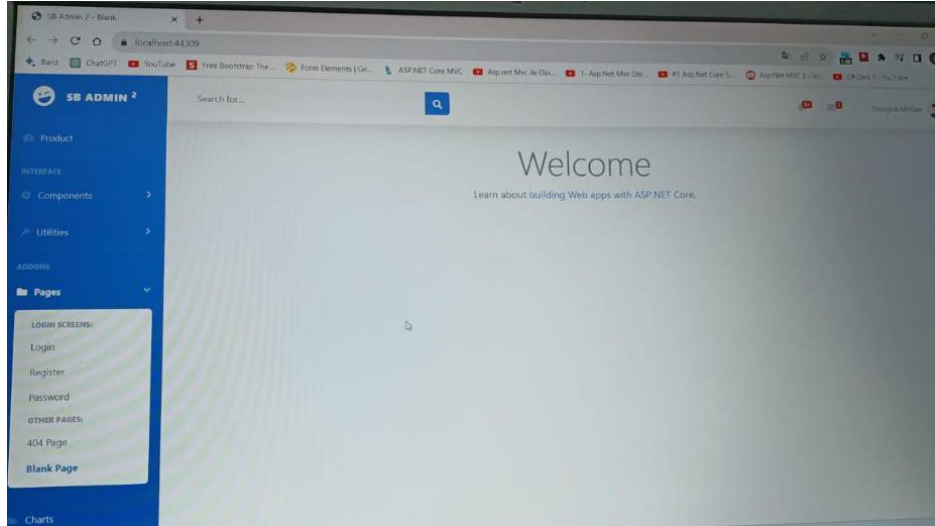


KISIM:

TARİH:21.08.2023

YAPILAN İŞ: Tanışma Toplantısı, Kurulumlar

- Stajımın ilk gününde firmayı gezdim ve IT bölümündeki çalışanlarla tanıştım. Hangi alanlarda çalıştıkları hakkında bilgi edindim. Çalıştığım alanlar ve yaptığım projeler hakkında bilgi verdim.
- Firma içi kullanılacak bir onay akış süreci projesi verildi ve bu proje hakkında toplantı yaptık. Projede firmada çalışan kişilere verilen cihazlar için tutulan tutanakların Asp.net core mvc5 kullanılarak dijitalleştirilmesinin istendiğini; form için bir sayfa istendiğini, form hazırlandığında onaylanması için mail atılması gerektiğini ve mail alan kişinin oluşturulan onay sayfasına yönlendirilmesini, onay sayfasında onaylama işlemi yapıldığında onaylandığına dair bir mail atılması istendi. Kişilerin oturum açarak kendi aldığı ürünleri görmek istediği ve form sayfasının sadece adminlere erişilebilir olması istendi. Staj sürecimde projede güncellemeler olabileceği ancak temel özelliklerinin onay akış süreci olduğu belirlendi.
- İlk olarak kurulumları tamamladım. Visual Studio, Sql server management studio kurulumlarını yaptım. Visual Studio’da , asp.net core mvc projesi oluşturdum. Bootstrap’tan bir template indirdim ve projede içe aktardım. Proje dosyasında wwwroot klasörünün içine indirdiğim template dosyasını kopyaladım. Visual studioda Shared altındaki layoutu düzenledim. Layout’a RenderBody() metodunu ekledim. Sonrasında klasörün içindeki html sayfalarını sildim ve bu sayfaları kendim oluşturdum. Layout içerisinde de uzantılarını düzenledim. wwwroot klasörünün içindeki kullanmadığım dosyaları da sildim.



KONTROL SONUCU:

KISIM:

TARİH:22.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Projeye başlarken öncelikle tutanak formu için form sayfası oluşturmaya karar verdim. ProductsController adında yeni bir controller oluşturdum. Add adında bir IActionResults oluşturdum ve view ekledim. View içerisinde kullanacağım bilgileri not alıp html tagi içerisinde head ve body taglarını oluşturdum. Head tagi içerisinde title oluşturdum ve style tagi içerisinde form sayfasının css kodları için class oluşturdum ve sonrasında body içerisinde class ile kullanarak form sayfasının görünümünü düzenledim. Body içerisinde form için h1 tagi içerisinde bir başlık ekledim. Div tagi içerisinde formu oluşturmaya başladım. Form içerisinde işyerinin adı select box ile açılır kutu olarak oluşturdum ve br tagi ile boşluk bıraktım. Teslim olunan malzeme için kullanıcıdan alabileceğim type'ı text olan input oluşturdum ve autocomplete="off" seçeneği ile daha önce yapılan işlemlerdeki bilgileri getirmesini önledim. Teslim tarihi için type="date" olan bir input oluşturdum ve autocomplete="off" yaptım. Teslim eden ve teslim alan kişilerin bilgileri için div classı içerisinde type'ı text olan input oluşturdum. Bu alanı adı, departmanı ve görevi için oluşturdum. Aynı şekilde teslim alan kişi için de oluşturdum ve bu kısmı div tagi içerisinde görünümünü düzenledim. Fotoğraf eklemek için type="file" olan bir input oluşturdum ve son olarak type="submit" olan bir buton oluşturdum.

