---------------------WEB application layer---------------------------------------------------------------

***Představení použitých entit***

MODEL:HeaderEntity

using System.ComponentModel.DataAnnotations;

namespace Web.AL.Models

{

public class HeaderEntity

{

[Display(Name ="HeadLine")]

[StringLength(50,ErrorMessage ="Headline darf nicht länger als 50 Zeichen sein")]

[Required(ErrorMessage ="HeadLine is verpflichtet")]

public string Title { get; set; }

[Display(Name = "Wortlaut\_des\_Kopfes")]

[StringLength(1500, ErrorMessage = "Höchslänge 1500 Zeichen wurde überschritten")]

[DataType(DataType.MultilineText)]

public string Text { get; set; }

[Display(Name ="Link 1")]

public LinkHeader Link1 { get; set; }

[Display(Name ="Link2")]

public LinkHeader Link2 { get; set; }

}

public class LinkHeader

{

[Display(Name = "Link\_Text")]

[StringLength(50, ErrorMessage = "Wortlaut des TextsLinkes darf nicht länger als 50 Zeichen sein")]

public string LinkText { get; set; }

[Display(Name ="AdresseLink")]

[StringLength(100, ErrorMessage = "Wortlaut des AdresseLinkes darf nicht länger als 100 Zeichen sein")]

public string LinkAddress { get; set; }

}

}

MODEL:TeaserEntity

using System;

using System.ComponentModel.DataAnnotations;

namespace Web.AL.Models

{

public class TeaserEntity

{

[Display(Name ="Identifizierungsnummer")]

public int Id { get; set; }

[Display(Name = "HeadLine")]

[StringLength(50, ErrorMessage = "Headline darf nicht länger als 50 Zeichen sein")]

[Required(ErrorMessage = "Geben Sie bitte eine Benenung der Headline an")]

public string Title { get; set; }

[Display(Name = "Inhalt\_der\_Rubrik")]

[StringLength(3000, ErrorMessage = "Höchslänge 3000 Zeichen wurde überschritten")]

[DataType(DataType.MultilineText)]

public string Text { get; set; }

[Display(Name ="Datum\_dieser\_Rubrik")]

[DataType(DataType.Date)]

[Required(ErrorMessage ="Wählen Sie bitte einen Datum aus dem Kalendar aus")]

public DateTime DateOfTeaser { get; set; }

public TeaserEntity(int iD, string title, string text, DateTime dateOfTeaser)

{

this.Id = iD;

this.Title = title;

this.Text = text;

this.DateOfTeaser = dateOfTeaser;

}

public TeaserEntity()

{ }

}

}

MODEL:FooterEntity

using System.ComponentModel.DataAnnotations;

namespace Web.AL.Models

{

public class FooterEntity

{

[Display(Name = "HeadLine\_1")]

[StringLength(50, ErrorMessage = "Headline darf nicht länger als 50 Zeichen sein")]

[Required(ErrorMessage = "Geben Sie bitte eine Benenung der Headline an")]

public string Title1 { get; set; }

[Display(Name = "HeadLine\_2")]

[StringLength(50, ErrorMessage = "Headline darf nicht länger als 50 Zeichen sein")]

[Required(ErrorMessage = "Geben Sie bitte eine Benenung der Headline an")]

public string Title2 { get; set; }

[Display(Name = "Wortlaut\_zum\_Link\_1")]

[StringLength(50, ErrorMessage = "Worlaut darf nicht länger als 50 Zeichen sein")]

public string Text1 { get; set; }

[Display(Name = "Link 1")]

public LinkFooter Link1 { get; set; } //celkem 12 linků, vytištěn jen jeden

}

public class LinkFooter

{

[Display(Name = "Link\_Text")]

[StringLength(50, ErrorMessage = "Wortlaut des TextsLinkes darf nicht länger als 50 Zeichen sein")]

public string LinkText { get; set; }

[Display(Name = "AdresseLink")]

[StringLength(200, ErrorMessage = "Wortlaut des AdresseLinkes darf nicht länger als 200 Zeichen sein")]

public string LinkAddress { get; set; }

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Zobrazení stránky web visitorovi***

CONTROLLER:HomeController

using System.Web.Mvc;

using Web.AL.Models;

namespace Web.AL.Controllers

{

public class HomeController : Controller

{

PageViewModel page = new PageViewModel();

public ActionResult Index()

{

return View(page);

}

public ActionResult DisplayPageWithOneTeaser(int teaserId)

{

page.SetPageWithOneTeaser(teaserId);

return View(page);

}

}

}

MODEL:PageViewModel

using System.Collections.Generic;

namespace Web.AL.Models

{

public class PageViewModel

{

HeaderViewModel header;

TeasersViewModel teasers;

FooterViewModel footer;

public PageViewModel()

{

header = new HeaderViewModel();

teasers = new TeasersViewModel();

footer = new FooterViewModel();

}

public HeaderEntity Header

{

get

{

return header.Header;

