



**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII
AL REPUBLICII MOLDOVA**

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software i Automatică

Raport

Lucrarea de laborator №4

Disciplina: Tehnologii Web.

**Tema: Accesarea, proiectarea bazelor de date
(EntityFramework)**

Realizat:

Bunescu Gabriel, TI-207

Profesor:

asis. univ. Rusu Cristian.

Scopul lucrării

Crearea și proiectarea la o bază de date utilizând Entity Framework și conectarea la proiectul ASP.NET.

Elaborarea lucrării:

O bază de date este o colecție de date care este stocată în conformitate cu schema de date proiectată. Definiția unei baze de date nu trebuie confundată cu definirea unui sistem de gestionare a bazelor de date (DBMS). DBMS - un set de instrumente software folosite pentru a crea și administra o bază de date. Acest laborator utilizează unul dintre sistemele de gestionare a bazelor de date client-server, Microsoft SQL Server, dezvoltat de Microsoft.

Entity Framework este o tehnologie orientată pe obiecte bazată în .NET framework pentru lucrul cu datele. Acest framework va permite lucrarea cu datele, indiferent de tipul acestora. Astfel, dacă la nivel fizic, dezvoltatorul operează cu tabele, chei, apoi la nivelul furnizat de cadrul entității, el lucrează cu obiecte, ceea ce simplifică foarte mult lucrul cu datele ca întreg. Conceptul central al Entity Framework este conceptul unei entități, care este un set de date asociate unui anumit obiect. Anume din această cauză tehnologia Entity Framework presupune lucrul cu obiectele, nu cu tabelele.

Instalarea NuGet package-ului Entity Framework pentru proiectul Web și BusinessLogic.

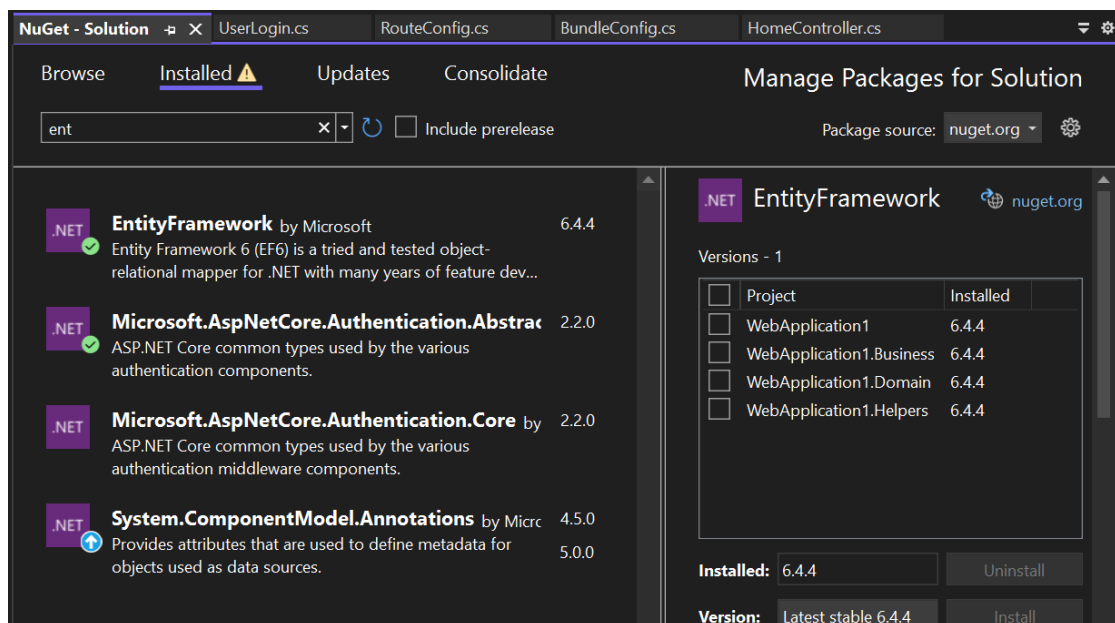


Figura 1 – Instalarea Entity Framework.

Adăugarea referinței System.ComponentModel.DataAnnotations pentru Domain layer.

