Dezvoltarea Aplicatiilor Web utilizand ASP.NET Core MVC
Laborator 9

EXERCITII:

1. Pornind de la versiunea implementata in laboratorul anterior, implementati sistemul de autentificare impreuna cu asocierea dintre roluri si utilizatori (**VEZI Curs 9 – Sectiunea Sistemul de autentificare**). Urmati toti pasii din Cursul 9.

2. Se considera urmatoarele cerinte:

- sa existe cel putin 4 tipuri de utilizatori: vizitator neinregistrat, utilizator inregistrat, editor si administrator;
- orice utilizator poate vizualiza stirile aparute pe site. Pe pagina principala vor aparea stirile cele mai recente;
- stirile vor fi impartite pe categorii (create dinamic): stiinta, tehnologie, sport, etc, existand posibilitatea de adaugare a noi categorii (administratorul poate face CRUD pe categorii);
- editorii se ocupa de publicarea stirilor noi si pot vizualiza, edita, sterge propriile stiri;
- utilizatorii pot adauga comentarii la stirile aparute, isi pot sterge si edita propriile comentarii;
- stirile pot fi cautate prin intermediul unui motor de cautare propriu;
- administratorii se ocupa de buna functionare a intregii aplicatii (ex: pot face CRUD pe stiri, pe categorii, etc.) si pot activa sau revoca drepturile utilizatorilor si editorilor;

Implementare – modificari asupra entitatii Article

Index

Metoda Index se ocupa de preluarea tuturor articolelor din baza de date, impreuna cu categoria din care fac parte. Toate rolurile trebuie sa aiba acces la acesta metoda. Se utilizeaza atributul **[Authorize]**.

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

De asemenea, se utilizeaza acest atribut si la nivel de Controller, astfel incat doar utilizatorii inregistrati sa aiba acces in aplicatie.

In momentul afisarii articolelor, sa se afiseze si utilizatorul care a publicat fiecare articol. Sa se modifice View-ul asociat metodei.

Pentru verificare sa se adauge manual in baza de date toate intrarile necesare.

Show

Metoda Show se ocupa se afisarea unui singur articol, in functie de id-ul sau, impreuna cu utilizatorul care a postat articolul respectiv, dar si cu categoria din care face parte. Toate rolurile trebuie sa aiba acces la aceasta metoda.

New

Un articol poate fi adaugat in baza de date doar de utilizatorii cu rolul Editor sau Admin. Metoda New se ocupa de adaugarea unui nou articol in baza de date. Doar editorii si administratorii pot adauga articole in platforma. In momentul adaugarii unui articol (POST), se retine si id-ul userului care a postat articolul.

Edit

Editorii pot edita propriile stiri, iar utilizatorii cu rolul Admin pot edita toate articolele existente in platforma. In momentul in care un utilizator care nu are dreptul la editare incearca sa editeze un articol, acesta o sa primeasca un mesaj de tipul “Nu aveti dreptul sa faceti modificari asupra unui articol care nu va apartine”.

Delete

Editorii isi pot sterge propriile articole, iar utilizatorii cu rolul Admin pot sterge orice articol existent in aplicatie. In cazul in care un utilizator rau intentionat incearca sa stearga un articol, acesta o sa primeasca un mesaj corespunzator: “Nu aveti dreptul sa stergeti un articol care nu va apartine”.

3. Sa se modifice codul existent in **Show**, atat in ArticlesController, cat si in View-ul Show asociat, astfel incat butoanele de **editare** si **stergere** sa fie vizibile doar pentru utilizatorii cu rolul Admin si Editor. Administratorul poate avea butoanele vizibile pentru orice articol, iar Editorul o sa le aiba vizibile doar pentru articolele care ii apartin. Utilizatorul cu rolul User nu o sa vada aceste butoane.

4. Sa se modifice View-urile **Index** si **Show** din **ArticlesController**, astfel incat sa nu mai existe legatura catre paginile de afisare a tuturor articolelor, a tuturor categoriilor sau catre pagina de adaugare a unui nou articol. Vom proceda astfel: aceste legaturi o sa fie adaugate in meniul din Shared -> _Layout.cshtml.

Utilizatorii cu rolul **User** o sa vada in meniu doar link-ul “Afisare articole”, utilizatorii cu rolul **Editor** o sa vada in meniu link-urile “Afisare articole” si “Adaugare articol”, iar userul cu rolul **Admin** o sa vada in aplicatie “Afisare articole”, “Adaugare articol”, “Afisare categorii”.

Se utilizeaza in pagina **_Layout.cshtml** urmatoarele verificari, dupa caz:

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

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

Pentru stilizarea meniului putem utiliza codul implementat in 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 – modificari asupra entitatii Category

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

- stirile vor fi impartite pe categorii (create dinamic): stiinta, tehnologie, sport, etc, existand posibilitatea de adaugare a noi categorii (administratorul poate face CRUD pe categorii);

- administratorii se ocupa de buna functionare a intregii aplicatii (ex: pot face CRUD pe stiri, pe categorii, etc.)

Modificati CategoriesController, astfel incat sa implementati functionalitatea de mai sus.

Implementare – modificari asupra entitatii Comment

Toti utilizatorii autentificati pot lasa comentarii, indiferent de rol. Fiecare utilizator poate edita si sterge doar comentariile proprii, iar administratorul poate edita si sterge orice comentariu.

/!\ OBSERVATIE

Clasa ApplicationUser poate contine atribute suplimentare fata de cele oferite de framework. De asemenea, trebuie definite si proprietatile specifice:

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

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