The screenshot shows a web browser displaying a form titled "TESLİM VE TESELLÜM TUTANAĞI". The form is part of a "TUTANAK" application. The form includes the following fields and sections:

- İŞYERİNİN UNVANI:** A dropdown menu with "Seçiniz..." as the placeholder.
- TESLİM OLANUN MALZEME TANIMI:** A text input field.
- DİĞER BİLGİLER:** A text input field.
- TESLİM TARİHİ:** A date input field with the placeholder "gg.aa.yyyy".
- Mülkiyeti:** A text input field with the placeholder "ye ait olan" and a dropdown menu with "demirbaşı" as the selected option.
- Teslim Eden:** A section with three text input fields: "ADI SOYADI", "DEPARTMAN", and "GÖREVİ".
- Teslim Alan:** A section with three text input fields: "ADI SOYADI", "DEPARTMAN", and "GÖREVİ".
- ONAYLIYORUM!:** A red button at the bottom of the form.

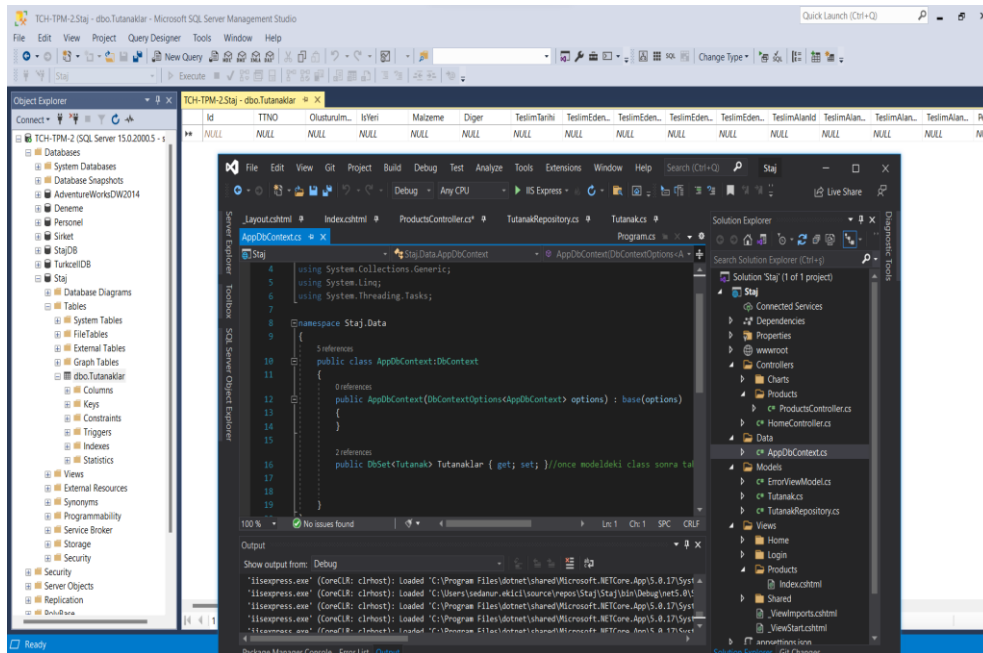
KONTROL SONUCU:

KISIM:

TARİH:23.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Projenin Database bağlantısını oluşturmayı datafirst yaklaşımıyla yapmaya karar verdim ve sql server management studioda Staj adında yeni bir tablo oluşturdum. New database ile Tutanaklar adında bir database oluşturdum ve içerisine Id yi primary key yaparak formda ve projede kullanacağım TTNO, OlusturulmaTarihi, IsYeri, Malzeme, Diger, TeslimTarihi, TeslimEdenId, TeslimEdenAd, TeslimEdenDep, TeslimEdenGorev, TeslimAlanId, TeslimAlanAd, TeslimAlanDep, TeslimAlanGorev, PersMailGonDurumu, MailOnayDurumu, OnaylamaTarihi, Resim değişkenlerini ekledim. Visual studioda model klasörü altında Tutanak adında bir class oluşturup database'de belirlediğim özellikleri ve tiplerini tanımladım. Daha sonrasında sql server object explorer ile bağlanmak istediğim databasei seçip connect dedim. Appsettings.json dosyasında ConnectionStrings bilgisini ekledim. Sql Server'da veritabanına karşılık gelecek ApplicationDbContext sınıfını oluşturdum. DbSet ile tabloyu ve model sınıfını ekledim. Program.cs sınıfında ApplicationDbContext'i ekledim.