}

}

public List<TeaserEntity> Teasers

{

get

{

return teasers.Teasers;

}

}

public FooterEntity Footer

{

get

{

return footer.Footer;

}

}

public TeaserEntity Teaser

{

get

{

return teasers.Teaser;

}

}

public void SetPageWithOneTeaser(int teaserId)

{

teasers.GetTeaser(teaserId);

footer = new FooterViewModel();

}

}

}

MODEL:HeaderViewModel

using Web.DL;

namespace Web.AL.Models

{

public class HeaderViewModel

{

DbRepository dbRepository = new DbRepository();

public HeaderEntity Header { get; set; }

public HeaderViewModel()

{

HeaderEntityDL headerFromDB = dbRepository.GetHeaderData();

Header = new HeaderEntity();Header.Link1 = new LinkHeader();Header.Link2 = new LinkHeader();

Header.Title = headerFromDB.Title ?? "Die angaben wurden nicht gelesen.";

Header.Text = headerFromDB.Text ?? string.Empty;

Header.Link1.LinkText = headerFromDB.Link1.LinkText ?? string.Empty;

Header.Link1.LinkAddress = headerFromDB.Link1.LinkAddress ?? string.Empty;

Header.Link2.LinkText = headerFromDB.Link2.LinkText ?? string.Empty;

Header.Link2.LinkAddress = headerFromDB.Link2.LinkAddress ?? string.Empty;

}

}

}

MODEL:AdminTeaserViewModel

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using Web.DL;

namespace Web.AL.Models

{

public class AdminTeaserViewModel

{

DbRepository dbRepository = new DbRepository();

[Display(Name = "Identifizierungsnummer")]

[Required(ErrorMessage = "Geben Sie bitte eine Identifizierungsnummer der Rubrik")]

[RegularExpression("([1-3])", ErrorMessage ="Es gibt nur drei Rubriken, deswegen geben Sie bitte eins, zwei oder drei an")]

public int? IdTeaser { get; set; }

public List<TeaserEntity> Teasers { get; set; }

public TeaserEntity TeaserOld { get; set; }

public TeaserEntity TeaserNew { get; set; }

public AdminTeaserViewModel()

{

List<TeaserEntityDL> teasersFromDB = dbRepository.GetAllTeasers();

Teasers = new List<TeaserEntity>();

Teasers = teasersFromDB.Select(x => new TeaserEntity(x.Id, x.Title, x.Text, x.DateOfTeaser)).ToList();

}

private int GetIdTeaser()

{

if (!IdTeaser.HasValue)

throw new InvalidOperationException($"{nameof(IdTeaser)} is empty");

return IdTeaser.Value;

}

public void FindTeaser(AdminTeaserViewModel adminTeaserViewModel)

{

TeaserEntityDL teaserFromDB = dbRepository.GetTeaserData(GetIdTeaser());

if (teaserFromDB == null)

TeaserOld = new TeaserEntity(0, "Die angaben wurden nicht gelesen", "", DateTime.Today);

TeaserOld = new TeaserEntity(teaserFromDB.Id, teaserFromDB.Title, teaserFromDB.Text, teaserFromDB.DateOfTeaser);

TeaserNew = TeaserOld;

}

public void UpdateTeaser(AdminTeaserViewModel adminTeaserViewModel)

{

TeaserEntityDL teaserToDB = new TeaserEntityDL();

teaserToDB.Id = adminTeaserViewModel.TeaserOld.Id;

teaserToDB.Title = adminTeaserViewModel.TeaserNew.Title;

teaserToDB.Text = adminTeaserViewModel.TeaserNew.Text ?? string.Empty;

teaserToDB.DateOfTeaser = adminTeaserViewModel.TeaserNew.DateOfTeaser;

dbRepository.UpdateTeaserData(teaserToDB);

}

}

}

MODEL:TeasersViewModel

using System.Collections.Generic;

using System.Linq;

using Web.DL;

namespace Web.AL.Models

{

public class TeasersViewModel

{

DbRepository dbRepository = new DbRepository();

private const string pathToDisplayImages = "/Content/Images/Teasers/";

public List<TeaserEntity> Teasers { get; set; }

public TeaserEntity Teaser { get; set; }

public TeasersViewModel()

{

List<TeaserEntityDL> teasersFromDB = dbRepository.GetAllTeasers();

Teasers = new List<TeaserEntity>();

Teasers = teasersFromDB.Select(x => new TeaserEntity(x.Id, x.Title, x.Text, x.DateOfTeaser, pathToDisplayImages + x.ImageName + ".jpg")).ToList();

}

public void GetTeaser(int teaserId)

{

TeaserEntityDL teaserFromDB = dbRepository.GetTeaserData(teaserId);

Teaser = new TeaserEntity(teaserFromDB.Id,teaserFromDB.Title,teaserFromDB.Text,teaserFromDB.DateOfTeaser,pathToDisplayImages + teaserFromDB.ImageName + ".jpg");

