

KOCAELİ ÜNİVERSİTESİ
KOCAELİ MESLEK YÜKSEKOKULU

BİLGİSAYAR PROGRAMCILIĞI

PROJE

EVRAK KAYIT OTOMASYONU

EMRE TÜMER

Öğretim Görevlisi Mustafa OF

MAYIS 2021

ÖNSÖZ VE TEŞEKKÜR

Bitirme projesi kapsamında fikir ve görüşleri ile yol gösteren danışman hocam Sayın Mustafa Of'a ve proje süresince yardımlarını esirgemeyen sevgili hocam İsmail GÜRSOY'a da teşekkürü bir borç bilirim.

Mayıs-2021

Emre TÜMER

İçindekiler

ÖNSÖZ VE TEŞEKKÜR	1
ÖZET	3
ABSTRACT	4
GİRİŞ	5
1.GENEL BİLGİLER	6
2.SONUÇLAR VE ÖNERİLER	24
KAYNAKLAR	25
ÖZGEÇMİŞ	26

EVRAK KAYIT OTOMASYONU

ÖZET

Günümüzde İnternetin gelişimi birçok alanda değişiklik ve yeniliklerin oluşmasına imkân sağlamıştır. Bu alanlardan biri de Elektronik Ticaret alanı alanıdır. Elektronik Ticaret'in gelişimi ve değişimi İnternet'ten sonra büyük ölçüde değiştiren ve geliştiren ise Mobil Dünyadaki gelişmeler ve değişimler olmuştur. Mobil Araçların gelişimi ve yaygınlaşması ile birlikte kullanıcıların İnternet'e ve dolayısı ile Elektronik Web Sitelerine ulaşmaları ve alışveriş yapma oranlarında önemli artış olmuştur. Elektronik Ticaret (E-ticaret) alanında faaliyet göstermek isteyen firmalar tanıtımlarını yapabilecekleri ve ürün veya hizmetlerini sunabilecekleri tasarım ve yazılım açısından güçlü ve etkileşimli Elektronik Ticaret Web sitesine sahip olmak istemektedirler. Web site tasarımında kullanılan ve Microsoft tarafından geliştirilen C sharp, ASP (Active Server Page- Aktif Sunucu Sayfası), MVC (Model View Controller – Model Görünüm Kontrol) gibi teknolojilerin gelişimi E-ticaret sitelerinin daha üst seviyede güvenli ve daha rahat tasarlanıp kodlanmasını sağlamaktadır. Bu çalışmada geliştirilen dinamik web içeriklerinin kullanılmasıyla MVC teknolojisini temel alan bir E-ticaret sayfası geliştirilmiştir. Model, görünüm ve kontrol bölümlerinin ayrı ayrı gerçekleştirilmesi ile ön taraf (Frontend) ve arka tarafa (Backend) yapılar birbirinden ayrı ayrı olarak geliştirilmiştir. Ön taraf bölümü ASP kaynak kodları ile geliştirilen çalışmada arka taraf bölümünde kontrol ve veri tabanı yer almaktadır. Bu tasarım ile kod optimizasyonu, kodun genişletilmesi, kodun yeniden kullanılması ve kodun güncellenmesi imkânları sunulmaktadır.

Anahtar Kelimeler: Model-View-Controller, ASP, Web Programlama, Dağıtık sistemler, Bilgi teknolojileri

DOCUMENT REGISTRATION AUTOMATION

ABSTRACT

Nowadays, the development of the Internet has allowed changes and innovations in many areas. One of these areas is the area of Electronic Commerce area. The developments and changes in the Mobile World have been the major factors that have changed and improved the development and change of electronic commerce after the internet. With the development and diffusion of Mobile Vehicles, there has been an important increase in the rate of the users reaching the Internet and therefore the Electronic Websites and making purchases. Companies wishing to operate in the field of electronic commerce want to have a strong and interactive Electronic Commerce Web site in terms of design and software that they can promote and offer their products or services. The development of technologies such as C #, ASP (Active Server Page) and MVC (Model View Controller), which are used in website design and developed by Microsoft, enable e-commerce sites to be designed and coded at a higher level, more secure and more comfortable. Using the dynamic web contents developed in this study, an e-commerce page based on MVC technology was developed. Front End and Back End structures have been developed separately with the realization of model, view and control parts separately. In the project whose Frontend part is developed with ASP source codes, there is a Controller and a database in the Backend part. With this design, opportunities are provided for code optimization, expansion and reuse of code, and updating the code.

