviciilor;
- Crearea și aplicarea migrațiilor pentru generarea bazei de date;
- Validarea și optimizarea modelului folosind un companion cu inteligență artificială;
- Înțelegerea principiilor de normalizare a bazelor de date prin interacțiune asistată de AI;

SCENARIU:

Se va realiza o aplicație ASP.NET Core MVC care gestionează articole și categorii. Fiecare categorie conține mai multe articole, iar fiecare articol aparține unei singure categorii. Vom construi întreaga bază de date de la zero, pas cu pas, și vom folosi un companion AI pentru a valida modelul și a identifica eventuale îmbunătățiri.

EXERCITII:

Exercițiul 1 – Crearea proiectului și configurarea EF Core

1. Creați un nou proiect, numit **Laborator5**, de tip ASP.NET Core Web App (Model-View-Controller).
2. Adăugați pachetele necesare pentru **Entity Framework Core** și **SQL Server** (**VEZI Curs 5 – secțiunea Instalare Entity Framework Core**).

Exercițiul 2 – Crearea modelelor

3. Adăugați cele două modele în baza de date → Article și Category (**VEZI Curs 5 – secțiunea Adăugarea Modelului**).

Se consideră entitățile **Article** și **Category** cu următoarele proprietăți:

Article:

- Id (int – primary key)
- Title (string)
- Content (string)
- Date (DateTime)

Category:

- Id (int – primary key)
- CategoryName (string)

Exemplu implementare:

```
public class Article
{
    [Key]
    public int Id { get; set; }

    public string Title { get; set; }

    public string Content { get; set; }

    public DateTime Date { get; set; }
}
```

La fel se procedează și pentru clasa Category.

Exercițiul 3 – Crearea și configurarea DbContext + Dependency Injection

4. Realizați conexiunea cu baza de date, utilizând **Dependency Injection** (**VEZI Curs 5 – secțiunea Conexiunea cu Baza de Date**).
5. Adăugați baza de date SQL Server (**VEZI Curs 5 – secțiunea Adăugarea unei Baze de Date SQL Server**). Connection String-ul o să se numească **ArticlesDB**.

Exercițiul 4 – Crearea bazei de date (migrații)

6. Integrați sistemul de migrații (**VEZI Curs 5 – secțiunea Crearea migrațiilor în baza de date**).

Exercițiul 5 – Implementare C.R.U.D.

Să se creeze un nou proiect în care să se adauge două Controllere **ArticlesController** și **CategoriesController** în care se vor implementa operațiile CRUD asupra entităților Article, respectiv Category.

ArticlesController o să conțină:

- **Index** – pentru afișarea tuturor articolelor din baza de date;
- **Show** – pentru afișarea unui singur articol;
- **New** – pentru adăugarea unui nou articol în baza de date;
- **Edit** – pentru editarea unui articol existent;
- **Delete** – pentru ștergerea unui articol;

CategoriesController o să conțină:

- **Index** – pentru afișarea tuturor categoriilor din baza de date;
- **Show** – pentru afișarea unei singure categorii;
- **New** – pentru adăugarea unei noi categorii în baza de date;
- **Edit** – pentru editarea unei categorii existente în baza de date;
- **Delete** – pentru ștergerea unei categorii;

7. Implementați CRUD asupra entității **Article** (**VEZI Curs 5 – secțiunea C.R.U.D. utilizând Entity Framework**).
8. Implementați CRUD asupra entității **Category** (**VEZI Curs 5 – secțiunea C.R.U.D. utilizând Entity Framework**).

Exercițiul 6 – Validarea modelului cu Inteligență Artificială

După ce baza de date a fost creată, vom folosi un **companion AI** pentru validarea codului.

Prompt sugerat:

- Verifică dacă acest model EF Core respectă convențiile de denumire și bunele practici.
- Analizează relația dintre Category și Article și spune-mi dacă pot optimiza structura modelului.
- Identifică eventuale probleme de tipuri de date, chei externe sau denumiri care pot fi îmbunătățite.

Exercițiul 7 – Analiza normalizării cu AI

Folosiți același companion AI și trimiteți structura bazei de date cu promptul:

- Analizează structura acestei baze de date (Article, Category).
Este normalizată conform formelor 1NF, 2NF și 3NF?
Explică de ce și cum aș putea îmbunătăți designul.

Exercițiul 8 – Documentarea automată

Prompt final:

- Generează o documentație Markdown pentru clasele Article și Category, explicând rolul fiecărui câmp și relația dintre entități.
Include un scurt rezumat al modelului EF Core.

Documentația generată se poate salva ca fișier .md sau .txt și poate fi inclusă ulterior în proiect.