Exercitiul 1

Se considera baza de date, cu cele doua modele Article.cs si Category.cs, din Laborator 8. Modificati atat View-urile cat si metodele si Controllere (atunci cand este necesar) astfel incat sa se utilizeze helpere pentru View. Pentru exemple – Vezi Curs 9.

Exemplu:

ArticleController – Index

```
public ActionResult Index()
      {
     // var articles = from article in db.Articles
      //               select article;
      // Solutia: Eager loading
      var articles = db.Articles.Include("Category");
      if (TempData.ContainsKey("message"))
      {
          ViewBag.message = TempData["message"].ToString();
      }
      ViewBag.Articles = articles;

      return View();
      }
```

Index.cshtml

```
@{
    ViewBag.Title = "Afisare articole";
}

<h2>@ViewBag.Title</h2>

@if (TempData.ContainsKey("message"))
{
    <h3 class="alert alert-info" role="alert">@ViewBag.message</h3>
}


@foreach (CursLab8.Models.Article article in ViewBag.Articles)
{
    @Html.Partial("ArticleInfo", article);

}

<a class="btn btn-sm btn-success" href="/Article/New">Adauga un articol</a>
<br />
```

“ArticleInfo” este un Partial, adaugat in folderul Shared si care contine urmatoarea secventa de cod:

```
<div class="panel panel-default">
    <div class="panel-heading">@Model.Title</div>
    <div class="panel-body">
        Continut articol: <strong>@Model.Content</strong>
        <br />
        <span class="label label-success">@Model.Date</span>
        <br />
        <i class="glyphicon glyphicon-globe"></i>
@Model.Category.CategoryName
    </div>
    <div class="panel-footer">
        <a class="btn btn-sm btn-success"
href="/Article/Show/@Model.Id">Afisare articol</a>
    </div>
</div>
<br />
```

ArticleController – New

```
public ActionResult New()
  {
      Article article = new Article();
      // preluam lista de categorii din metoda GetAllCategories()
      article.Categories = GetAllCategories();
      return View(article);
  }

  [NonAction] //pentru aceasta metoda si dropdown (Vezi curs Pag 16-18)
  public IEnumerable<SelectListItem> GetAllCategories()
  {
      // generam o lista goala
      var selectList = new List<SelectListItem>();
      // Extragem toate categoriile din baza de date
      var categories = from cat in db.Categories select cat;
      // iteram prin categorii
      foreach(var category in categories)
```

```
            {
                // Adaugam in lista elementele necesare pentru dropdown
                selectList.Add(new SelectListItem
                {
                    Value = category.CategoryId.ToString(),
                    Text = category.CategoryName.ToString()
                });
            }
            // returnam lista de categorii
            return selectList;
        }


        [HttpPost]
        public ActionResult New(Article article)
        {
            article.Categories = GetAllCategories();
            try
            {
                if (ModelState.IsValid)
                {
                    db.Articles.Add(article);
                    db.SaveChanges();
                    TempData["message"] = "Articolul a fost adaugat!";
                    return RedirectToAction("Index");
                }
                else
                {
                    return View(article);
                }
            }
            catch (Exception e)
            {
                return View(article);
            }
              }
```

New.cshtml (Validarile se pot adauga la final)

```
      @model CursLab8.Models.Article
@{
    ViewBag.Titlu = "Creare articol";
}

<h2>@ViewBag.Titlu</h2>
<form method="post" action="/Article/New">
    @Html.ValidationSummary(false, "", new { @class = "text-danger" })
    @Html.Label("Title", "Titlu Articol")
    <br />
    @Html.TextBox("Title", null, new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.Title, "Titlul este 
obligatoriu", new { @class = "text-danger " })
    <br /><br />
    @Html.Label("Content", "Continut Articol")
    <br />
    @Html.TextArea("Content", null, new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.Content, "Continutul este 
obligatoriu", new { @class = "text-danger " })
    <br /><br />
    @Html.Hidden("Date", DateTime.Now, new { @class = "form-control" })
    <label>Selectati categoria</label>
    @Html.DropDownListFor(m => m.CategoryId, new SelectList(Model.Categories, 
"Value", "Text"), "Selectati categoria", new { @class = "form-control" })
    <br />
    <button class="btn btn-sm btn-success" type="submit">Adauga 
articol</button>
</form>
```

CategoriesController – Index

```
public ActionResult Index()
     {
         var categories = from category in db.Categories
                          orderby category.CategoryName
                          select category;

         ViewBag.Categories = categories;

         if (TempData.ContainsKey("message"))
         {
             ViewBag.message = TempData["message"].ToString();
         }
         return View();
        }
```

Index.cshtml

```
@{
    ViewBag.Titlu = "Afisare categorii";
}

<h2>@ViewBag.Titlu</h2>

@if (TempData.ContainsKey("message"))
{
    <h3 class="alert alert-info" role="alert">@ViewBag.message</h3>
}

@foreach (var category in ViewBag.Categories)
{
    <p>@category.CategoryName</p>
    <a class="btn btn-sm btn-success"
href="/Categories/Show/@category.CategoryId">Afisare categorie</a>
    <br />
    <hr />
    <br />
}

<a class="btn btn-sm btn-success" href="/Categories/New">Adauga categorie</a>
<br />
```

CategoriesController – New

```
public ActionResult New()
     {
         return View();
     }

     [HttpPost]
     public ActionResult New(Category category)
     {
         try
         {
             db.Categories.Add(category);
             db.SaveChanges();
             TempData["message"] = "Categoria a fost adaugata!";
             return RedirectToAction("Index");
         }
         catch (Exception e)
         {
             return View();
         }

        }
```

New.cshtml

```
@model CursLab8.Models.Category
@{
    ViewBag.Title = "Adaugare categorie";
}

<h2>@ViewBag.Title</h2>

<form method="post" action="/Categories/New">
    @Html.Label("CategoryName", "Denumire categorie")
    <br />
    @Html.TextBox("CategoryName", null, new { @class = "form-control" })
    @Html.ValidationMessageFor(model => model.CategoryName, null, new {
@class = "text-danger " })
    <br /><br />
    <button class="btn btn-sm btn-success" type="submit">Adauga
categorie</button>
</form>
```

La final sa se adauge validari astfel:

- Pentru modelul **Article** – Title si Content sa fie obligatorii. Adaugati aceste validari folosind helper-ul `@Html.ValidationMessageFor`

Exemplu(Curs 9 – pag 22):

```
  @Html.ValidationSummary(false, "", new { @class = "text-danger"
})
  @Html.Label("Title", "Titlu Articol")
 <br />
  @Html.TextBox("Title", null, new { @class = "form-control" })
  @Html.ValidationMessageFor(model => model.Title, "Titlul este
  obligatoriu", new { @class = "text-danger " })
```

- Pentru modelul **Category** – CategoryName sa fie obligatoriu. Adaugati aceasta validare completand mesajul la nivel de Model astfel:

```
[Required(ErrorMessage = "Numele categoriei este obligatoriu")]
```

- **Pe baza exemplelor din curs si laborator dezvoltati intreaga aplicatie.**