Keywords: Model-View-Controller, ASP, Web Programming, Distributed systems, Information technologies

GİRİŞ

MVC uzun yıllardır birçok framework'te ve programlama dilinde kullanılmış (Java, PHP, vb.) ve olgunlaşmış bir desendir. ASP.NET MVC sayesinde .Net framework dilleri ile MVC pattern kullanılarak hızlı çalışan, test edilebilir, tekrar kullanılabilir parçaları olabilen web uygulamaları geliştirilebilmesi sağlanmıştır.

1. GENEL BİLGİLER

appsettings.json:

Açıklama: Veritabanını Sql'e bağlama sayfasıdır.

```
{
  "ConnectionStrings":
  {
    "DefaultConnection": "Server=LAPTOPEPIBUCNH;Database=DbEvrakTakip;Trusted_Connection=
    True"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

_Layout Sayfası:

Açıklama: Sitedeki kullanıcı ve admin sayfalarında görüntülenmesini istenen görüntüler için kodlardır.

```
<body class="search-background">
<header>
<nav class="navbar navbar-expand-sm navbar-togglerable-sm navbar-dark bg-dark border-
bottom box-shadow mb-3">
<div class="container">
<a class="navbar-brand" asp-area="" asp-page="/Index">

<label class="text-info" style="cursor:pointer">Evrak Takip Uygulaması</label></a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target=".navbar-collapse" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
<partial name="_LoginPartial" />
<ul class="navbar-nav flex-grow-1">
@if (User.IsInRole(StatikRoller.AdminKullanici))
{
<li class="nav-item">
<a class="nav-link text-info" asp-area="" asp-page="/EvrakTipleri/Index">Evrak
Tipleri</a>
</li>
<li class="nav-item">
<a class="nav-link text-info" asp-area="" asp-
page="/Kullanicilar/Index">Kullanicilar</a>
</li>
}
@if (User.IsInRole(StatikRoller.MusteriKullanici))
{
```

```

<li class="nav-item">
<a class="nav-link text-info" asp-area="" asp-page="/Mevraklar/Index">Evraklar</a>
</li>
}
</ul>
</div>
</div>
</nav>
</header>

```

Evrak Tipi Modeli:

Açıklama: Evrak Tipi oluştururken istenen bilgileri isteyen kodlardır.

```

namespace EvrakTakip.Models
{
    public class EvrakTipi
    {
        public int Id { get; set; }

        [Required(ErrorMessage = "Evrak adının girilmesi gerekmektedir!")]
        public string EvrakAdi { get; set; }

        [Required(ErrorMessage = "Evrak fiyatının girilmesi gerekmektedir!")]
        public double EvrakFiyati { get; set; }
    }
}

```

Evrak Modeli:

Açıklama: Evrak kayıt ederken istenen bilgileri isteyen kodlardır.

```

namespace EvrakTakip.Models
{
    public class Evrak
    {
        [Key]
        public int Id { get; set; }

        [Required]
        public string Gonderen { get; set; }

        [Required]
        public string Alici { get; set; }

        [Required]
        public string Model { get; set; }

        public string EvrakTipi { get; set; }

        [Required]
        public int Yil { get; set; }

        [Required]
        public double EvrakSaatSayac { get; set; }

        public string EkAciklama { get; set; }
    }
}

```



```

        public string KullaniciId { get; set; }

        [ForeignKey("KullaniciId")]
        public virtual ApplicationUser ApplicationUser { get; set; }
    }
}

```

ApplicationUser Modeli:

Açıklama: Kullanıcı oluştururken istenen bilgileri gösteren kodlardır.

```

namespace EvrakTakip.Models
{
    public class ApplicationUser : IdentityUser
    {
        public string AdSoyad { get; set; }
        public string Adres { get; set; }
        public string Sehir { get; set; }
        public string PostaKodu { get; set; }
    }
}

```

ApplicationDbContext:

Açıklama: Oluşturulan modellerin veritabanına bağlamak için tanımlama sayfasıdır.