}

}

}

MODEL:FooterViewModel

using Web.DL;

namespace Web.AL.Models

{

public class FooterViewModel

{

DbRepository dbRepository = new DbRepository();

public FooterEntity Footer { get; set; }

public FooterViewModel()

{

FooterEntityDL footerFromDB = dbRepository.GetFooterData();

Footer = new FooterEntity(); Footer.Link1 = new LinkFooter(); Footer.Link2 = new LinkFooter(); Footer.Link3 = new LinkFooter(); Footer.Link4 = new LinkFooter(); Footer.Link5 = new LinkFooter(); Footer.Link6 = new LinkFooter(); Footer.Link7 = new LinkFooter(); Footer.Link8 = new LinkFooter(); Footer.Link9 = new LinkFooter(); Footer.Link10 = new LinkFooter(); Footer.Link11 = new LinkFooter(); Footer.Link12 = new LinkFooter();

Footer.Title1 = footerFromDB.Title1 ?? "Die angaben wurden nicht gelesen.";

Footer.Title2 = footerFromDB.Title2 ?? "Die angaben wurden nicht gelesen.";

Footer.Text1 = footerFromDB.Text1;

Footer.Link1.LinkText = footerFromDB.Link1.LinkText;

Footer.Link1.LinkAddress = footerFromDB.Link1.LinkAddress; //+11 dalších linků

}

}

}

VIEW:Home:Index

@model Web.AL.Models.PageViewModel

@{

ViewBag.Title = "Home Page";

}

@using (Html.BeginForm())

{

Html.AntiForgeryToken();

Html.ValidationSummary();

<div class="jumbotron">

<h1> @Model.Header.Title</h1>

<p class="lead">@Model.Header.Text</p>

<p>

<a href="@Model.Header.Link1.LinkAddress" class="btn btn-primary btn-lg"> @Model.Header.Link1.LinkText</a>

<a href="@Model.Header.Link2.LinkAddress" class="btn btn-primary btn-lg"> @Model.Header.Link2.LinkText</a>

</p>

</div>

}

@foreach (var teaser in Model.Teasers)

{

<div class="form-horizontal">

<div class="col-md-4">

<h2>@teaser.Title</h2>

<p>

@teaser.Text.Remove(50)

</p>

<p>

<h4> @Html.ActionLink("lesen weiter", "DisplayPageWithOneTeaser", "Home", new { teaserId=teaser.Id}, new { @class = "btn btn-default" })</h4><br />

</p>

<p><img src="@teaser.ImageName" width="150"/> </p>

<p>@Html.TextBoxFor(model => teaser.DateOfTeaser, @"{0:dd/MM/yyyy}", new { @class = "form-control", @readonly="readonly" })</p>

</div>

</div>

}

<div class="col-md-6" style="text-align:center">

<h4>@Model.Footer.Title1</h4>

<div class="row">

@Model.Footer.Text1

<a href="@Model.Footer.Link1.LinkAddress">@Model.Footer.Link1.LinkText</a><br/>

// a dalších 11 linků

</div>

</div>

VIEW:Home:DisplayPageWithOneTeaser

@model Web.AL.Models.PageViewModel

@{

ViewBag.Title = "DisplayPageWithOneTeaser";

}

<div class="jumbotron">

<h2>@Model.Teaser.Title</h2>

<p>@Model.Teaser.Text</p>

<p>@Model.Teaser.DateOfTeaser.ToShortDateString()</p>

<p> <img src="@Model.Teaser.ImageName" width="300" /> </p><br/>

<h4> @Html.ActionLink("Zurück zur Startseite", "Index", "Home", null, new { @class = "btn btn-default" })</h4><br />

</div>

<div class="col-md-6" style="text-align:center">

<h4>@Model.Footer.Title1</h4>

<div class="row">

@Model.Footer.Text1

<a href="@Model.Footer.Link1.LinkAddress">@Model.Footer.Link1.LinkText</a><br/>

// a dalších 11 linků

</div>

</div>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Nabídka funkcionalit Adminovi***

CONTROLLER:AdminOptionController

using System.Web.Mvc;

namespace Web.AL.Controllers

{

public class AdminOptionController : Controller

{

[Authorize]

public ActionResult Option()

{

return View();

}

}

}

VIEW:AdminOption:Option

@{

ViewBag.Title = "AdminOption";

}

<h2>Administration der web Seiten</h2>

<h4> @Html.ActionLink("Änderung der Kopfseite", "UpdateHeader", "AdminHeader", null, new { @class = "btn btn-default" })</h4><br/>

<h4> @Html.ActionLink("Änderung der Fußseite", "UpdateFooter", "AdminFooter", null, new { @class= "btn btn-default" })</h4><br/><br/>

<h4> @Html.ActionLink("Abbildung aller Rubriken", "Display", "AdminTeaser", null, new { @class = "btn btn-default" })</h4><br />

<h4> @Html.ActionLink("Änderung einer Rubrik aufgrund ID", "Update", "AdminTeaser", null, new { @class = "btn btn-default" })</h4><br />

