

# Dezvoltarea Aplicațiilor Web utilizând ASP.NET Core MVC

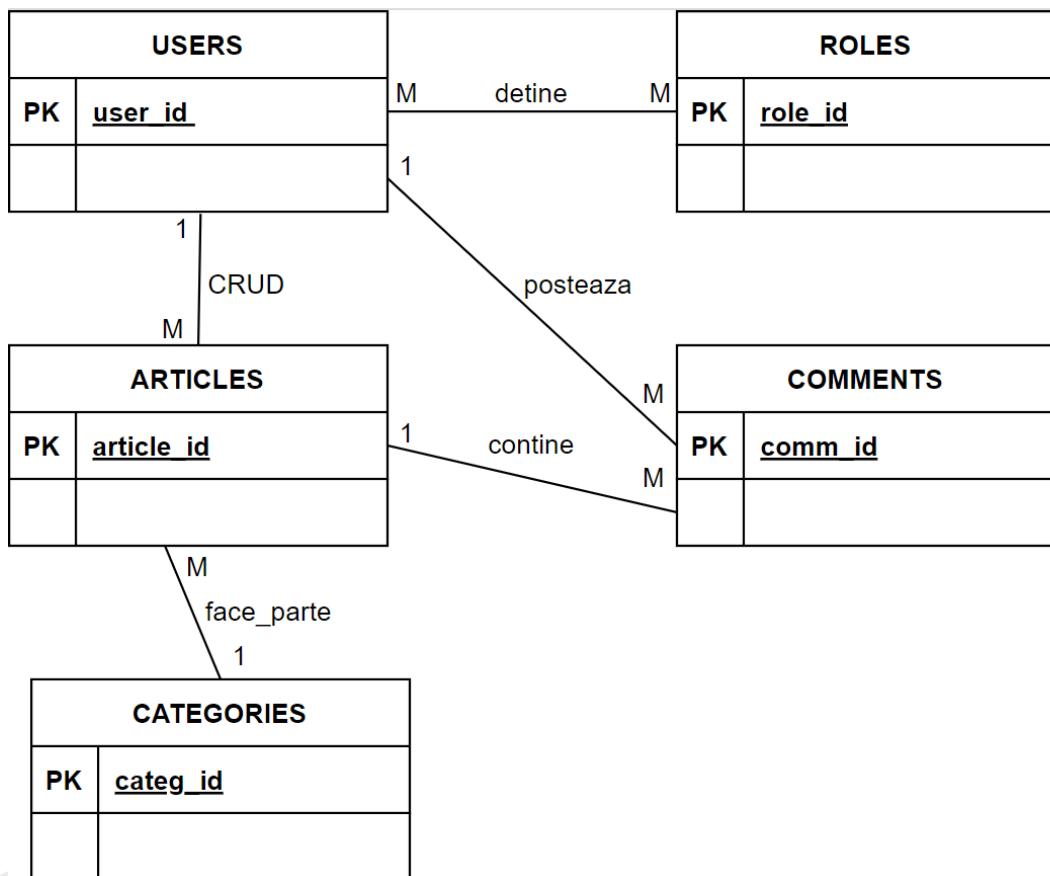
## Laborator 9

### EXERCITII:

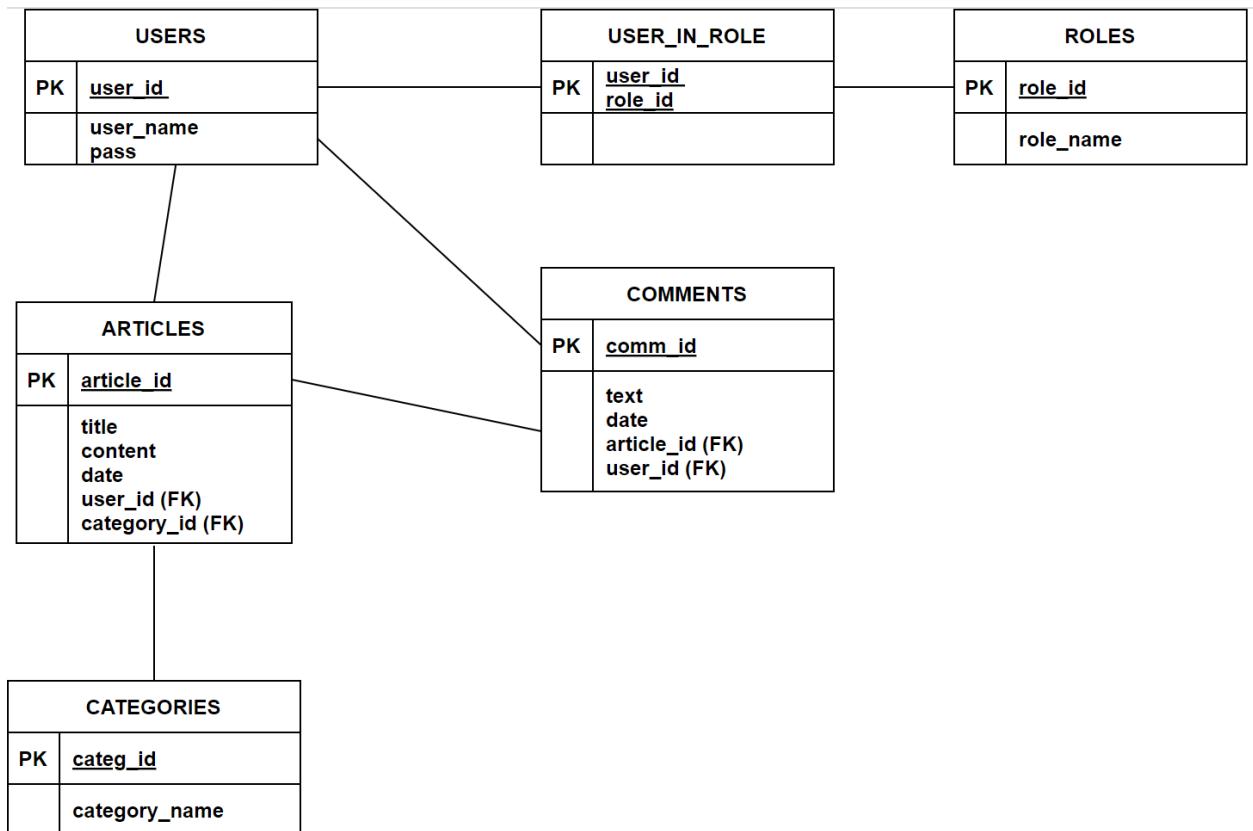
1. Pornind de la versiunea implementată în laboratorul anterior, implementați sistemul de autentificare, împreună cu asocierea dintre roluri și utilizatori (**VEZI Curs 9 – Secțiunea Sistemul de autentificare**). Urmați toți pașii din Cursul 9.
  
2. Se consideră următoarele cerințe:
  - Să existe cel puțin 4 tipuri de utilizatori: vizitator neînregistrat, utilizator înregistrat, editor și administrator;
  - Orice utilizator poate vizualiza știrile apărute pe site. Pe pagina principală vor apărea știrile cele mai recente, în funcție de data la care au post postate. În acest caz, se alege un număr de x știri pentru afișare;
  - Știrile vor fi împărțite pe categorii (create dinamic de către administrator): știință, tehnologie, sport, etc, existând posibilitatea de adăugare a noi categorii (administratorul poate face CRUD pe categorii);
  - Știrile să se afișeze paginat, alegându-se un număr x de știri pe o pagină;
  - O să se includă și un editor de text, astfel încât să se editeze textul în momentul în care un editor publică o nouă știre. Știrea se poate scrie folosind și elemente de markup;
  - Editorii se ocupă de publicarea știrilor noi și pot vizualiza, edita, șterge propriile știri;
