

## ***Dezvoltarea Aplicatiilor Web-Anul 3***

### ***Laborator 8***

---

#### **Exercitiul 1**

Se considera entitatile Article si Category cu urmatoarele proprietati:

##### **Article**

- Id
- Title
- Content
- Date
- CategoryId

##### **Category:**

- CategoryId
- CategoryName

Sa se creeze modelele corespunzatoare, contextul pentru baza de date si relatia dintre aceste modele folosind Entity Framework.

Sa se implementeze C.R.U.D. pentru modelul de Articole impreuna cu Categoria acestuia. Un articol poate sa aiba o singura categorie, iar o categorie poate fi atasata la mai multe articole. Sa se implementeze componentele astfel:

- sa se adauge controller-ul necesar pentru procesarea categoriilor(CategoriesController)
  - Metoda Index afiseaza toate categoriile
  - Metoda Show afiseaza o singura categorie
  - Metoda New creeaza o noua categorie
  - Metoda Edit modifica numele unei categorii
  - Metoda Delete sterge o categorie din baza de date

- sa se adauge controller-ul necesar pentru procesarea articolului (ArticleController):
  - Metoda Index afiseaza toate articolele din baza de date impreuna cu categoria articolului

Exemplu:

```
public ActionResult Index()
{
    //var articles = from article in db.Articles
    //                select article;
    // Solutia: Eager loading
    var articles = db.Articles.Include("Category");
    ViewBag.Articles = articles;

    return View();
}

@foreach (var article in ViewBag.Articles)
{
    <p>@article.Title</p>
    <p>@article.Content</p>
    <p>@article.Date</p>
    <p>@article.Category.CategoryName</p>
    <a href="/Article/Show/@article.Id">Afisare articol</a>
    <br />
    <hr />
    <br />
}
<a href="/Categories/Index/">Afisare lista categorii</a>
<br />
<a href="/Article/New">Adauga articol</a>
<br />
```

- Metoda Show afiseaza datele unui articol si categoria acestuia

```
public ActionResult Show(int id)
{
    Article article = db.Articles.Find(id);
    ViewBag.Article = article;
    ViewBag.Category = article.Category;

    return View();
}
```

```

<h2>Afisare articol</h2>
<p>@ViewBag.Article.Title</p>
<p>@ViewBag.Article.Content</p>
<p>@ViewBag.Article.Date</p>

<p>Categorie: @ViewBag.Category.CategoryName</p>

<a href="/Article/Edit/@ViewBag.Article.Id">Modifica articol</a>

<form method="post" action="/Article/Delete/@ViewBag.Article.Id">
    @Html.HttpMethodOverride(HttpVerbs.Delete)
    <button type="submit">Sterge articol</button>
</form>

<hr />
<a href="/Article/Index">Inapoi la articole</a>
<br />
<a href="/Article/New">Adauga articol</a>
<br />

```

- In adaugarea de articol sa se selecteze dintr-un dropdown (element html select) o categorie existenta in baza de date; data articolului sa fie data si ora din momentul adaugarii

```

public ActionResult New()
{
    var categories = from cat in db.Categories select cat;
    ViewBag.Categories = categories;
    return View();
}

[HttpPost]
public ActionResult New(Article article)
{
    //TODO
}

<form method="post" action="/Article/New">
    <label>Titlu articol</label>
    <input type="text" name="Title" />
    <br />
    <label>Continut articol</label>
    <textarea name="Content"></textarea>
    <br />
    <input type="hidden" name="Date" value="@DateTime.Now" />
    <br />
    <label>Selectati categoria</label>
    <select name="CategoryId">

```

```

        @foreach (var category in ViewBag.Categories)
        {
            <option
value="@category.CategoryId">@category.CategoryName</option>
        }
    </select>
    <br />
    <button type="submit">Adauga articol</button>
</form>

```

- In modificarea articolului sa se afiseze datele articolului si categoria anterioara a acestuia (deja selectata); la modificarea articolului sa se modifice si data de creare a articolului

```

public ActionResult Edit(int id)
{
    Article article = db.Articles.Find(id);
    ViewBag.Article = article;
    ViewBag.Category = article.Category;
    var categories = from cat in db.Categories select cat;
    ViewBag.Categories = categories;
    return View();
}

```

```

[HttpPut]
public ActionResult Edit(int id, Article requestArticle)
{
    try
    {
        Article article = db.Articles.Find(id);
        if (TryUpdateModel(article))
        {
            article.Title = requestArticle.Title;
            article.Content = requestArticle.Content;
            article.Date = requestArticle.Date;
            article.CategoryId = requestArticle.CategoryId;
            db.SaveChanges();
        }
        return RedirectToAction("Index");
    }
    catch (Exception e)
    {
        return View();
    }
}

```

```

<form method="post" action="/Article/Edit/@ViewBag.Article.Id">
    @Html.HttpMethodOverride(HttpVerbs.Put)
    <label>Titlu articol</label>
    <br />
    <input type="text" name="Title" value="@ViewBag.Article.Title" />
    <br /><br />
    <label>Continut articol</label>
    <br />
    <input type="text" name="Content" value="@ViewBag.Article.Content" />
    <br /><br />
    <label>Data</label>
    <br />
    <input type="text" name="Date" value="@ViewBag.Article.Date" />
    <br />
    <select name="CategoryId">
        @foreach (var category in ViewBag.Categories)
        {
            <option
                @{ if (category.CategoryId ==
ViewBag.Category.CategoryId) {
                    @:selected="selected"
                }
            >
value="@category.CategoryId">@category.CategoryName</option>
        }
    </select>
    <br />
    <button type="submit">Modifica articol</button>
</form>

```

- Posibilitatea stingerii unui articol

```

[Http...]
public ActionResult Delete(int id)
{
    //TODO
}

```