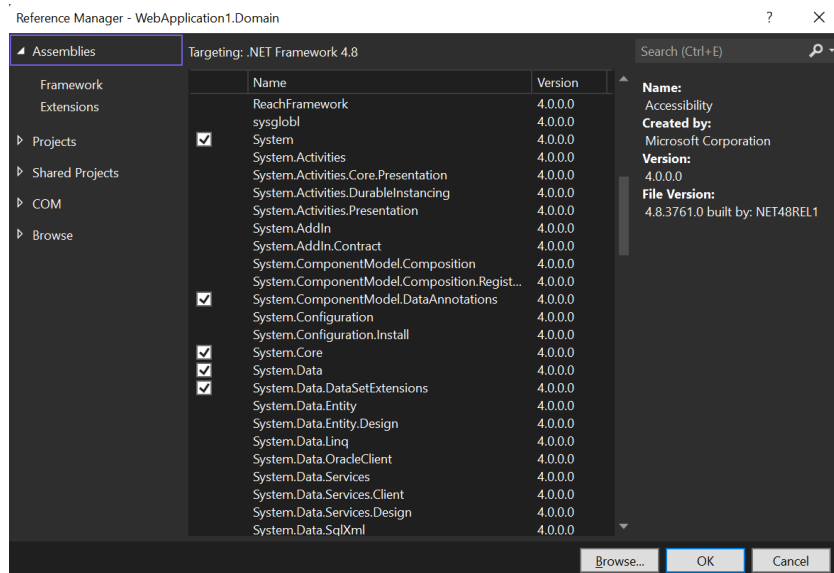


Figura 2 – Manager-ul referințelor pentru Domain.

Crearea acestei clase este una esențială în lucrul cu baza de date, astfel anume în această clasă se vor salva toată informația despre utilizator. După cum se observă, fiecare variabilă este anotată cu unele reguli, cum ar fi, Id primește [Key] ceea ce înseamnă că anume acest câmp va fi cheia principală din baza de date. [Required] reprezintă modul obligatoriu de a fi completat câmpul, [StringLength] indică lungimea maximă și minimă a câmpului.

using System;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using WebApplication1.Domain.Enums;

namespace WebApplication1.Domain.Entities.User

{

public class UDbTable

{

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int Id { **get**; **set**; }

[Required]

[Display(Name = "Username")]

[StringLength(30, MinimumLength = 5, ErrorMessage = "Username cannot be longer than 30 characters.")]

public string Username { **get**; **set**; }

[Required]

[Display(Name = "Password")]

```
[StringLength(50, MinimumLength = 8, ErrorMessage = "Password cannot be shorter than 8 characters."))]
```

```
public string Password { get; set; }
```

```
[Required]
```

```
[Display(Name = "Email Address")]
```

```
[StringLength(30)]
```

```
public string Email { get; set; }
```

```
[DataType(DataType.Date)]
```

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

```
[StringLength(30)]
```

```
public string LasIp { get; set; }
```

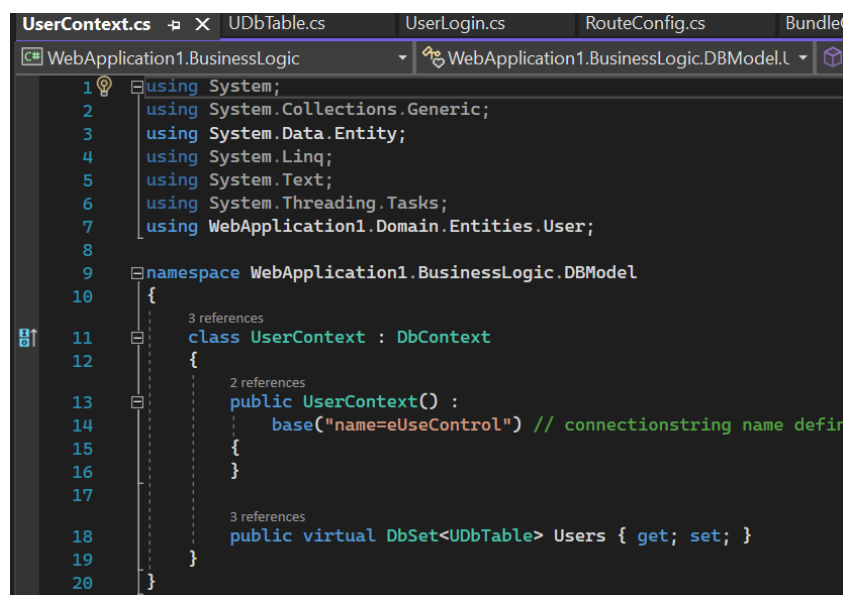
```
public URole Level { get; set; }
```

```
}
```

```
}
```