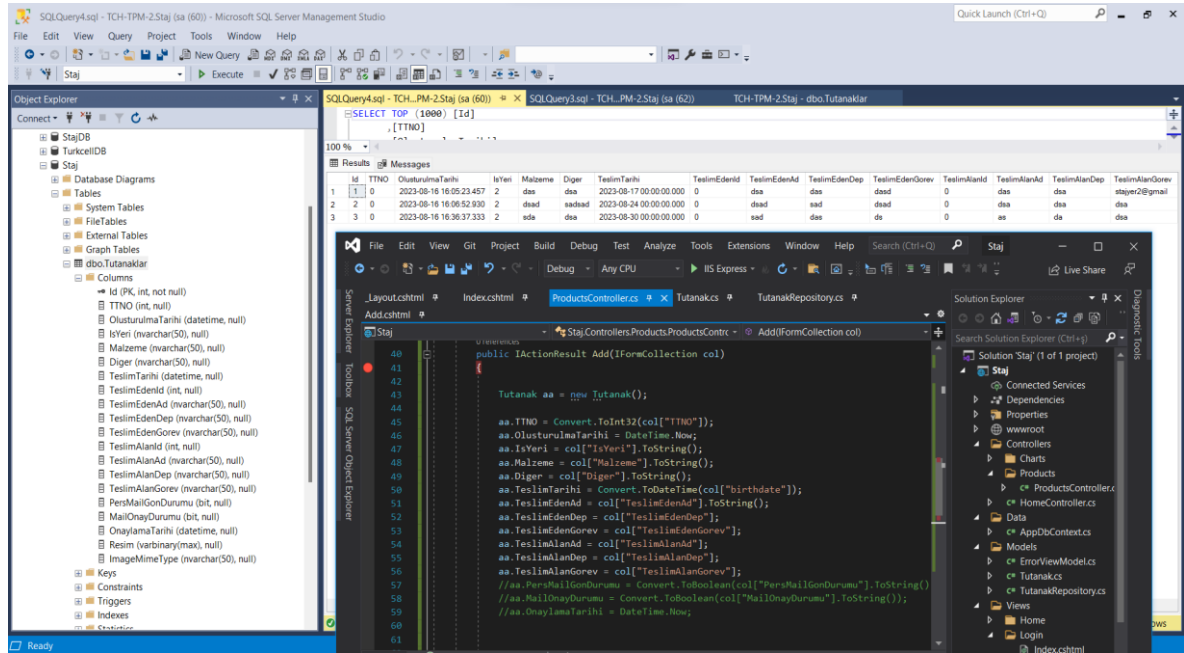
KONTROL SONUCU:

KISIM:

TARİH:24.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Projenin Database bağlantısını oluşturduktan sonra veri kaydetme işlemini yaptım. ProductsController oluşturum. Bir constructor içerisinde database nesnesi oluşturum. Add adında oluşturduğum IActionResult içerisinde return view ekledim ve veri ekleme yollarını araştırdım. Bunun için öncelikle oluşturduğum Add sayfasını düzenledim. Html.BeginForm kullanarak kullanıcıdan alınan dataları aktarmaya denedim. FormMethod'u post olarak seçtim ve productController'daki Add IActionResult olduğunu belirttim. Daha sonrasında formda kullanıcıdan alınan data bilgilerini asp-for ile IActionResults'a ilettim. Add IActionResult post içerisinde IFormCollection içerisinde col adında değişken oluşturum. Post'un dışında bir tutanak nesnesi oluşturum ve post içerisinde bu tutanak nesnesine aa.TTNO = Convert.ToInt32(col["TTNO"]); şeklinde col ile viewden controllera veriyi taşıyıp database'e ekleme işlemini yapmaya devam ettim. Oluşturduğum context nesnesi ile tutanaklara Add metodunu kullanarak tutanak nesnesini kayıt ettim ve bu yaptığım kayıt işleminin database' de kalıcı hale gelmesini sağlamak için context nesnesi üzerinden SaveChanges metodunu kullandım. Son olarak ise Index adında IActionResult içerisinde oluşturduğum \_context nesnesi ile tutanaklardaki ürünleri tolist metoduyla listeledim.



