

Ajax Project

Ajax ile veritabanından veri çekme yükleme

Öncelik olarak hiç bir fark olmadan veri tabanımızı migration yönetimi le oluşturuyoruz.

Sonra ise tablolar için controller oluşturup kodlamaya başlıyoruz.

```
public JsonResult YazarList()  
{  
    var data=context.Yazars.ToList();  
    return new JsonResult(data);  
}  
  
//Bu kodun sonucunda bize bir json formatında tablo verisi döner.  
  
[HttpPost]  
public JsonResult Add(Yazar yazar)  
{  
    var yaz = new Yazar()  
    {  
        YazarAdSoyad = yazar.YazarAdSoyad,  
        YazarMemleket = yazar.YazarMemleket,  
        YazarYas = yazar.YazarYas,  
        YazarKitapYaz = yazar.YazarKitapYaz  
    };  
    context.Yazars.Add(yaz);  
    context.SaveChanges();  
    return new JsonResult("Ekleme İşlemi Başarılı");  
}  
  
[HttpGet]  
public JsonResult Edit(int id)  
{  
    var data = context.Yazars.Where(e => e.YazarID == id).SingleOrDefault();  
    return new JsonResult(data);  
}
```

```

    }
    [HttpPost]
    public JsonResult Edit(Yazar yazar)
    {
        context.Yazars.Update(yazar);
        context.SaveChanges();
        return new JsonResult("İşlem Başarılı");
    }
    public JsonResult Delete(int id)
    {
        var data=context.Yazars.Where(w=>w.YazarID==id).SingleOrDefault();
        context.Remove(data);
        context.SaveChanges();
        return new JsonResult("İşlem Başarılı");
    }
}

```

//Yukardaki kodlarda belirtildiği kodlar sonucunda json formatında veriler getirir veya alır.

Sırada ise view tarafını yazmak var:

İlk olarak sayfanın kullanacağı Layout belirtilmiştir.

Sonra ise sayfanın ana iskeletini oluşturulmuştur ve oluşturulurken BOOTSTRAP kullanılmıştır.

i Sayfayı kodlarken dikkat edilmesi gereken nokta verilen İD ve CLASS isimlerinin mantıklı ve akılda kalıcı olması gereklidir.

```

@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<button class="btn btn-sm btn-outline-primary m-2" id="btnAddYazar">Yeni Yazar Ekle</button>
<table class="table table-dark rounded">
    <thead>
        <tr class="text-center">
            <th>Yazar Id</th>

```

```
        <th>AdSoyad</th>
        <th>Yazar Memleket</th>
        <th>Yazar Yas</th>
        <th>Yazar Kitap Sayısı</th>
        <th>İşlemler</th>
    </tr>

</thead>
<tbody class="text-center" id="table_date">
</tbody>
</table>
<div class="modal fade" id="yazarModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h2 class="modal-title" id="ustModal"></h2>
                <button type="button" class="close" onclick="HideModalPopUp()" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form method="post">
                    <div class="form-group" id="yazarIDdiv">
                        <label for="recipient-name" class="col-form-label">Yazar ID:</label>
                        <input type="text" class="form-control" id="yazarID">
                    </div>
                    <div class="form-group">
                        <label for="recipient-name" class="col-form-label">Yazar Adı:</label>
                        <input type="text" class="form-control" id="yazarAd">
                    </div>
                    <div class="form-group">
                        <label for="recipient-name" class="col-form-label">Yazar Memleket:</label>
                        <input type="text" class="form-control" id="yazarMemleket">
                    </div>
                    <div class="form-group">
                        <label for="recipient-name" class="col-form-label">Yazar Yaşı:</label>
```

```

        <input type="text" class="form-control" id="yazarYas">
    </div>
    <div class="form-group">
        <label for="recipient-name" class="col-form-label">Yazar Kitap Sayısı:</label>
        <input type="text" class="form-control" id="yazarKS">
    </div>
</form>
</div>
<div class="modal-footer">
    <button class="btn btn-primary" onclick="yazarAdd()" id="addBtn">Save</button>
    <button class="btn btn-warning" onclick="update()" style="display:none" id="updateBtn">Update</button>
    <button class="btn btn-danger btn-default" onclick="HideModalPopUp()">Close</button>
</div>
</div>
</div>
<div>
    <input type="hidden" id="urlYazarData" value="@Url.Action("YazarList","Yazar")" />
</div>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.esm.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.esm.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.js"></script>
<a href="~/lib/jquery/dist/jquery.min.map"></a>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/jquery/dist/jquery.js"></script>
<script src="~/js/jsyazar.js"></script>

```

Şimdi ise asıl kısma geldik JS,JQUERY ve AJAX kodlarına



Kod dosyalarını eklerken sıralamasının düzgün olmasına dikkat ediniz.