  - Utilizatorii care au cont pot adăuga comentarii la știrile apărute, își pot șterge și edita propriile comentarii;
  - Utilizatorii care nu au cont pot să vadă doar pagina principală a aplicației. Dacă doresc să citească știri și să interacționeze cu alții utilizatori, aceștia vor fi redirecționați către pagina de înregistrare/autentificare;

- Știrile pot fi căutate prin intermediul unui motor de căutare propriu, în funcție de titlu, conținut, sau chiar conținutul comentariilor;
- Administratorii se ocupă de buna funcționare a întregii aplicații (ex: pot face CRUD pe știri, pe categorii, pe utilizatori etc.) și pot activa sau revoca drepturile utilizatorilor și editorilor;

### Proiectarea diagramei E/R:



## Diagrama Conceptuală:



## Implementare – modificări asupra entității Article

### Index

Metoda **Index** se ocupă de preluarea tuturor articolelor din baza de date, împreună cu categoria din care fac parte. Toate rolurile trebuie să aibă acces la această metodă. Se utilizează atributul **[Authorize]**.

```
[Authorize(Roles = "User,Editor,Admin")]
```

De asemenea, se utilizează acest atribut și la nivel de Controller, astfel încât doar utilizatorii înregistrați să aibă acces în aplicație.

În momentul afișării articolelor, să se afișeze și utilizatorul care a publicat fiecare articol. Să se modifice View-ul asociat metodei.