Crearea fișierului UserContext.cs în proiectul Business Logic

Datorită anumei acestei clase se va executa conexiunea la baza de date, iar datele fiind salvate în clasa UsersDbTable.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Data.Entity;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using WebApplication1.Domain.Entities;
8
9 namespace WebApplication1.BusinessLogic.DBModel
10 {
11     3 references
12     class UserContext : DbContext
13     {
14         2 references
15         public UserContext() :
16             base("name=eUseControl") // connectionstring name defin
17         {
18         }
19
20         3 references
21         public virtual DbSet<UDbTable> Users { get; set; }
```

Figura 3 – UsersDbConext.cs.

Modificarea funcției de logare si inregistrare din Business Logic/UserAPI, pentru a ne conecta la baza de date.

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Data.Entity.Validation;
```

```
using System.Linq;
```

```
using System.Text;
```

```

using System.Threading.Tasks;
using WebApplication1.BusinessLogic.DBModel;
using WebApplication1.Domain.Entities.User;
using WebApplication1.Domain.Enums;
namespace WebApplication1.BusinessLogic.Core
{
    public class UserApi
    {
        internal ULoginResp UserLoginAction(ULoginData data)
        {
            UsersDbTable result;
            using (var db = new UserContext())
            {
                result = db.Users.FirstOrDefault(u => u.Username == data.Credential && u.Password
                == data.Password);
            }
            if (result == null)
            {
                return new ULoginResp { Status = false, StatusMsg = "The username or password is
                incorrect" };
            }
            return new ULoginResp { Status = true };
        }
        internal ULoginResp UserRegisterAction(URegisterData data)
        {
            UsersDbTable result;
            var user = new UsersDbTable();
            user.Email = data.Email;
            user.Username = data.Username;
            user.Password = data.Password;
            user.RegisterDate = data.RegisterDateTime;
            user.Level = URole.Guest;
            try
            {
                using (var db = new UserContext())
                {
                    result = db.Users.FirstOrDefault(u => u.Username == data.Username);
                }
            }
        }
    }
}

```

```

if (result == null)
{
    db.Users.Add(user);
    db.SaveChanges();
    return new ULoginResp { Status = true };
}
else
{
    return new ULoginResp
    { Status = false,
    StatusMsg = "Invalid Username."
    };
} } }
catch (DbEntityValidationException e)
{
    throw;
} } } }

```

Adăugarea conexiunii în Web.config cu baza de date.

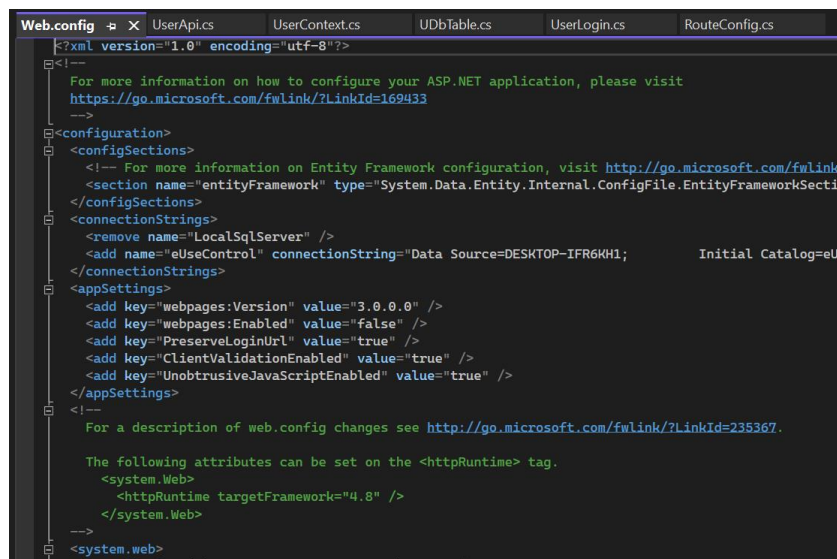


Figura 4 – Web.config.

Mai întâi trebuie de create conexiunea cu baza noastră de date.