Kod dosyamızı WWW alanındaki JS alanına oluşturuyoruz ki karışıklık olmasın.

Kodta: Safya hazır olduğu zaman çalıştırılması gereken metot belirtilmiştir.

```
$(document).ready(function () {  
    showYazarData();  
});
```

Kodta: Verilen web adresin değerlerini alır, GET yöntemi ile verileri sadece getireceğimizi söyleriz, veri tipi olarak ise JSON olacağı belirtilir. Sonra gelecek içeriğin türünü belirtiriz ve verinin dili gibi özellikleri belirtiriz.

```
function showYazarData() {  
    var url = $('#urlYazarData').val(); //Deger nereden gelecek  
    $.ajax({  
        url: url,  
        type: 'Get', //İşlem türü  
        dataType: 'json', //format türü  
        contentType: 'application/json;charset=utf-8;', //hangi veri türünde gelecek  
        success: function (result, status, xhr) { //Başarılı olunca çalışacak  
            var object = ''; //Degerler buraya atanacak.  
            $.each(result, function (index, item) { //gelen degerleri tabloya tek tek atabilmek için foreach kullanırız  
                object += '<tr>'; //Sonra ise gelen degerleri HTML türünde object değişkenine atarız  
                object += '<td>' + item.yazarID + '</td>';  
                object += '<td>' + item.yazarAdSoyad + '</td>';  
                object += '<td>' + item.yazarMemleket + '</td>';  
                object += '<td>' + item.yazarYas + '</td>';  
                object += '<td>' + item.yazarKitapYaz + '</td>';  
                object += '<td><a href="#" class="btn btn-primary" onclick="edit(' + item.yazarID + ');">Edit</a> || <a href="#" class="btn btn-danger" onclick="Delete(' + item.yazarID + ');">Delete</a></td>';  
                object += '</tr>';  
            });  
            $('#table_date').html(object) // En son olarak ise object değişkenini Tablo buza göndeririz ve böylelikle deger yazdırılmış olur.  
        }, error: function () { //Hatalı işlemde buraya girer ve bildirim verir.
```

```
        alert("Data yüklenemedi")
    }
});
}
```

Listeleme işlemleri bu kadardı.

Şimdi ise bazı işlemleri yapacağız:

```
//Burada kullanıcının ekleme işlemi yapacağı zaman ekleme ekranının açılmasını sağlayan kodu yazıyoruz
$("#btnAddYazar").click(function () {
    $("#yazarModal").modal('toggle');//Ekran açma-kapama
    $("#yazarIDdiv").hide();//istenilen alanı gizleme
    $("#AddYazar").css('display', 'block');//Button görünür yapma
    $("#updateYazar").css('display', 'none');//Button gizleme
    $("#ustModal").text('Yeni Yazar Ekle');//Açılacak alanın üst yazısı
});

function ClearAl() { //Bütün alanları temizler
    $("#yazarID").val("");
    $("#yazarAd").val("");
    $("#yazarMemleket").val("");
    $("#yazarYas").val("");
    $("#yazarKS").val("");
}

HideModalPopUp = () => { //açılır alanı Gizler
    $('#yazarModal').modal('hide');
}
```

İşlemler bu kadardı

Yeni kullanıcı ekleme işlemi:

Önce kullanıcıdan alınan değerleri bir değişkene atıyoruz.