Pentru verificare, să se adauge manual, în baza de date, toate intrările necesare.

### Show

Metoda **Show** se ocupă de afișarea unui singur articol, în funcție de id-ul său, împreună cu utilizatorul care a postat articolul respectiv, dar și cu categoria din care face parte. Toate rolurile trebuie să aibă acces la această metodă.

### New

Un articol poate fi adăugat în baza de date doar de utilizatorii cu rolul Editor sau Admin. Metoda **New** se ocupă de adăugarea unui nou articol în baza de date. Doar editorii și administratorii pot adăuga articole în platformă. În momentul adăugării unui articol (*POST*), se reține și Id-ul utilizatorului care a postat articolul.

## Edit

Editorii pot edita propriile știri, iar utilizatorii cu rolul Admin pot edita toate articolele existente în platformă. În momentul în care un utilizator care nu are dreptul la editare încearcă să editeze un articol, acesta va primi un mesaj de tipul: “Nu aveți dreptul să faceți modificări asupra unui articol care nu vă aparține”.

## Delete

Editorii își pot șterge propriile articole, iar utilizatorii cu rolul Admin pot șterge orice articol existent în aplicație. În cazul în care un utilizator rău intenționat încearcă să șteargă un articol, acesta va primi un mesaj corespunzător: “Nu aveți dreptul să ștergeți un articol care nu vă aparține”.

3. Să se modifice codul existent în **Show**, atât în ArticlesController, cât și în View-ul Show asociat, astfel încât butoanele de **editare** și **stergere** să fie vizibile doar pentru utilizatorii cu rolul Admin și Editor. Administratorul poate avea butoanele vizibile pentru orice articol, iar Editorul o să le aibă vizibile doar pentru articolele care îi aparțin. Utilizatorul cu rolul User nu o să vadă aceste butoane.
4. Să se modifice View-urile **Index** și **Show** din **ArticlesController**, astfel încât să nu mai existe legătură cu paginile de afișare a tuturor articolelor, a tuturor categoriilor sau cu pagina de adăugare a unui nou articol. Aceste legături o să fie adăugate în meniul din **Shared -> \_Layout.cshtml**.

Utilizatorii cu rolul **User** vor vedea în meniu doar link-ul “Afișare articole”, utilizatorii cu rolul **Editor** vor vedea în meniu link-urile “Afișare articole” și „Adăugare articol”, iar utilizatorii cu rolul **Admin** vor vedea în aplicație “Afișare articole”, “Adăugare articol” și “Afișare categorii”.

Se utilizează în pagina `_Layout.cshtml` următoarele verificări, după caz:

```
@if (User.IsInRole("Admin")) { ... }
```

```
@if (User.IsInRole("Editor")) { ... }
```

Pentru stilizarea meniului, se poate utiliza codul implementat în laboratorul anterior:

```
.navbar {
    background-color: #dfe6e9 !important;
    border: none !important;
    box-shadow: 0 3px 12px #636e72 !important;
}

.a.nav-link {
    color: #2c2c2c !important;
}

.a.nav-link:hover {
    color: #00b894 !important;
}

.logoutbtn {
    color: #2c2c2c !important;
}

.logoutbtn:hover {
    color: #00b894 !important;
}
```

## Implementare – modificări asupra entității Category

Utilizatorii cu rolul **Admin** sunt cei care pot face C.R.U.D. pe categorii.

- Știrile vor fi împărțite pe categorii (create dinamic de către administrator): știință, tehnologie, sport, etc, existând posibilitatea de adăugare a noi categorii (administratorul poate face CRUD pe categorii);
- Administratorii se ocupă de buna funcționare a întregii aplicații (ex: pot face CRUD pe știri, pe categorii, pe utilizatori etc.) și pot activa sau revoca drepturile utilizatorilor și editorilor;

Modificați **CategoriesController**, astfel încât să implementați funcționalitățile de mai sus.

## Implementare – modificări asupra entității Comment

- Utilizatorii care au cont pot adăuga comentarii la știrile apărute, își pot șterge și edita propriile comentarii;

### **Explicație:**

Toți utilizatorii autenticați pot lăsa comentarii, indiferent de rol. Fiecare utilizator poate edita și șterge doar comentariile proprii, iar administratorul poate edita și șterge orice comentariu.

### **! OBSERVAȚIE**

Clasa  **ApplicationUser**  poate conține atribute suplimentare față de cele oferite de framework. De asemenea, trebuie definite și proprietățile specifice:

```
public class ApplicationUser : IdentityUser
{
    public virtual ICollection<Comment>? Comments { get; set; }

    public virtual ICollection<Article>? Articles { get; set; }
}
```