<h4> @Html.ActionLink("Löschung eines Bildes", "DisplayImages", "AdminImage", null, new { @class = "btn btn-default" })</h4>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Změna hlavičky Adminem***

CONTROLLER:AdminHeaderController

using System.Web.Mvc;

using Web.AL.Models;

namespace Web.AL.Controllers

{

public class AdminHeaderController : Controller

{

[Authorize]

public ActionResult UpdateHeader()

{

AdminHeaderViewModel adminHeaderViewModel = new AdminHeaderViewModel();

return View(adminHeaderViewModel);

}

[HttpPost]

[Authorize]

public ActionResult UpdateHeader(AdminHeaderViewModel adminHeaderViewModel)

{

if (!ModelState.IsValid)

{

return View(adminHeaderViewModel);

}

adminHeaderViewModel.Update(adminHeaderViewModel);

TempData["AdminResult"] = "Die Anforderung wurde erfüllt";

return RedirectToAction("Option", "AdminOption");

}

}

}

MODEL:AdminHeaderViewModel

using Web.DL;

namespace Web.AL.Models

{

public class AdminHeaderViewModel

{

DbRepository dbRepository = new DbRepository();

public HeaderEntity HeaderOld { get; set; }

public HeaderEntity HeaderNew { get; set; }

public AdminHeaderViewModel()

{

HeaderEntityDL headerFromDB = dbRepository.GetHeaderData();

HeaderOld = new HeaderEntity(); HeaderOld.Link1 = new LinkHeader(); HeaderOld.Link2 = new LinkHeader();

HeaderNew = new HeaderEntity(); HeaderNew.Link1 = new LinkHeader(); HeaderNew.Link2 = new LinkHeader();

HeaderOld.Title = headerFromDB.Title ?? "Die angaben wurden nicht gelesen.";

HeaderOld.Text = headerFromDB.Text;

HeaderOld.Link1.LinkText = headerFromDB.Link1.LinkText;

HeaderOld.Link1.LinkAddress = headerFromDB.Link1.LinkAddress;

HeaderOld.Link2.LinkText = headerFromDB.Link2.LinkText;

HeaderOld.Link2.LinkAddress = headerFromDB.Link2.LinkAddress;

HeaderNew = HeaderOld;

}

public void Update (AdminHeaderViewModel adminHeaderViewModel)

{

HeaderEntityDL headerToDB = new HeaderEntityDL();

headerToDB.Link1 = new LinkDL(); headerToDB.Link2 = new LinkDL();

headerToDB.Id = 1;

headerToDB.Title = adminHeaderViewModel.HeaderNew.Title;

headerToDB.Text = adminHeaderViewModel.HeaderNew.Text ?? string.Empty;

headerToDB.Link1.LinkText = adminHeaderViewModel.HeaderNew.Link1.LinkText ?? string.Empty;

headerToDB.Link1.LinkAddress = adminHeaderViewModel.HeaderNew.Link1.LinkAddress ?? string.Empty;

headerToDB.Link2.LinkText = adminHeaderViewModel.HeaderNew.Link2.LinkText ?? string.Empty;

headerToDB.Link2.LinkAddress = adminHeaderViewModel.HeaderNew.Link2.LinkAddress ?? string.Empty;

dbRepository.UpDateHeader(headerToDB);

}

}

}

VIEW:AdminHeader:UpdateHeader

@model Web.AL.Models.AdminHeaderViewModel

@{

ViewBag.Title = "UpdateHomePage";

}

<h2>Änderung der Heimatseite</h2>

@using (Html.BeginForm())

{

Html.AntiForgeryToken();

Html.ValidationSummary(true);

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Title, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Title, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Text, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Text, new { @class = "form-control", @readonly = "readonly" })

</div>

</div><br/>

@Html.LabelFor(model => model.HeaderOld.Link1, new { @class= "control-label col-md-2" })

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Link1.LinkText, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Link1.LinkText, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Link1.LinkAddress, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Link1.LinkAddress, new { @class = "form-control", @readonly = "readonly" })

</div>

</div><br/><br/>

@Html.LabelFor(model => model.HeaderOld.Link2,new { @class= "control-label col-md-2" })

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Link2.LinkText, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Link2.LinkText, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.HeaderOld.Link2.LinkAddress, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.HeaderOld.Link2.LinkAddress, new { @class = "form-control", @readonly = "readonly" })

</div>

</div><br/><br/>

<p><strong>Geben Sie bitte neue Texte an oder lassen Sie die ursprüngliche Fassung</strong></p>

<div class="form-horizontal">

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Title, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Title, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.HeaderNew.Title, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Text, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Text, new { @class = "form-control"})

@Html.ValidationMessageFor(model => model.HeaderNew.Text, "", new { @class = "text-danger" })

</div>

</div><br/>

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Link1, new { @class = "form-control" })<br/>