Sonra ise bunları AJAX yardımıyla veri tabanına POST(veri tabanına veri ekleme) işlemi yapıyoruz.

```
function yazarAdd() {  
    var objdata = {  
        YazarAdSoyad: $("#yazarAd").val(),  
        YazarMemleket: $("#yazarMemleket").val(),  
        YazarYas: $("#yazarYas").val(),  
        YazarKitapYaz: $("#yazarKS").val()  
    }  
    $.ajax({  
        url: "/Yazar/Add", //eklenecek adres  
        type: "post", //gönderim türü  
        data: objdata, //gidecek veri  
        contentType: 'application/x-www-form-urlencoded;charset=utf-8;', //ne tür bir veri  
        dataType: "json", //hangi veri türünde gidecek  
  
        success: function () { //başarılı olduğu zaman çalışacak kod  
            alert("Kayıt başarılı");  
            HideModalPopUp();  
            showYazarData();  
        },  
        error: function () { //başarısız olduğu zaman çalışacak kod  
            alert("İşlem gerçekleştirilemedi");  
        }  
    })  
}
```

Ekleme işlemi bu kadar.

Düzenle butonuna basılınca neler olacağının kodları:

```
function edit(id) { //id ye göre veriyi getirir
    $.ajax({
        url: "/Yazar/Edit/" + id, //Verinin gelecegi adres
        type: "get", //işlem türü
        contentType: 'application/json;charset=utf-8;', //verinin nasıl gelecegi
        dataType: 'json', //veri formatı

        success: function (response) { //gelen veriyi açılır pencereye aktarıyoruz.
            $("#yazarModal").modal('show'); //açılır pencereyi açıyoruz.
            $("#yazarID").val(response.yazarID);
            $("#yazarAd").val(response.yazarAdSoyad);
            $("#yazarMemleket").val(response.yazarMemleket);
            $("#yazarYas").val(response.yazarYas);
            $("#yazarKS").val(response.yazarKitapYaz);
            // $("#yazarIDdiv").hide();
            $("#addBtn").css('display', 'none');
            $("#updateBtn").css('display', 'block');
            $("#ustModal").text('Yazar Güncelle');
        },
        error: function () { //İşlem başarısız ise olması gereken
            alert("Veriler gelmedi, Hata Oluştı")
        }
    })
}
```

Açılır pencereye verileri getirdik.

Güncelleme işlemi kodları:

```
function update() {
    var objdata = { //Açılır pencereye doldurulan verileri bir değişkene atadık
```



```

        YazarID: $("#yazarID").val(),
        YazarAdSoyad: $("#yazarAd").val(),
        YazarMemleket: $("#yazarMemleket").val(),
        YazarYas: $("#yazarYas").val(),
        YazarKitapYaz: $("#yazarKS").val()
    }
    $.ajax({
        url: "/Yazar/Edit/", //verinin gidecegi adres
        type: "post", //işlem türü
        data: objdata, //deger
        contentType: 'application/x-www-form-urlencoded;charset=utf-8;', //deger türü
        dataType: 'json', //veri formatı
        success: function () { //işlem başarılı olursa olması gereken
            alert("İşlem Başarılı, Güncelleme Tamam")
            HideModalPopUp(); //alanı gizle
            ClearAl(); //alanları sil
            showYazarData(); //Listeyi göster
        },
        Error: function () { //İşlem başarısız olursa olması gerken
            alert("Güncellem işlemi başarısız")
        }
    })
}

```

Güncelleme kodu bu kadardı

Silme kodu:

```

function Delete(id) { //İD alıyor ve onun üzerinde işlem yapıyor
    if (confirm("Silmek İstediginize Emin Misiniz")) { //Bildirim geliyor eminmisiniz diye
        $.ajax({
            url: "/Yazar/Delete/" + id, //baglantı adresi
            success: function () { //Başarılı olursa olacaklar
                alert("Silme işlemi başarılı");
            }
        })
    }
}

```

```

        showYazarData(); //sildikten sonra listeyi gösterir.
    },
    Error: function () { //Başarısız olursa olacaklar.
        alert("Silme Yapılamadı");
    }
}
})
}
}
}

```

Silme kodu bu kadardı.

Bu şekilde işlemleri yaparsak Sayfalar düzgün bir şekilde çalışacaktır.