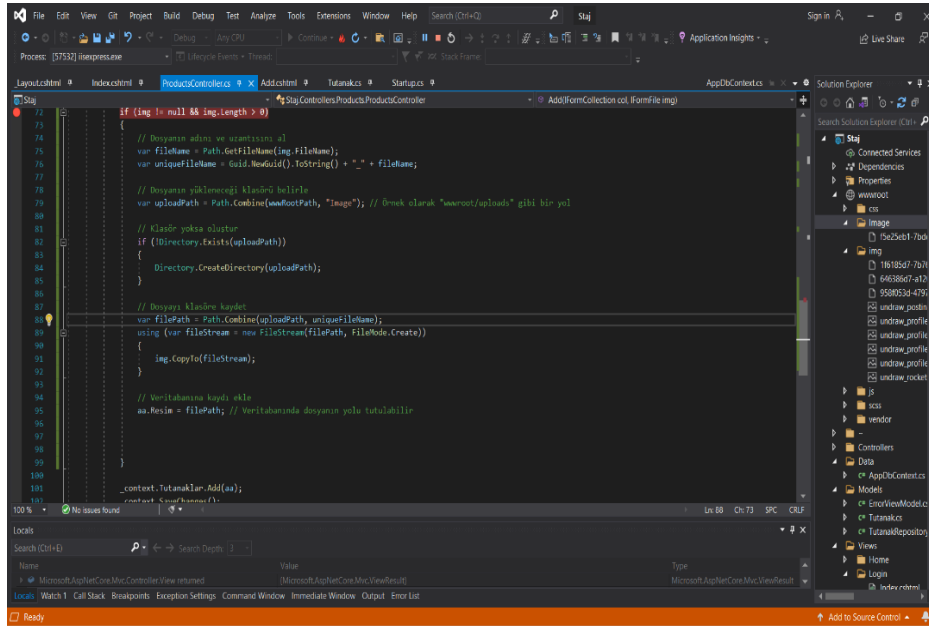
KONTROL SONUCU:

KISIM:

TARİH:25.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Dün yaptığım veri ekleme bölümünde eksiklerim vardı. İşyeri adını sayı olarak alıyordu bunun yerine productController'ın Add view'inde value değeri olarak şirket isimlerini yazdım ve databasede veriyi doğru kaydettim. Resmin yolunu databasede kayıt etme işlemini düzenledim ve resimi projede klasöre kayıt ettim. Öncelikle formda eklenen fotoğrafı proje dosyasında wwwroot altında Image klasörü oluşturup bu klasör içerisinde kaydettim. Daha sonrasında ise fotoğraf adını ve kayıt ettiğim dosya yolunu database de kayıt etme işlemini yaptım. wwwroot un yolunu alabilmek için IWebHostEnvironment kullandım. Startup'ta ConfigureServices içerisinde services.AddSingleton(\_webHostEnvironment); kodunu ekledim. Controller içerisinde Path.GetFileName ve Guid.NewGuid() ile dosya adını ve uzantısını aldım. Path.Combine ile dosyanın yükleneceği klasörü belirledim. Eğer klasör yoksa Directory.CreateDirectory ile oluşturdum. FileMode.Create ve img.CopyTo ile dosyayı klasöre kaydettim. Dosyanın yolunu ise databasede kayıt ettim. View sayfasında dosya yükleyebilmek için type="file" olan bir input oluşturdum ve id, name vererek controllerda kullanıcının yüklediği verileri aldım. Son olarak bootstrap kütüphanelerini projeye ekledim.