@Html.LabelFor(model => model.HeaderNew.Link1.LinkText, new { @class = "control-label col-md-2", })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Link1.LinkText, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.HeaderNew.Link1.LinkText, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Link1.LinkAddress, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Link1.LinkAddress, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.HeaderNew.Link1.LinkAddress, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Link2, new { @class = "form-control" })<br/>

@Html.LabelFor(model => model.HeaderNew.Link2.LinkText, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Link2.LinkText, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.HeaderNew.Link2.LinkText, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.HeaderNew.Link2.LinkAddress, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.HeaderNew.Link2.LinkAddress, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.HeaderNew.Link2.LinkAddress, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="ändern" class="btn btn-default" />

</div>

</div>

</div>

@Html.ActionLink("Zurück zur Administration","Option","AdminOption")

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Změna patičky Adminem***

CONTROLLER:AdminFooterController

using System.Web.Mvc;

using Web.AL.Models;

namespace Web.AL.Controllers

{

public class AdminFooterController : Controller

{

[Authorize]

public ActionResult UpdateFooter()

{

AdminFooterViewModel adminFooterViewModel = new AdminFooterViewModel();

return View(adminFooterViewModel);

}

[Authorize]

[HttpPost]

public ActionResult UpdateFooter(AdminFooterViewModel adminFooterViewModel)

{

if (!ModelState.IsValid)

{

return View(adminFooterViewModel);

}

adminFooterViewModel.Update(adminFooterViewModel);

TempData["AdminResult"] = "Die Anforderung wurde erfüllt";

return RedirectToAction("Option", "AdminOption");

}

}

}

MODEL:AdminFooterViewModel

using Web.DL;

namespace Web.AL.Models

{

public class AdminFooterViewModel

{

DbRepository dbRepository = new DbRepository();

public FooterEntity FooterOld { get; set; }

public FooterEntity FooterNew { get; set; }

public AdminFooterViewModel()

{

FooterOld = new FooterEntity();FooterOld.Link1 = new LinkFooter(); + 11 dalších linků

FooterNew = new FooterEntity(); FooterNew.Link1 = new LinkFooter();+ 11 dalších linků

FooterEntityDL footerFromDB = dbRepository.GetFooterData();

FooterOld.Title1 = footerFromDB.Title1 ?? "Die angaben wurden nicht gelesen.";

FooterOld.Title2 = footerFromDB.Title1 ?? "Die angaben wurden nicht gelesen.";

FooterOld.Text1 = footerFromDB.Text1;

FooterOld.Link1.LinkText = footerFromDB.Link1.LinkText;

FooterOld.Link1.LinkAddress = footerFromDB.Link1.LinkAddress; // +11 dalších Linků

FooterNew = FooterOld;

}

public void Update(AdminFooterViewModel adminFooterViewModel)

{

FooterEntityDL footerToDB = new FooterEntityDL();

footerToDB.Id = 1;

footerToDB.Link1 = new LinkDLFooter();+ 11 dalších linků

footerToDB.Title1 = adminFooterViewModel.FooterNew.Title1;

footerToDB.Title2 = adminFooterViewModel.FooterNew.Title2;

footerToDB.Text1 = adminFooterViewModel.FooterNew.Text1 ?? string.Empty;

footerToDB.Link1.LinkText = adminFooterViewModel.FooterNew.Link1.LinkText ?? string.Empty;

footerToDB.Link1.LinkAddress = adminFooterViewModel.FooterNew.Link1.LinkAddress ?? string.Empty; // + 11 dalších Linků

dbRepository.UpDateFooter(footerToDB);

}

}

}

VIEW:AdminFooter:UpdateFooters

@model Web.AL.Models.AdminFooterViewModel

@{

ViewBag.Title = "UpdateFooter";

}

<h2>Änderung der Fußseite</h2>

@using (Html.BeginForm())

{

Html.AntiForgeryToken();

Html.ValidationSummary(true);

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.FooterOld.Title1, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.FooterOld.Title1, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.FooterOld.Title2, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.FooterOld.Title2, new { @class = "form-control", @readonly = "readonly" })

</div>

</div><br />

@Html.LabelFor(model => model.FooterOld.Link1, new { @class = "control-label col-md-2" })

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.FooterOld.Text1, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.FooterOld.Text1, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.FooterOld.Link1.LinkText, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.FooterOld.Link1.LinkText, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.FooterOld.Link1.LinkAddress, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.FooterOld.Link1.LinkAddress, new { @class = "form-control", @readonly = "readonly" })

</div>

</div> // + 11 dalších FooterOld linků

<h4><strong>Geben Sie bitte neue Texte an oder lassen Sie die ursprüngliche Fassung</strong></h4>

<div class="form-horizontal">

<div class="form-group">

@Html.LabelFor(model => model.FooterNew.Title1, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.FooterNew.Title1, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.FooterNew.Title1, "", new { @class = "text-danger" })

</div>

</div> // +11 dalších FooterNewLinků

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="ändern" class="btn btn-default" />

</div>

</div>

</div>

@Html.ActionLink("Zurück zur Administration", "Option", "AdminOption")