Login Sayfası Ekleme:

Önce Kullanıcı tablosu oluştur

```

public class User
{
    [Key]
    public int kID { get; set; }
    public string kAd { get; set; }
    public string kSifre { get; set; }

}

```

Controller kodlamasına geldik:

```

public readonly ApplicationDbContext context;
public AccountController(ApplicationDbContext context) //veri tabanı bağlantıları ile tablo arasındaki ilişkiyi kuruyoruz.
{
    this.context = context;
}
public IActionResult Index()
{

```

```

        return View();
    }
    public IActionResult Login()
    {
        return View();
    }
    [HttpPost]
    public IActionResult Login(User user)
    {
        if (ModelState.IsValid) //Eger bir sorun yoksa yani daha hatalı giriş yapılmadı ise diyoruz.
        {
            var data = context.Users.Where(e => e.kAd ==user.kAd).SingleOrDefault();//kullanıcı adını kontrol ediyoruz
            if (data != null) //kullanıcı adı null değil ise
            {
                bool isValid = (data.kAd == user.kAd && data.kSifre == user.kSifre);//girilen bilgiler eşleştigini kontrol ediyoruz.
                if (isValid) //deger dogru ise
                {
                    var identity = new ClaimsIdentity(new[] { new Claim(ClaimTypes.Name, user.kAd) },
CookieAuthenticationDefaults.AuthenticationScheme);//çerez şeklinde kullanıcın adını tutuyor
                    var prinp = new ClaimsPrincipal(identity); //kullanıcının bilgileri bu diyerek yeni nesneye aktarır.

                    HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, prinp); //kullanıcı başarılı bir şekilde oturum
açtığını belirtiyoruz.
                    HttpContext.Session.SetString("UserName", user.kAd); //kullanıcı bilgilerini tutuyoruz ilerde kullanmak için. İçeride bu
bilgiller ile kalacak
                    return RedirectToAction("Index", "Yazar");//işlem başarılı olursa belirlenen sayfaya gidecek.
                }
                else
                {
                    TempData["errorpsw"] = "Hatalı Şifre";//mesajları diğer sayfalara taşımak için kullanılır.
                    return View(user);
                }
            }
        }
        else
        {
            TempData["errorUsername"] = "Hatalı Kullanıcı adı";//mesajları diğer sayfalara taşımak için kullanılır.

```

```
        return View(user);
    }
}
else
{
    return View();
}
}
public IActionResult Logout()
{
    HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme); //kullanıcının sistemden çıkmasını sağlar.
    var storedCookies = Request.Cookies.Keys; //ifadesi, gelen HTTP isteğindeki çerezlerin anahtarlarını içeren bir koleksiyon döndürür. Bu,
    genellikle HTTP isteği içindeki çerezlerin isimlerine erişmek veya bu çerezler üzerinde işlem yapmak için kullanılabilir.
    foreach (var key in storedCookies)
    {
        Response.Cookies.Delete(key); //gelen çerezleri yok ediyoruz
    }
    return RedirectToAction("Login", "Account");
}
public IActionResult SingUp()
{
    return View();
}
[HttpPost]
public IActionResult SingUp(User user) //yeni bir kullanıcı eklemek için kullanırız.
{
    if (ModelState.IsValid) //yine aynı şekilde öncesinde hatalı bir işlem yapılmadıysa
    {
        var data = new User() //verileri veriyoruz
        {
            kAd = user.kAd,
            kSifre = user.kSifre
        };
        context.Users.Add(data); //ekleme işlemi yapıyoruz
        context.SaveChanges(); //işlemi veritabanına gönderiyoruz
        TempData["successMessage"] = "başarılı";
    }
}
```

```

        return RedirectToAction("Login");
    }
    else
    {
        TempData["errorMessage"] = "Başarısız";
        return View(user);
    }
}

```

Controller işlemleri bu kadar.