```

namespace EvrakTakip.Data
{
    public class ApplicationDbContext : IdentityDbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }

        public DbSet<EvrakTipi> EvrakTipi { get; set; }
        public DbSet<ApplicationUser> ApplicationUser { get; set; }
        public DbSet<Evrak> Evrak { get; set; }

        public DbSet<BakimHizmetKart> BakimHizmetKart { get; set; }
        public DbSet<BakimHizmetiGenel> BakimHizmetiGenel { get; set; }
        public DbSet<BakimHizmetiDetay> BakimHizmetiDetay { get; set; }
    }
}

```

Register.cshtml

Açıklama: Kayıt sayfasında istenilen bilgileri gösteren kodlardır.

```
@page
@model RegisterModel
@using EvrakTakip.Utility
@{
    ViewData["Title"] = "Kayıt";
}

<h2 class="text-info pt-2 pb-3">Yeni Hesap Oluştur</h2>
<div class="row backgroundwhite border">
    <div class="col-md-8">
        <form asp-route-returnUrl="@Model.ReturnUrl" method="post">
            <div asp-validation-summary="All" class="text-danger"></div>
            <div class="form-group">
                <div class="row">
                    <div class="col-4">
                        Ad-Soyad:
                        @*<label asp-for="Input.AdSoyad"></label>*@
                    </div>
                    <div class="col-8">
                        <input asp-for="Input.AdSoyad" class="form-control" />
                    </div>
                </div>
                <span asp-validation-for="Input.AdSoyad" class="text-danger"></span>
            </div>
            <div class="form-group">
                <div class="row">
                    <div class="col-4">
                        E-Posta:
                        @*<label asp-for="Input.Email"></label>*@
                    </div>
                    <div class="col-8">
                        <input asp-for="Input.Email" class="form-control" />
                    </div>
                </div>
                <span asp-validation-for="Input.Email" class="text-danger"></span>
            </div>
            <div class="form-group">
                <div class="row">
                    <div class="col-4">
                        Telefon Numarası:
                        @*<label asp-for="Input.PhoneNumber"></label>*@
                    </div>
                    <div class="col-8">
                        <input asp-for="Input.PhoneNumber" class="form-control" />
                    </div>
                </div>
                <span asp-validation-for="Input.PhoneNumber" class="text-danger"></span>
            </div>
            <div class="form-group">
                <div class="row">
                    <div class="col-4">
                        Adres:
                        @*<label asp-for="Input.Adres"></label>*@
                    </div>
                    <div class="col-8">
                        <input asp-for="Input.Adres" class="form-control" />
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>
```

```

        </div>
        <span asp-validation-for="Input.Adres" class="text-danger"></span>
    </div>
</div>
<div class="form-group">
    <div class="row">
        <div class="col-4">
            Şehir:
            @*<label asp-for="Input.Sehir"></label>*@
        </div>
        <div class="col-8">
            <input asp-for="Input.Sehir" class="form-control" />
        </div>
    <span asp-validation-for="Input.Sehir" class="text-danger"></span>
</div>
</div>
<div class="form-group">
    <div class="row">
        <div class="col-4">
            Posta Kodu:
            @*<label asp-for="Input.PostaKodu"></label>*@
        </div>
        <div class="col-8">
            <input asp-for="Input.PostaKodu" class="form-control" />
        </div>
    <span asp-validation-for="Input.PostaKodu" class="text-danger"></span>
</div>
</div>
<div class="form-group">
    <div class="row">
        <div class="col-4">
            Şifre:
            @*<label asp-for="Input.Password"></label>*@
        </div>
        <div class="col-8">
            <input asp-for="Input.Password" class="form-control" />
        </div>
    <span asp-validation-for="Input.Password" class="text-danger"></span>
</div>
</div>
<div class="form-group">
    <div class="row">
        <div class="col-4">
            Şifre Tekrar:
            @*<label asp-for="Input.ConfirmPassword"></label>*@
        </div>
        <div class="col-8">
            <input asp-for="Input.ConfirmPassword" class="form-control" />
        </div>
    <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
</div>
</div>
@if (User.IsInRole(StatikRoller.AdminKullanici))
{
    <div class="form-group">
        <div class="row">
            <div class="col-4">
                Role Admin Mi?
                @*<label asp-for="Input.AdminMi"></label>*@
            </div>
            <div class="col-8">

```