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Zobrazení všech teasers a úprava jednoho z nich Adminem***

CONTROLLER:AdminTeaserController

using System.Web.Mvc;

using Web.AL.Models;

using System;

namespace Web.AL.Controllers

{

public class AdminTeaserController : Controller

{

AdminTeaserViewModel adminTeaserViewModel = new AdminTeaserViewModel();

[Authorize]

public ActionResult Display()

{

return View(adminTeaserViewModel);

}

[Authorize]

public ActionResult Update()

{

return View(adminTeaserViewModel);

}

[Authorize]

[HttpPost]

public ActionResult Update(AdminTeaserViewModel adminTeaserViewModel)

{

if (!ModelState.IsValid)

{

return View(adminTeaserViewModel);

}

adminTeaserViewModel.FindTeaser(adminTeaserViewModel);

return View("TeaserForUpdate", adminTeaserViewModel);

}

[Authorize]

[HttpPost]

public ActionResult UpdateTeaser(AdminTeaserViewModel adminTeaserViewModel)

{

if (adminTeaserViewModel.TeaserNew.Title == null || adminTeaserViewModel.TeaserNew.DateOfTeaser == DateTime.MinValue ||

adminTeaserViewModel.TeaserNew.Title.Length > 50 || adminTeaserViewModel.TeaserNew.Text.Length > 3000

)

{

return View("TeaserForUpdate",adminTeaserViewModel);

}

adminTeaserViewModel.UpdateTeaser(adminTeaserViewModel);

TempData["AdminResult"] = "Die Anforderung wurde erfüllt";

return RedirectToAction("Option", "AdminOption");

}

}

}

MODEL:AdminTeasersViewModel

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using Web.DL;

namespace Web.AL.Models

{

public class AdminTeasersViewModel

{

DbRepository dbRepository = new DbRepository();

private const string pathToDisplayImages = "/Content/Images/Teasers/";

private const string pathToUpdateImages = "C:/Repos/training-david/Web.AL/Web.AL/Content/Images/Teasers/";

[Display(Name = "Identifizierungsnummer")]

[Required(ErrorMessage = "Geben Sie bitte eine Identifizierungsnummer der Rubrik")]

[RegularExpression("([1-3])", ErrorMessage ="Es gibt nur drei Rubriken, deswegen geben Sie bitte eins, zwei oder drei an")]

public int? IdTeaser { get; set; }

public List<TeaserEntity> Teasers { get; set; }

public TeaserEntity TeaserOld { get; set; }

public TeaserEntity TeaserNew { get; set; }

public AdminTeasersViewModel()

{

List<TeaserEntityDL> teasersFromDB = dbRepository.GetAllTeasers();

Teasers = new List<TeaserEntity>();

Teasers = teasersFromDB.Select(x => new TeaserEntity(x.Id, x.Title, x.Text, x.DateOfTeaser,pathToDisplayImages + x.ImageName + ".jpg")).ToList();

}

private int GetIdTeaser()

{

if (!IdTeaser.HasValue)

throw new InvalidOperationException($"{nameof(IdTeaser)} is empty");

return IdTeaser.Value;

}

public void FindTeaser(AdminTeasersViewModel adminTeaserViewModel)

{

TeaserEntityDL teaserFromDB = dbRepository.GetTeaserData(GetIdTeaser());

if (teaserFromDB == null)

TeaserOld = new TeaserEntity(0, "Die angaben wurden nicht gelesen", "", DateTime.Today,string.Empty);

TeaserOld = new TeaserEntity(teaserFromDB.Id, teaserFromDB.Title, teaserFromDB.Text, teaserFromDB.DateOfTeaser, pathToDisplayImages + teaserFromDB.ImageName + ".jpg");

TeaserNew = TeaserOld;

}

private string SaveImage(AdminTeasersViewModel adminTeaserViewModel)

{

try

{

adminTeaserViewModel.TeaserNew.Image.SaveAs(pathToUpdateImages + adminTeaserViewModel.TeaserOld.Id + ".jpg");

return string.Empty;

}

catch (Exception ex)

{

return $"Das Bild zum Artikel Id: {adminTeaserViewModel.TeaserNew.Id} wurde nicht eingespeichert. Unten Fehlerbeschreibung: \n{ex.Message.ToString()}";

}

}

public string UpdateTeaser(AdminTeasersViewModel adminTeaserViewModel)

{

TeaserEntityDL teaserToDB = new TeaserEntityDL();

if (adminTeaserViewModel.TeaserNew.Image == null)

{

teaserToDB.Id = adminTeaserViewModel.TeaserOld.Id;

teaserToDB.Title = adminTeaserViewModel.TeaserNew.Title;

teaserToDB.Text = adminTeaserViewModel.TeaserNew.Text ?? string.Empty;

teaserToDB.DateOfTeaser = adminTeaserViewModel.TeaserNew.DateOfTeaser;

teaserToDB.ImageName = null;

return dbRepository.UpdateTeaserData(teaserToDB);