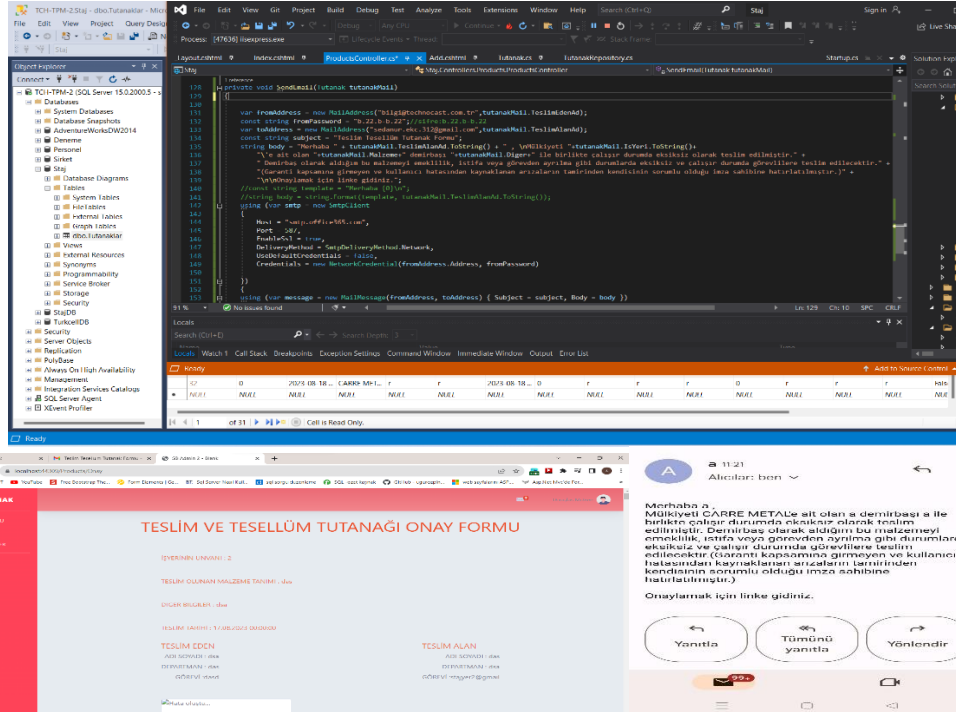
KONTROL SONUCU:

KISIM:

TARİH:28.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Add sayfasında hazırladığım form onaylandığı zaman teslim alan kişiye mail gönderdim. Add sayfasınının HttpPost kısmında IActionResult içerisinde tutanak modeli ekledim. Eğer bu model boş değilse daha sonrasında tanımladığım SendEmail() ile oluşturduğum modeli gönderdim ve databasede PersMailGonDurumu'nu true olarak kayıt ettim. TempData ile mail gönderildiğine dair kullanıcıya bilgi verdim. Mail gönderme işlemi için SendMail adında bir fonksiyon oluşturdum. Bu fonksiyona bir tutanak nesnesi gönderdim. MailAddress ile gönderilecek kişinin mailini ekledim. String olarak subject oluşturdum. String türünde link oluşturdum. Body oluşturup içerisinde dinamik olarak doldurulan formdan bilgileri aldım ve oluşturduğum linki body içerisine ekledim. using (var smtp = new SmtpClient{}) içerisinde mail gönderilmesi için port ayarlarını düzenledim. using (var message = new MailMessage(fromAddress, toAddress) { Subject = subject, Body = body, IsBodyHtml = true }) ile mail gönderilme ayarlarını düzenledim.
- Mail gönderildiğinde mail içerisindeki linke tıklandığında açılacak olan sayfa için Onay adında controller ve view oluşturup model ile verileri databaseden aldım. Oluşturduğum bu sayfada form onaylanınca onaylanan kişi ve ürün ile ilgili bilgiler yer almaktadır.



KONTROL SONUCU:

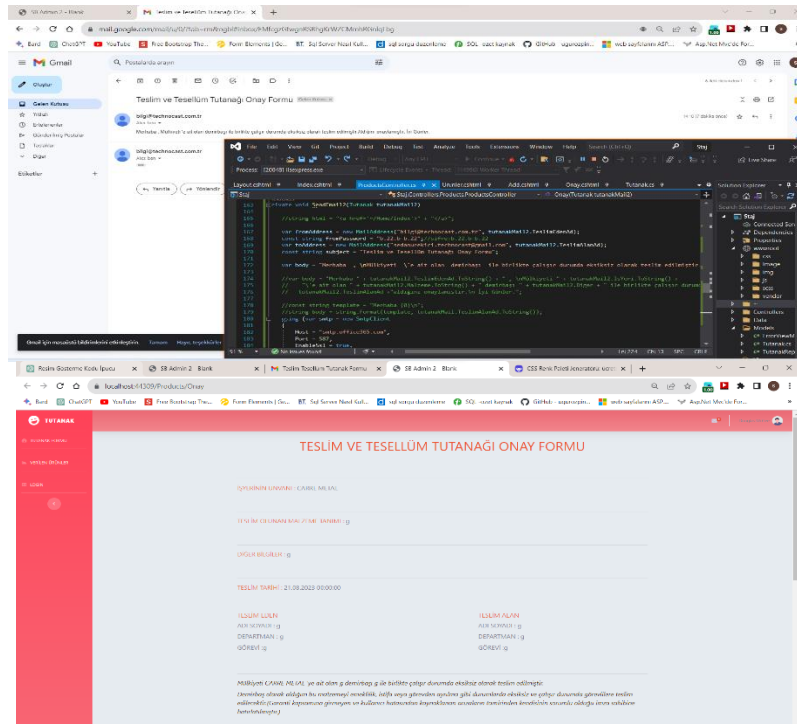


KISIM:

TARİH:29.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Bugün diğer mail gönderim fonksiyonunu oluşturdum. Bu mail ürünleri teslim alan kişi teslim aldığına dair formu onayladığında teslim eden kişiye formun onaylandığına dair bilgi amaçlı mail göndermektedir. SendMail2 adında bir fonksiyon oluşturup gönderen mail adresini ve teslim alan kişinin mail adreslerini girdim. String olarak subject yazdım. Subject yazarken fonksiyon tanımında oluşturduğum tutanak nesnesini de kullanarak formdaki bilgileri de kullanarak daha detaylı ve onaylanan forma özel bir mail oluşturdum. using (var smtp = new SmtplibClient{}) içerisinde mail gönderilmesi için port ayarlarını düzenledim. using (var message = new MailMessage(fromAddress, toAddress) { Subject = subject, Body = body, IsBodyHtml = true }) ile mail gönderilme ayarlarını düzenledim.
- Sonrasında dün oluşturduğum teslim ve teslim tutanağı onay formunu düzenledim. Tanımlamış olduğum @model List<Tutanak> ile Html.BeginForm içerisinde Model ile formda onaylanan bilgileri onay sayfasına ilettim. Div tagları içerisinde label ve input ile metin alanlarını düzenledim. Teslim eden ve teslim alan bilgilerini yan yana getirmek için iç içe divler yardımıyla classs ve style bilgilerini düzenledim. Oluşturduğum formdaki bilgi metnini model nesneni yardımıyla farklı bir div içerisinde düzenledim.



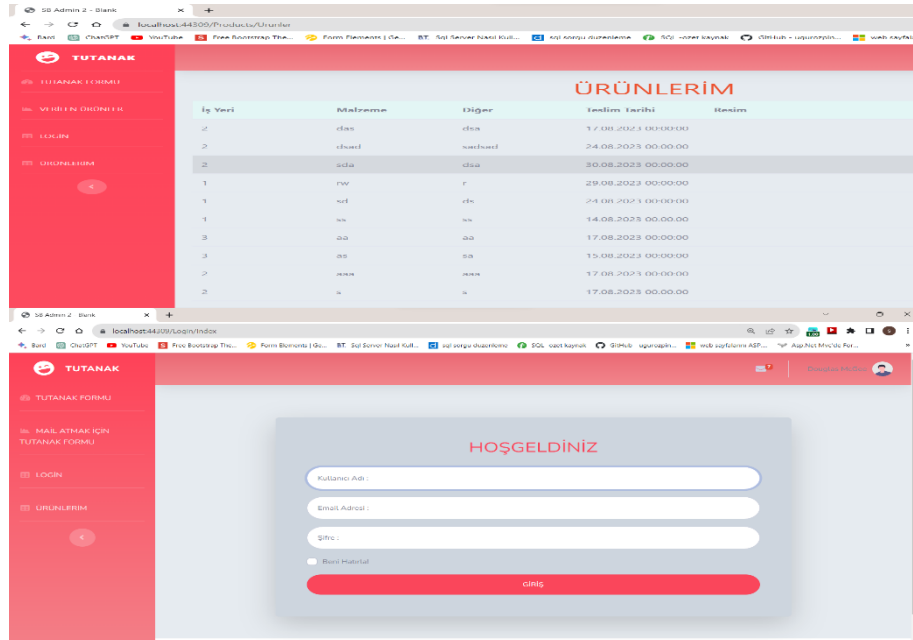
KONTROL SONUCU:

KISIM:

TARİH:31.08.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Teslim edilen ürünlerin gösterilmesi için bir Ürünlerim sayfası oluşturdum. Öncelikle ProductsControllerda Ürünler adında bir IActionResults oluşturdum. var tutanaklar = \_context.Tutanaklar.ToList(); yazarak context ile databasede kayıtlı olan Tutanaklar tablosundaki verileri tutanaklara attım ve return View'e tutanakları gönderdim. Ürünler için View oluşturdum. @model List<Tutanak> ile Model nesnelerini kullandım. <html><head> içerisinde title oluşturdum ve sonrasında kullandığım css için style classları oluşturdum. <body> içerisinde div ile bir text yazdım. Ürünleri bir tablo halinde göstermek için <table> kullandım. <thead><tr> ile ilk sütündaki başlıklar için alan oluşturdum ve her başlığı <th> içerisinde tanımladım. <tbody> içerisinde Model nesnesini kullanarak <tr><td> ile kullanıcıya göstermek istediğim bilgileri ekledim.
- Kullanıcıların sisteme giriş yapabilmesi için bir login sayfası oluşturdum. Öncelikle login sayfalarını araştırdım ve kullanıcı adı, email adresi, şifre, beni hatırla butonu ve giriş butonu eklemeye karar verdim. Login için controller klasörünün altında bir klasör oluşturup içerisinde LoginController adında bir controller oluşturdum. Bu controllerda Index adındaki IActionResultView'de view sayfası oluşturdum. Oluşturduğum View sayfası içerisinde <html><head> içerisinde title ve style bilgilerini ekledim. <body> içerisinde içi içe div tagları kullanarak kullanıcı adı, email adresi, şifre, beni hatırla butonu ve giriş butonunu oluşturdum ve osn css düzenlemelerini tamamladım.



KONTROL SONUCU:

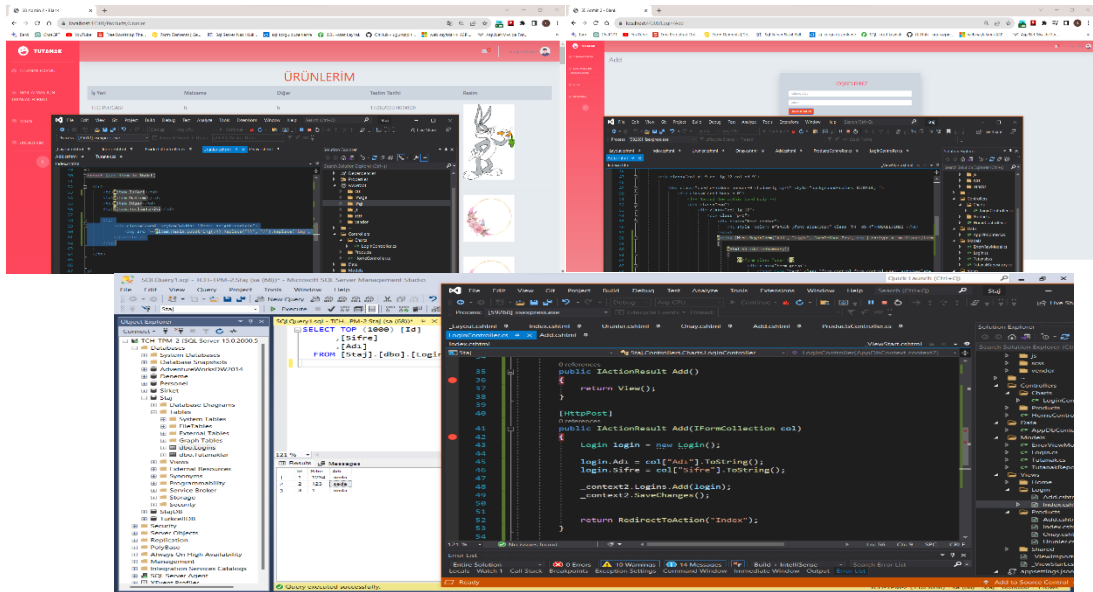


KISIM:

TARİH:01.09.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Bugün oluşturmuş olduğum Login sayfasını düzenlemeye devam ettim. Öncelikle Giriş yapan kullanıcıların bilgilerini database'de kayıt işlemlerini yapmak için Logins adında bir tablo oluşturdum. Bu tabloda Id, Adı, Sifre bilgilerini kayıt ettim. Id'yi primary key olarak belirledim. Tablonun projede database bağlantısını gerçekleştirdim. Model klasöründe Logins adında yeni bir class oluşturup değerleri tanımladım. Sonrasında ApplicationDbContext'te Logins tablosunu tanımladım. LoginController'da bir constructor oluşturup içerisinde \_context nesnesini tanımladım. Index IActionResult'ta kullanıcıları listeleyp return view'e aktardım. Sonrasında Add adında yeni bir IActionResult oluşturup kullanıcılar giriş yaptığında açılacak olan sayfa işlemlerini ve database'de kullanıcı verilerini kayıt etme işlemlerini yaptım. Öncelikle hazırlamış olduğum login sayfasını Add için oluşturduğum View'e ekledim ve bu sayfayı düzenledim. Add sayfası içerisinde model ile Login'i aldım. Kullanıcının girdiği değerleri controllara iletebilmek için html BeginForm ekledim. Controller'ı, IActionResults'u ve Add sayfasında butona tıklandığında gerçekleşecek işlemleri düzenlemek için oluşturduğum post IActionResults' kullanacağımı belirttim. Login sayfasında asp-for ile kullanıcının girdiği Adı ve Sifre değişkenlerini yazarak girilen değerlere controller'dan ulaştım. HttpPost içerisinde new ile yeni login oluşturup kullanıcı bilgilerini database'e kaydettim ve sayfayı Index sayfasına yönlendirdim. Ürünler sayfasında ve Form doldurulduğunda atılan mail ile açılan Onay sayfasındaki resimleri view'de göstermekte hata veriyordu. Hatayı inceledim ve resim yolunu düzenlemem gerektiğini anladım ve hatayı çözdüm.