```

        <input asp-for="Input.AdminMi" class="form-control" />
    </div>
</div>
</div>
}
<div class="row">
    <div class="col-4">
    </div>
    <div class="col-8">
        <button type="submit" class="btn btn-primary">Kayıt Ol</button>
    </div>
</div>
</form>
</div>
</div>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

Register.cshtml.cs:

```

namespace EvrakTakip.Areas.Identity.Pages.Account
{
    [AllowAnonymous]
    public class RegisterModel : PageModel
    {
        private readonly SignInManager<IdentityUser> _signInManager;
        private readonly UserManager<IdentityUser> _userManager;
        private readonly ILogger<RegisterModel> _logger;
        private readonly IEmailSender _emailSender;
        private readonly RoleManager<IdentityRole> _roleManager;
        private readonly ApplicationDbContext _db;

        public RegisterModel(
            UserManager<IdentityUser> userManager,
            SignInManager<IdentityUser> signInManager,
            ILogger<RegisterModel> logger,
            IEmailSender emailSender,
            RoleManager<IdentityRole> roleManager,
            ApplicationDbContext db)
        {
            _userManager = userManager;
            _signInManager = signInManager;
            _logger = logger;
            _emailSender = emailSender;
            _db = db;
            _roleManager = roleManager;
        }

        [BindProperty]
        public InputModel Input { get; set; }
        public string ReturnUrl { get; set; }
        public IList<AuthenticationScheme> ExternalLogins { get; set; }
        public class InputModel
        {
            [Required]
            [EmailAddress(ErrorMessage = "Geçerli Bir Email Adresi Giriniz")]
            [Display(Name = "Email")]
            public string Email { get; set; }
        }
    }
}

```

```

        [Required]
        [StringLength(100, ErrorMessage = "{0} en az {2} ve en çok {1}
karakter uzunluğunda olmalıdır.", MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "Şifre")]
        public string Password { get; set; }
        [DataType(DataType.Password)]
        [Display(Name = "Şifre Doğrulama")]
        [Compare("Password", ErrorMessage = "Şifre ve Şifre Doğrulama
alanları eşleşmiyor.")]
        public string ConfirmPassword { get; set; }
        [Required]
        public string AdSoyad { get; set; }
        public string Adres { get; set; }
        public string Sehir { get; set; }
        public string PostaKodu { get; set; }
        [Required]
        public string PhoneNumber { get; set; }
        public bool AdminMi { get; set; }
    }
    public async Task OnGetAsync(string returnUrl = null)
    {
        ReturnUrl = returnUrl;
        ExternalLogins = (await
_signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
    }
    public async Task<IActionResult> OnPostAsync(string returnUrl = null)
    {
        returnUrl = returnUrl ?? Url.Content("~/");
        ExternalLogins = (await
_signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
        if (ModelState.IsValid)
        {
            var user = new ApplicationUser
            {
                UserName = Input.Email,
                Email = Input.Email,
                AdSoyad = Input.AdSoyad,
                Adres = Input.Adres,
                Sehir = Input.Sehir,
                PostaKodu = Input.PostaKodu,
                PhoneNumber = Input.PhoneNumber
            };
            var result = await _userManager.CreateAsync(user,
Input.Password);
            if (result.Succeeded)
            {
                if (!await _roleManager.RoleExistsAsync(StatikRoller.AdminKullanici))
                {
                    await _roleManager.CreateAsync(new IdentityRole(StatikRoller.AdminKullanici));
                }
                if (!await _roleManager.RoleExistsAsync(StatikRoller.MusteriKullanici))
                {
                    await _roleManager.CreateAsync(new IdentityRole(StatikRoller.MusteriKullanici));
                }
                if (Input.AdminMi)
                {
                    await _userManager.AddToRoleAsync(user, StatikRoller.AdminKullanici);