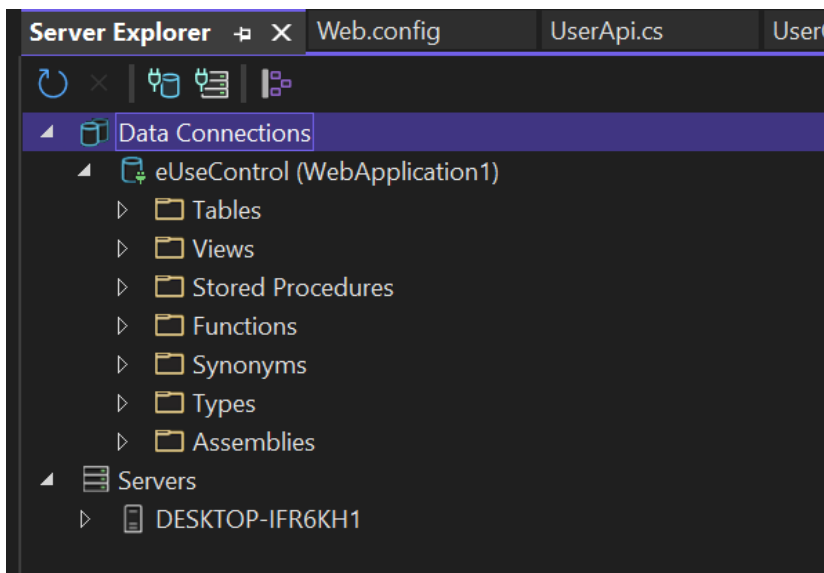


Figura 5 – Cu conexiune la baza de date.

Executăm conectarea la SQL-Server introducând numele calculatorului.

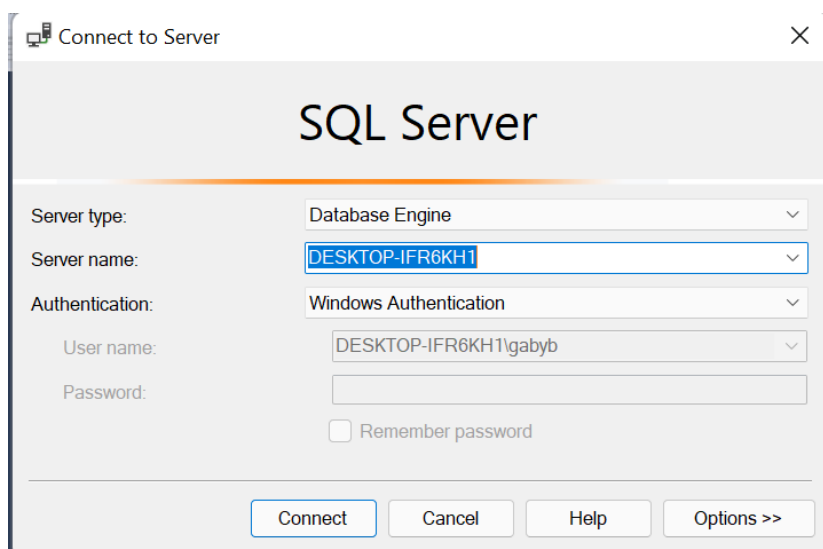


Figura 6 – Conectarea la server.

Verificarea dacă baza de date Microsoft SQL Server Management Studio lucrează și se stochează datele în ea de la înregistrare și logare. Pe care le putem vizualiza în tabelul sub numele UdbTable.

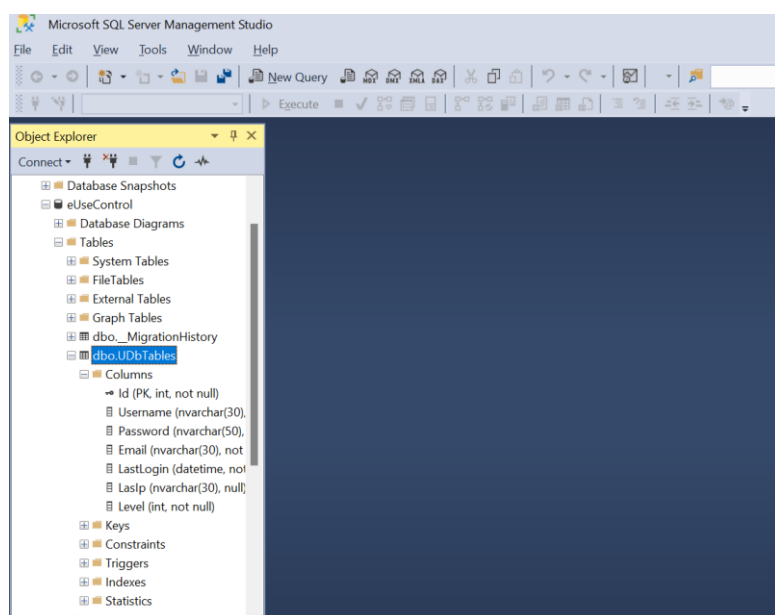


Figura 7 – Vizualizarea bazei de date.

Concluzie:

În urma acestui laborator s-a făcut cunoștință cu Entity Framework, care ne-a ajutat la prelucrarea datelor dintr-o bază de date în cadrul a ASP.NET. Cu ajutorul entităților de date s-a creat o bază de date cu tabele în ea, respective și cu câmpuri care urmează a fi completate de utilizator