}

string ImageSaveError = SaveImage(adminTeaserViewModel);

if (ImageSaveError != string.Empty)

return ImageSaveError;

teaserToDB.Id = adminTeaserViewModel.TeaserOld.Id;

teaserToDB.Title = adminTeaserViewModel.TeaserNew.Title;

teaserToDB.Text = adminTeaserViewModel.TeaserNew.Text ?? string.Empty;

teaserToDB.DateOfTeaser = adminTeaserViewModel.TeaserNew.DateOfTeaser;

teaserToDB.ImageName = adminTeaserViewModel.TeaserOld.Id.ToString();

return dbRepository.UpdateTeaserData(teaserToDB);

}

}

}

VIEW:AdminTeaser:Display

@model Web.AL.Models.AdminTeasersViewModel

@{

ViewBag.Title = "AdminDisplayTeasers";

}

<h2>Abbildung aller Rubriken</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

foreach (var teaser in Model.Teasers)

{

<div class="row">

<div class="col-md-2">

@Html.LabelFor(m => teaser.Id, new { @class = "control-label" })

@Html.TextBoxFor(m => teaser.Id, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(m => teaser.Title, new { @class = "control-label" })

@Html.TextBoxFor(m => teaser.Title, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(m => teaser.Text, new { @class = "control-label" })

@Html.TextBoxFor(m => teaser.Text, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(m => teaser.DateOfTeaser, new { @class = "control-label" })

@Html.TextBoxFor(m => teaser.DateOfTeaser,"{0:dd/MM/yyyy}" ,new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

<img src="@teaser.ImageName" width="150" />

</div>

</div>

@Html.ActionLink("Änderung dieser Rubrik", "UpDateFromDisplay", "AdminTeasers", new { teaserId=teaser.Id }, new { @class = "btn btn-default" })

<br /><br/>

}

}

@Html.ActionLink("Zurück zur Administration", "Option", "AdminOption", null, new { @class = "btn btn-default" })

VIEW:AdminTeaser:Update

@model Web.AL.Models.AdminTeaserViewModel

@{

ViewBag.Title = "AdminTeaserUpdate";

}

<h2>Änderung einer Rubrik aufgrund ID</h2><br />

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

<div class="form-horizontal">

<div class="form-group">

@Html.LabelFor(model => model.IdTeaser, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.TextBoxFor(model => model.IdTeaser, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.IdTeaser, "", new { @class = "text-danger" })

</div>

</div><br />

<div class="form-group">

<div class="col-md-offset-2 col-md-10" ¨>

<input type="submit" value="finden" class="btn btn-default" />

</div>

</div>

</div>

@Html.ActionLink("Zurück zur Administration", "Option", "AdminOption")

}

VIEW:AdminTeaser:TeaserForUpdate

@model Web.AL.Models.AdminTeasersViewModel

@{

ViewBag.Title = "AdminTeaserUpdate";

}

<h2>Abbildung einer Rubrik im Zwecke ihrer Änderung</h2><br />

@using (Html.BeginForm("UpdateTeaser", "AdminTeasers",FormMethod.Post, new { enctype="multipart/form-data"}))

{

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

<div class="row">

<div class="col-md-2">

@Html.LabelFor(model => model.TeaserOld.Id , new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.TeaserOld.Id, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.TeaserOld.Title, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.TeaserOld.Title, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.TeaserOld.Text, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.TeaserOld.Text, new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

@Html.LabelFor(model => model.TeaserOld.DateOfTeaser, new { @class = "control-label col-md-2" })

@Html.TextBoxFor(model => model.TeaserOld.DateOfTeaser,@"{0:dd/MM/yyyy}", new { @class = "form-control", @readonly = "readonly" })

</div>

<div class="col-md-2">

<img src="@Model.TeaserOld.ImageName" width="150" />

</div>

</div><br />

<p><h4><strong>Geben Sie bitte neue Angaben an oder lassen Sie die ursprüngliche Fassung</strong></h4></p><br/>

<div class="form-horizontal">

<div class="form-group">

@Html.LabelFor(model => model.TeaserNew.Title, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.TeaserNew.Title, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.TeaserNew.Title, "", new { @class = "text-danger" })

</div>

</div><br />

<div class="form-group">

@Html.LabelFor(model => model.TeaserNew.Text, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.TeaserNew.Text, new { @class = "form-control" })

</div>

</div>

<br />

<div class="form-group">

@Html.LabelFor(model => model.TeaserNew.DateOfTeaser, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.TeaserNew.DateOfTeaser, new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.TeaserNew.DateOfTeaser, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.TeaserNew.Image, new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.TextBoxFor(model => model.TeaserNew.Image, new { @type="file", @accept=".jpg", @class = "form-control" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="ändern" class="btn btn-default" />

</div>

</div>

</div>

@Html.ActionLink("Zurück zur Administration", "Option", "AdminOption")

}

---------------------WEB data layer---------------------------------------------------------------