```

```

        ///
        var code = await
_userManager.GenerateEmailConfirmationTokenAsync(user);
        code =
WebEncoders.Base64UrlEncode(Encoding.UTF8.GetBytes(code));
        var callbackUrl = Url.Page(
            "/Account/ConfirmEmail",
            pageHandler: null,
            values: new { area = "Identity", userId = user.Id,
code = code, returnUrl = returnUrl },
            protocol: Request.Scheme);

        await _emailSender.SendEmailAsync(Input.Email, "Email
adresinizi doğrulayınız",
            $"Lütfen hesabınızı<a
href='{HtmlEncoder.Default.Encode(callbackUrl)}'>buraya tıklayarak
doğrulayınız</a>.");
        return RedirectToPage("/Kullanici/Index");
    }
    else
    {
        //!!!
        await _userManager.AddToRoleAsync(user,
StatikRoller.MusteriKullanici);

        //await _signInManager.SignInAsync(user, isPersistent:
false);

        if (User.IsInRole(StatikRoller.AdminKullanici))
        {
            return RedirectToPage("/Kullanici/Index");
        }
        else
        {
            return LocalRedirect(returnUrl);
        }
    }
}

```

Login.cshtml:

Açıklama: Giriş sayfasında istenilen bilgileri gösteren kodlardır.

```

@page
@model LoginModel
@{
    ViewData["Title"] = "Giriş";
}
<h1>@ViewData["Title"]</h1>
<div class="row">
    <div class="col-md-4">
        <section>
            <form id="account" method="post">
                <h4>Giriş Yapmak İçin Hesap Bilgilerinizi Girin.</h4>
                <hr />
                <div asp-validation-summary="All" class="text-danger"></div>
                <div class="form-group">
                    <label asp-for="Input.Email"></label>
                    <input asp-for="Input.Email" class="form-control" />
                    <span asp-validation-for="Input.Email" class="text-danger"></span>
                </div>
            </form>
        </section>
    </div>
</div>

```

```

        <div class="form-group">$ifre
            @*<label asp-for="Input.Password"></label>*@
            <input asp-for="Input.Password" class="form-control" />
            <span asp-validation-for="Input.Password" class="text-danger"></span>
        </div>
        <div class="form-group">
            <div class="checkbox">
                <label asp-for="Input.RememberMe">
                    <input asp-for="Input.RememberMe" />&nbsp; Beni Hatırla
                    @*@Html.DisplayNameFor(m => m.Input.RememberMe)*@
                </label>
            </div>
        </div>
        <div class="form-group">
            <button type="submit" class="btn btn-
primary">Giriş</button>
        </div>
        <div class="form-group">
            <p>
                <a id="forgot-password" asp-
page="./ForgotPassword">$ifremi Unuttum</a>
            </p>
            <p>
                <a asp-page="./Register" asp-route-
returnUrl="@Model.ReturnUrl">Başka Bir Kullanıcı Olarak Kaydol</a>
            </p>
        </div>
    </form>
</section>
</div>
<div class="col-md-6 col-md-offset-2">
    <section>
        <h4>@*Use another service to log in.*@</h4>
        <hr />
        @if
        {
            if ((Model.ExternalLogins?.Count ?? 0) == 0)
            {
                <div>
                    <p>
                    </p>
                </div>
            }
            else
            {
                <form id="external-account" asp-page="./ExternalLogin" asp-route-
returnUrl="@Model.ReturnUrl" method="post" class="form-horizontal">
                    <div>
                        <p>
                            @foreach (var provider in Model.ExternalLogins)
                            {
                                <button type="submit" class="btn btn-primary" name="provider"
                                value="@provider.Name" title="Log in using your @provider.DisplayName
                                account">@provider.DisplayName</button>
                            }
                        </p>
                    </div>
                </form>
            }
        }
    </section>
</div>

```

```

</div>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

_LoginPartial.cshtml:

Açıklama: Giriş kısmında istenilen bilgileri gösteren kodlardır.

