WEEK 4

Sayfalama İşlemi

Controller kısmı

public class MembersController : Controller

{

DBLibraryEntities entities = new DBLibraryEntities();

public ActionResult Index(int sayfa=1,string p="")

{

var query = entities.Member.AsQueryable();

if (!string.IsNullOrEmpty(p))

{

query = query.Where(x => x.MemberName.Contains(p) || x.Username.Contains(p) || x.MemberSurname.Contains(p));

}

var values =query.OrderBy(x=>x.MemberId).ToPagedList(sayfa, 5);

return View(values);

}

Bu kod sayesinde hem sayfalama hemde arama kısmı oluyor view kısımları ise

@using MvcLibrary.Models.Entity

@using PagedList

@using PagedList.Mvc

@model PagedList.IPagedList<Member>

Bunlar kütüphaneler

@using (Html.BeginForm("Index", "Members", FormMethod.Post))

{

<div style="margin-bottom:15px; margin-top:15px">

<p>Aranacak Üyenin Adını giriniz : </p>@Html.TextBox("p")

<input type="submit" value="ara" />

</div>

}

Bu aramanın kodları

@Html.PagedListPager((IPagedList)Model, sayfa => Url.Action("Index", new { sayfa }),new PagedListRenderOptions

{

LiElementClasses=new List<string>

{

{"page-link" }

}

})

Buda sayfalamanın kodları

AUTHORİZE İŞLEMLERİ

İlk işimiz web config system webin sonuna bunu yaz

<authentication mode="Forms">

<forms loginUrl="/Login/Login/"></forms>

</authentication>

Bunun amacı sisteme giriş yapmadığın sürece forms aktif olunca buna yönlendir ardından

Controllere dönüp

public class PanelController : Controller

{

// GET: Panel

[Authorize]

public ActionResult Index()

{

return View();

}

}

Nerede authorize işlemi olacak yani yapılmadan bu sayfaya gidemez bu sayfaya gitmesi için authorize olması lazım

Giriş İşlemleri ise aşağıdaki gibidir

DBLibraryEntities entities = new DBLibraryEntities();

// GET: Login

[HttpGet]

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Member member)

{

var values = entities.Member.FirstOrDefault(x => x.MemberMail == member.MemberMail && x.Password == member.Password); // bilgiler doğrumu tut

if (values != null)

{

FormsAuthentication.SetAuthCookie(values.MemberMail, false); //form da giriş yaparsak buradan maili tut cookide kullanıcı adı yerine

return RedirectToAction("Index", "Panel");

}

else

{

return View();

}

Session işlemleri yani profili getirme işlemleri

Sisteme giriş yapan kişinin adını ve ya mailini getirme yöntemi

@HttpContext.Current.User.Identity.Name

FormsAuthentication.SetAuthCookie(values.MemberMail, false); //form da giriş yaparsak buradan maili tut cookide kullanıcı adı yerine

Session["Mail"] = values.MemberMail.ToString();

TempData["ID"] = values.MemberId.ToString();

TempData["Ad"] = values.MemberName.ToString();

TempData["SoyAd"] = values.MemberSurname.ToString();

TempData["Mail"] = values.MemberMail.ToString();

TempData["KullanıcıAdı"] = values.Username.ToString();

TempData["Şifre"] = values.Password.ToString();

TempData["telefon"] = values.Telephone.ToString();

TempData["Üniversite"] = values.School.ToString();

Böylelikle bu bilgileri getirebiliriz bu session içindeki veriler string gibi bir değer atama

<td> <label style="font-weight:bolder">Adınız : </label> @TempData["Ad"].ToString()</td>

Örnek böylede çağırabiliriz

Tempdata Controllerdeki verileri view tarafına taşıyan bir araçtır.

[HttpPost]

public ActionResult Index2(Member member)

{

var kullanici = (string)Session["Mail"]; //açılan kullanıcı kimdi gibi mantık var burada

var values = entities.Member.FirstOrDefault(x => x.MemberMail == kullanici);

values.Password = member.Password;

values.Telephone = member.Telephone;

entities.SaveChanges();

return View("Index", values);

}

Böylelikle bilgileri değiştirdik

Ama amele yöntem kaçınıp böyle yapmak mantıklı

if (values != null)

{

FormsAuthentication.SetAuthCookie(values.MemberMail, false); //form da giriş yaparsak buradan maili tut cookide kullanıcı adı yerine

Session["Mail"] = values.MemberMail.ToString();

return RedirectToAction("Index", "Panel");

}

<tr>

<td>

<label style="font-weight:bolder">ID</label>

<br /> <br />

@Html.TextBoxFor(x => x.MemberId, new { @class = "form-control", @readonly = "readonly" })

</td>

</tr>

<tr>

<td>

<label style="font-weight:bolder">AD</label>

<br /> <br />

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

</td>

</tr>

Ve böylelikle yapabiliriz

MESAJ İŞLEMLERİ

// GET: Message

DBLibraryEntities libraryEntities = new DBLibraryEntities();

public ActionResult Index()

{

//sadece giriş yapan kullanıcıya gelen mesajlar

var mail = (string)Session["Mail"];

var mesajlar = libraryEntities.Message.Where(x=>x.Receiver==mail).ToList();

return View(mesajlar);

}

[HttpGet]

public ActionResult NewMessage()

{

return View();

}

[HttpPost]

public ActionResult NewMessage(Message message)

{

var senderMail = (string)Session["Mail"];//maili getirme

message.Sender = senderMail.ToString();//gönderici mail

message.Date = Convert.ToDateTime(DateTime.Now.ToShortDateString());//bugünün zamanı

libraryEntities.Message.Add(message);

libraryEntities.SaveChanges();

return RedirectToAction("Index");

}

[Authorize] // bu controller ve index tarafında authorize işlemi yapar

public class PanelController : Controller

GlobalFilters.Filters.Add(new AuthorizeAttribute()); // sistem basında authorize yapar ama

[AllowAnonymous] // bu komut eklemezsek hiç bir sayfaya gitmez bu komut ile sadece bu sayfaya authorize işlemi yapmamızı sağlarım

Hata Sayfaları yapma

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MvcLibrary.Controllers

{

public class ErrorController : Controller

{

// GET: Error

public ActionResult Page400()

{

Response.StatusCode = 400;

Response.TrySkipIisCustomErrors = true;

return View();

}

public ActionResult Page403()

{

Response.StatusCode = 403;

Response.TrySkipIisCustomErrors = true;

return View();

}

public ActionResult Page404()

{

Response.StatusCode = 404;

Response.TrySkipIisCustomErrors = true;

return View();

}

}

}

Controllerı oluştur ve yaz View oluştur ve view kodlarını hallet

System webin içinde bunları yaz

<customErrors mode="On">

<error statusCode ="404" redirect="/Error/Page404/"/>

</customErrors>

Bu kadar