***Představení použitých entit***

MODEL:HeaderEntityDL

namespace Web.DL

{

public class HeaderEntityDL

{

public int Id { get; set; }

public string Title { get; set; }

public string Text { get; set; }

public LinkDL Link1 { get; set; }

public LinkDL Link2 { get; set; }

}

public class LinkDL

{

public string LinkText { get; set; }

public string LinkAddress { get; set; }

}

}

MODEL:TeaserEntityDL

using System;

namespace Web.DL

{

public class TeaserEntityDL

{

public int Id { get; set; }

public string Title { get; set; }

public string Text { get; set; }

public DateTime DateOfTeaser { get; set; }

public string ImageName { get; set; }

}

}

MODEL:FooterEntityDL

namespace Web.DL

{

public class FooterEntityDL

{

public int Id { get; set; }

public string Title1 { get; set; }

public string Title2 { get; set; }

public string Text1 { get; set; }

public LinkDLFooter Link1 { get; set; } // + 11 dalších linků

}

public class LinkDLFooter

{

public string LinkText { get; set; }

public string LinkAddress { get; set; }

}

}

MODEL:EntityFramework

using System.Data.Entity;

namespace Web.DL

{

public class EntityFramework:DbContext

{

public DbSet<HeaderEntityDL> HeaderEntityDLTable { get; set; }

public DbSet<FooterEntityDL> FooterEntityDLTable { get; set; }

public DbSet<TeaserEntityDL> TeaserEntityDLTable { get; set; }

public EntityFramework():base("DefaultConnection")

{ }

}

}

MODEL:DbRepository

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

namespace Web.DL

{

public class DbRepository

{

private const string ConnectionString = @"Data Source=.\sqlexpress;Initial Catalog=WebDB;Integrated Security=True";

// HEADER

public HeaderEntityDL GetHeaderData()

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

return accessToDB.HeaderEntityDLTable.SingleOrDefault();

}

}

catch (Exception ex)

{

HeaderEntityDL error = new HeaderEntityDL();

error.Title = "Die KopfSeite wurde nicht eingelesen";

error.Text = "Unten Fehlerbeschreibung: \n";

error.Link1.LinkText = ex.Message.ToString();

return error;

}

public string UpDateHeader(HeaderEntityDL headerToDB)

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

accessToDB.Entry(headerToDB).State = EntityState.Modified;

accessToDB.SaveChanges();

return "Die Anforderung wurde erfüllt";

}

}

catch (Exception ex)

{

string error = $"Die Anforderung wurde nicht erfüllt. Unten Fehlerbeschreibung: \n{ex.Message.ToString()}";

return error;

}

// Footer

public FooterEntityDL GetFooterData()

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

return accessToDB.FooterEntityDLTable.SingleOrDefault();

}

}

catch (Exception ex)

{

FooterEntityDL error = new FooterEntityDL();

error.Title1 = "Die Fußseite wurde nicht eingelesen";

error.Title2 = "Unten Fehlerbeschreibung: \n";

error.Text1 = ex.Message.ToString();

return error;

}

public string UpDateFooter(FooterEntityDL footerToDB)

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

accessToDB.Entry(footerToDB).State = EntityState.Modified;

accessToDB.SaveChanges();

return "Die Anforderung wurde erfüllt";

}

}

catch (Exception ex)

{

string error = $"Die Anforderung wurde nicht erfüllt. Unten Fehlerbeschreibung: \n:{ex.Message.ToString()}";

return error;

}

// Teasers

public List<TeaserEntityDL> GetAllTeasers()

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

return accessToDB.TeaserEntityDLTable.ToList();

}

}

catch (Exception ex)

{

List<TeaserEntityDL> error = new List<TeaserEntityDL>();

TeaserEntityDL error1 = new TeaserEntityDL();

error1.ImageName = "Die Artikel wurden nicht eingelesen";

error1.Text = "Unten Fehlerbeschreibung: \n";

error1.ImageName = ex.Message.ToString();

error.Add(error1);

return error;

}

}

public TeaserEntityDL GetTeaserData (int iDteaser)

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

return accessToDB.TeaserEntityDLTable.Where(x => x.Id == iDteaser).SingleOrDefault();

}

}

catch (Exception ex)

{

TeaserEntityDL error = new TeaserEntityDL();

error.Title = $"Der Artikel Id={iDteaser} wurde nicht eingelesen";

error.Text = "Fehlerbeschreibung:";

error.ImageName = $"{ex.Message.ToString()}";

return error;

}

public string UpdateTeaserData(TeaserEntityDL teaserToDB)

{

try

{

using (EntityFramework accessToDB = new EntityFramework())

{

accessToDB.Entry(teaserToDB).State = EntityState.Modified;

accessToDB.SaveChanges();

return "Die Anforderung wurde erfüllt";

}

}

catch (Exception ex)

{

string error= $"Die Anforderung wurde nicht erfüllt. Unten Fehlerbeschreibung:\n {ex.Message.ToString()}";

return error;

}