```

@using Microsoft.AspNetCore.Identity
@inject SignInManager<IdentityUser> SignInManager
@inject UserManager<IdentityUser> UserManager

<ul class="navbar-nav">
@if (SignInManager.IsSignedIn(User))
{
    <li class="nav-item">
        <a class="nav-link text-info" asp-area="Identity" asp-
page="/Account/Manage/Index" title="Manage">@await
Component.InvokeAsync("KullaniciAdi")</a>
    </li>
    <li class="nav-item">
        <form class="form-inline" asp-area="Identity" asp-
page="/Account/Logout" asp-route-returnUrl="@Url.Page("/", new { area = "" })"
method="post" >
            <button type="submit" class="nav-link btn btn-link text-
info">Çıkış</button>
        </form>
    </li>
}
else
{
    <li class="nav-item">
        <a class="nav-link text-info" asp-area="Identity" asp-
page="/Account/Register">Kayıt Ol</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-info" asp-area="Identity" asp-
page="/Account/Login">Giriş Yap</a>
    </li>
}
</ul>

```


Create.cshtml:

Açıklama: Evrak oluşturmak için istenilen bilgileri içeren kodlardır.

```
@page
@model EvrakTakip.Pages.Mevraklar.CreateModel
@{
    ViewData["Title"] = "Create";
    Layout = "~/Pages/Shared/_Layout.cshtml";
}
<br />
<h2 class="text-info">Yeni Evrak Tipi Oluştur</h2>
<br />
<form method="post">
    <div class="border backgroundwhite">
        <input hidden asp-for="Evrak.KullaniciId" />
        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="form-group row">
            <div class="col-2">Gönderen:
                @*<label asp-for="Evrak.Gonderen"></label>*@
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Gonderen" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Gonderen" class="text-
danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">Alıcı:
                @*<label asp-for="Evrak.Alici"></label>*@
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Alici" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Alici" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">Evrak Modeli:
                @*<label asp-for="Evrak.Model"></label>*@
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Model" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Model" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">Evrak Tipi:
                @*<label asp-for="Evrak.EvrakTipi"></label>*@
            </div>
            <div class="col-5">
                <input asp-for="Evrak.EvrakTipi" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.EvrakTipi" class="text-
danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">Evrak Kayıt Yılı:
                @*<label asp-for="Evrak.Yil"></label>*@
            </div>
```

```

        </div>
        <div class="col-5">
            <input asp-for="Evrak.Yil" class="form-control" />
        </div>
        <span asp-validation-for="Evrak.Yil" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">Ek Açıklama:
            @*<label asp-for="Evrak.EkAciklama"></label>*@
        </div>
        <div class="col-5">
            <input asp-for="Evrak.EkAciklama" class="form-control" />
        </div>
        <span asp-validation-for="Evrak.EkAciklama" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-5 offset-2">
            <partial name="_OlusturVeGeriDonButonlari" />
        </div>
    </div>
</div>
</form>
@section Scripts {
    @{ await Html.RenderPartialAsync("_ValidationScriptsPartial"); }
}

```

Create.cshtml.cs:

```

namespace EvrakTakip.Pages.Mevraklar
{
    [Authorize]
    public class CreateModel : PageModel
    {
        private readonly ApplicationDbContext _db;
        [BindProperty]
        public Evrak Evrak { get; set; }

        [TempData]
        public string DurumMesaj { get; set; }

        public CreateModel(ApplicationDbContext db)
        {
            _db = db;
        }
        public IActionResult OnGet(string userId = null)
        {
            Evrak = new Evrak();
            {
                if (userId == null)
                {
                    var claimsIdentity = (ClaimsIdentity)User.Identity;
                    var claim =
claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
                    userId = claim.Value;
                }
            }
            Evrak.KullaniciId = userId;
        }
    }
}

```

```

        return Page();
    }
}

public async Task<IActionResult> OnPostAsync()
{
    if (!ModelState.IsValid)
    {
        return Page();
    }
    _db.Evrak.Add(Evrak);
    await _db.SaveChangesAsync();
    DurumMesaj = "Evrak Bilgileri Kaydedilmiştir!";
    return RedirectToPage("Index", new { kullanıcıId =
Evrak.KullanıcıId });
}
}
}

```

Delete.cshtml:

Açıklama: Evrak silmek için istenilen bilgileri içeren kodlardır.