Program CS ye gelerek login işlemlerini yaptığımızı belirtmemiz gerek bunun için:

```

builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme).AddCookie(option => //AddAuthentication-metodu, kimlik
doğrulama ayarlarını yapılandırmak için kullanılır.
{ //AddCookie-metodu, cookie tabanlı kimlik doğrulama şemasını ekler.
    option.ExpireTimeSpan = TimeSpan.FromMinutes(60 * 1); //ExpireTimeSpan - var sayılan oturum süresini belirler. cookiebuilder nesnesinden
türetilir.
    option.LoginPath = "/Account/Login"; //giriş yapamadığında gideceği sayfa
    option.AccessDeniedPath = "/Account/Login"; //AccessDeniedPath-kullanıcının erişim izni olmadığında yönlendirileceği adresi belirtir.

});
builder.Services.AddSession(option => //AddSession-metodu, ASP.NET Core uygulamasında oturum yönetimi için gerekli olan ayarları yapılandırmak için
kullanılır.
{

    option.Cookie.IsEssential = true; //Cookie.IsEssential, oturum çerezi (session cookie) için temel bir çerez olup olmadığını belirtir.

});

app.UseAuthentication();//Bu kod, uygulamanın kimlik doğrulama işlemlerini etkinleştirir
app.UseAuthorization();// Bu kod, uygulamanın yetkilendirme işlemlerini etkinleştirir.

```

Program CS kodları bu kadardı.

Login View kodları: Bu alanlarda pek anlatılacak bir şey yok anlaşılır diye düşünüyorum(biraz bilgisi varsa)

```
@model odev_coreAjaxProject.Models.Account.User
@{
    ViewData["Title"] = "Login";
    Layout = "~/Views/Shared/_Layoutlgn.cshtml";
}
<div class="container">
    <div class="row mt-lg-5">
        <div class="col-4 offset-4 border border-primary rounded pl-5 pr-5 pb-5">
            <div class="text-center">
                <a class="text-light btn btn-primary font-weight-bold"
                    asp-action="SingUp" asp-controller="Account">Sign Up</a>
            </div>
            <form method="post" asp-action="Login">
                <div class="form-group">
                    <label asp-form="Username"></label>
                    <input type="text" asp-for="kAd" placeholder="Enter username" class="form-control" autocomplete="off" />
                    <span class="text-danger">@TempData["errorUsername"]</span>
                </div>

                <div class="form-group">
                    <label asp-form="Password"></label>
                    <input id="password" type="password" asp-for="kSifre" placeholder="Enter password" class="form-control" autocomplete="off" />
                    <span class="text-danger">@TempData["errorpsw"]</span>

                    <i id="open" class="fa fa-eye float-right" style="margin-top:-22px;margin-right:6px;cursor:pointer;"></i>
                    <i id="closed" class="fa fa-eye-slash float-right" style="margin-top:-22px;margin-right:6px;cursor:pointer;display:none;"></i>
                </div>
                <div class="form-group">
                    <input type="submit" class="btn btn-block btn-primary mb-2" value="Login" />
                </div>
            </form>
        </div>
    </div>
</div>
```

```

        </form>
        <div class="text-center">
            <a class="text-primary" asp-action="ForgotUsername">Forgot Username?</a><a class="text-primary" asp-action="ForgotPassword">Forgot
Password</a>
        </div>
    </div>
</div>
<script src="~/lib/jquery/dist/jquery.js"></script>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $("#open").click(function () {
            $("#open").hide();
            $("#closed").show();
            $("#password").attr("type", "text");
        });
        $("#closed").click(function () {
            $("#closed").hide();
            $("#open").show();
            $("#password").attr("type", "password");
        });
    });
</script>

```

SingUp view kodları: Bu alanlarda pek anlatılacak bir şey yok anlaşılır diye düşünüyorum(biraz bilgisi varsa)

```

@model odev_coreAjaxProject.Models.Account.User
@{
    ViewData["Title"] = "SignUp";
    Layout = "~/Views/Shared/_LayoutIgn.cshtml";
}
<div class="container">
    <div class="row">
        <div class="col-6 offset-3 border border-primary rounded pl-5 pr-5 pb-5">

```

```
<div class="text-center">
  <h1 class="text-info">User Registration Form</h1>
</div>
<form method="post" asp-action="SingUp">
  <div class="form-group">
    <label asp-for="kAd"></label>
    <input type="text" asp-for="kAd" placeholder="Enter username" class="form-control" autocomplete="off" />

  </div>
  <div class="form-group">
    <label asp-for="kSifre"></label>
    <input type="text" asp-for="kSifre" placeholder="Enter Password" class="form-control" autocomplete="off" />

  </div>
  <div class="form-group">
    <input type="submit" class="mt-2 btn btn-success btn-lg mb-1 float-right" value="Registration" />
  </div>
</form>
</div>
</div>
</div>
```