```

@page
@model EvrakTakip.Pages.Mevraklar.DeleteModel
@{
    ViewData["Title"] = "Delete";
    Layout = "~/Pages/Shared/_Layout.cshtml";
}
<br />
<h2 class="text-info">Evrak Bilgisi Sil</h2>
<br />
<form method="post">
    <div class="border backgroundwhite">
        <input hidden asp-for="Evrak.Id" />
        <input hidden asp-for="Evrak.KullanıcıId" />
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Evrak.Gonderen"></label>
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Gonderen" disabled class="form-control"
/>
            </div>
        </div>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.Alici"></label>
        </div>
        <div class="col-5">
            <input asp-for="Evrak.Alici" disabled class="form-control" />
        </div>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.Model"></label>
        </div>
    </div>

```

```

        <div class="col-5">
            <input asp-for="Evrak.Model" disabled class="form-control" />
        </div>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.EvrakTipi"></label>
        </div>
        <div class="col-5">
            <input asp-for="Evrak.EvrakTipi" disabled class="form-control"
/>
        </div>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.Yil"></label>
        </div>
        <div class="col-5">
            <input asp-for="Evrak.Yil" disabled class="form-control" />
        </div>
    </div>

    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.EvrakSaatSayac"></label>
        </div>
        <div class="col-5">
            <input asp-for="Evrak.EvrakSaatSayac" disabled class="form-
control" />
        </div>
    </div>

    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Evrak.EkAciklama"></label>
        </div>
        <div class="col-5">
            <input asp-for="Evrak.EkAciklama" disabled class="form-
control" />
        </div>
    </div>
    <div class="form-group row">
        <div class="col-5 offset-2">
            <div class="row">
                <div class="col-6">
                    <input type="submit" class="btn btn-danger form-
control" value="Sil" />
                </div>
                <div class="col-6">
                    <a asp-page="Index" asp-route-
kullaniciId="@Model.Evrak.KullaniciId" class="btn btn-success form-
control">Listeye Geri Dön</a>
                </div>
            </div>
        </div>
    </div>
</div>
</form>

```

Delete.cshtml.cs:

```
namespace EvrakTakip.Pages.Mevraklar
{
    [Authorize]
    public class DeleteModel : PageModel
    {
        private readonly ApplicationDbContext _db;

        [TempData]
        public string DurumMesaj { get; set; }

        public DeleteModel(ApplicationDbContext db)
        {
            _db = db;
        }

        [BindProperty]
        public Evrak Evrak { get; set; }

        public async Task<IActionResult> OnGetAsync(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }

            Evrak = await _db.Evrak.Include(a =>
a.ApplicationUser).FirstOrDefaultAsync(b => b.Id == id);

            if (Evrak == null)
            {
                return NotFound();
            }

            return Page();
        }

        public async Task<IActionResult> OnPostAsync()
        {
            if (Evrak == null)
            {
                return NotFound();
            }

            var kullanıcıId = Evrak.KullanıcıId;

            _db.Evrak.Remove(Evrak);

            await _db.SaveChangesAsync();

            DurumMesaj = "Evrak Bilgisi Silinmiştir.";
            return RedirectToPage("./Index", new { kullanıcıId });
        }
    }
}
```

Edit.cshtml:

Açıklama: Evrak düzenlemek için istenilen bilgileri içeren kodlardır.

```
@page
@model EvrakTakip.Pages.Mevraklar.EditModel
@{
    ViewData["Title"] = "Edit";
    Layout = "~/Pages/Shared/_Layout.cshtml";
}
<br />
<h2 class="text-info">Evrak Bilgisi Güncelle</h2>
<br />
<form method="post">
    <div class="border backgroundwhite">
        <input hidden asp-for="Evrak.Id" />
        <input hidden asp-for="Evrak.KullaniciId" />
        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Evrak.Gonderen"></label>
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Gonderen" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Gonderen" class="text-
danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Evrak.Alici"></label>
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Alici" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Alici" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Evrak.Model"></label>
            </div>
            <div class="col-5">
                <input asp-for="Evrak.Model" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.Model" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Evrak.EvrakTipi"></label>
            </div>
            <div class="col-5">
                <input asp-for="Evrak.EvrakTipi" class="form-control" />
            </div>
            <span asp-validation-for="Evrak.EvrakTipi" class="text-
danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
```

```

        <label asp-for="Evrak.Yil"></label>
    </div>
    <div class="col-5">
        <input asp-for="Evrak.Yil" class="form-control" />
    </div>
    <span asp-validation-for="Evrak.Yil" class="text-danger"></span>
</div>
<div class="form-group row">
    <div class="col-2">
        <label asp-for="Evrak.EvrakSaatSayac"></label>
    </div>
    <div class="col-5">
        <input asp-for="Evrak.EvrakSaatSayac" class="form-control" />
    </div>
    <span asp-validation-for="Evrak.EvrakSaatSayac" class="text-
danger"></span>
</div>
<div class="form-group row">
    <div class="col-2">
        <label asp-for="Evrak.EkAciklama"></label>
    </div>
    <div class="col-5">
        <input asp-for="Evrak.EkAciklama" class="form-control" />
    </div>
    <span asp-validation-for="Evrak.EkAciklama" class="text-
danger"></span>
</div>
<div class="form-group row">
    <div class="col-5 offset-2">
        <div class="row">
            <div class="col-6">
                <input type="submit" class="btn btn-info form-control"
value="Güncelle" />
            </div>

            <div class="col-6">
                <a asp-page="Index" asp-route-
kullaniciId="@Model.Evrak.KullaniciId" class="btn btn-success form-
control">Listeye Geri Dön</a>
            </div>
        </div>
    </div>
</div>
</div>
</form>
@section Scripts {
    @{ await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

Edit.cshtml.cs:

```
namespace EvrakTakip.Pages.Mevraklar
{
    [Authorize]
    public class EditModel : PageModel
    {
        private readonly ApplicationDbContext _db;

        public EditModel(ApplicationDbContext db)
        {
            _db = db;
        }
        [BindProperty]
        public Evrak Evrak { get; set; }

        [TempData]
        public string DurumMesaj { get; set; }

        public async Task<IActionResult> OnGetAsync(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }
            Evrak = await _db.Evrak.Include(a =>
a.ApplicationUser).FirstOrDefaultAsync(b => b.Id == id);

            if (Evrak == null)
            {
                return NotFound();
            }
            return Page();
        }
        public async Task<IActionResult> OnPostAsync()
        {
            if (!ModelState.IsValid)
            {
                return Page();
            }

            _db.Attach(Evrak).State = EntityState.Modified;

            await _db.SaveChangesAsync();
            DurumMesaj = "Evrak Bilgisi Güncellenmiştir.";
            return RedirectToPage("../Index", new { kullanıcıId =
Evrak.KullanıcıId });
        }
    }
}
```


2. SONUÇLAR VE ÖNERİLER

Proje uzun uğraşlar sonucunda bitti. Projeye e-posta doğrulama sayfası eklenebilir, facebook veya başka sosyal medya platformları ile kayıt olma imkânı sağlanabilir. Bu çalışmalarımı daha sonra devam ettirmeye çalışacağım.

KAYNAKLAR

İsmail G, <https://www.youtube.com/channel/UC9BqxtiQjfzup3E0oqwGd8Q>

Yazılım kodlama, <https://www.yazilimkodlama.com/programlama/c-sql-server-veritabani-baglantis-i-select-insert-update-delete/>,

Yücedağ M, <https://www.youtube.com/watch?v=C71JUcJbXyg&t=2s>,

ÖZGEÇMİŞ

2000 yılında Karabük’de doğdu. İlk ve orta öğretimini İstanbul Sarıyer’de tamamladı. İstanbul Beşiktaş Erhan Gedikbaşı Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri bölümünden 2018 yılında mezun oldu. Bir yıl mezun olarak geçirilen zamandan sonra Eylül ayında Kocaeli Üniversitesi Kocaeli Meslek Yüksekokulu Bilgisayar Programcılığı bölümüne başladı.