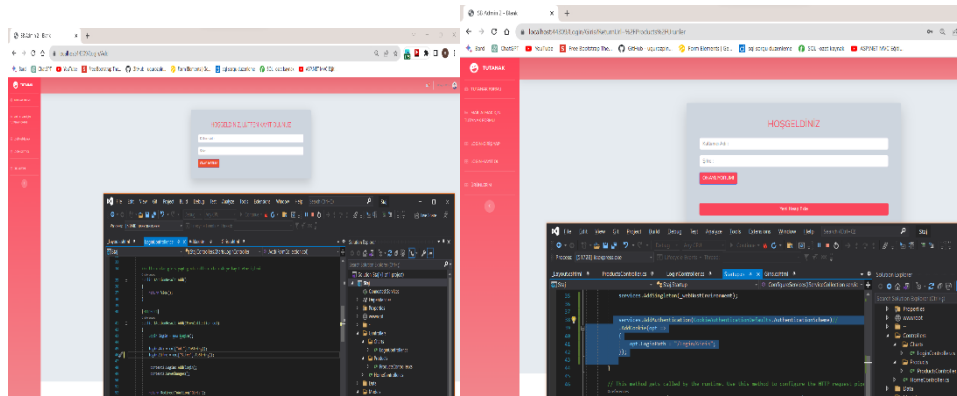
KONTROL SONUCU:

KISIM:

TARİH:04.09.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Login işlemlerinde hem giriş yapmayı hem de yeni kullanıcı için kayıt işlemlerini yaptım. Bunun için LoginController'da Giriş adında bir IActionResults oluşturup View sayfasında daha öncesinde hazırlamış olduğum giriş sayfasını ekledim. Controller bölümünde ise HttpPost bölümünde \_context2 nesnesi yardımıyla eğer Logins tablosunda girilen kullanıcı adı ve şifre ile eşleşen kayıt varsa giriş işleminin başarılı olmasını ve ProductController'da Add sayfasında yönlendirerek kullanıcıya TempData ile bilgi verdim. Eğer kayıt yoksa else'e girecek ve burada ise kullanıcıyı bilgilendirerek tekrar Giriş sayfasına yönlendirdim. Yeni kullanıcının kayıt işlemleri için Add View'de Kullanıcı adı ve şifre bilgilerini alarak bir buton yardımıyla onaylanmasını sağladım ve burada girilen bilgileri LoginController'daki HttpPost'ta IFormCollection ile alarak oluşturduğum Login nesnesinde tuttum. Bu kullanıcı bilgilerini \_context2 ile Logins tablosuna kayıt ettim ve SaveChanges metodu ile değişiklikleri kayıt ettim. Kayıt işleminden sonra Giriş sayfasına yönlendirdim.
- Login sayfalarını oluşturduktan sonra Authentication bağlantısını kurdum. Authentication sistemde giriş yapan kullanıcı var mı yok mu bunu kontrol etmemizi ve sistemde giriş yapan kullanıcı varsa bazı linklerde gezebilsin ama diğerlerinde gezemesin gibi oturum kontrolü yapmamızı sağlar. Öncelikle startup.cs'de services.AddAuthentication ile ekledim ve Login/Giris sayfasına yönlendirmesini tanımladım. Giriş sayfasında Claim'ler oluşturdum. Eğer giriş başarılıysa oluşturduğum Login nesnesi yardımıyla adını ClaimTypes.Name'e ekledim. ClaimsIdentity nesnesi oluşturdum ve principal adında ClaimsIdentity nesnesi oluşturdum. HttpContext.SignInAsync ile kullanıcının oturum açmasını sağladım ve Urunler sayfasına yönlendirerek kullanıcıya mesaj verdim. ProductsControllerda ise [Authorize] ile sayfalara erişim için giriş yapma zorunluluğu getirdim.

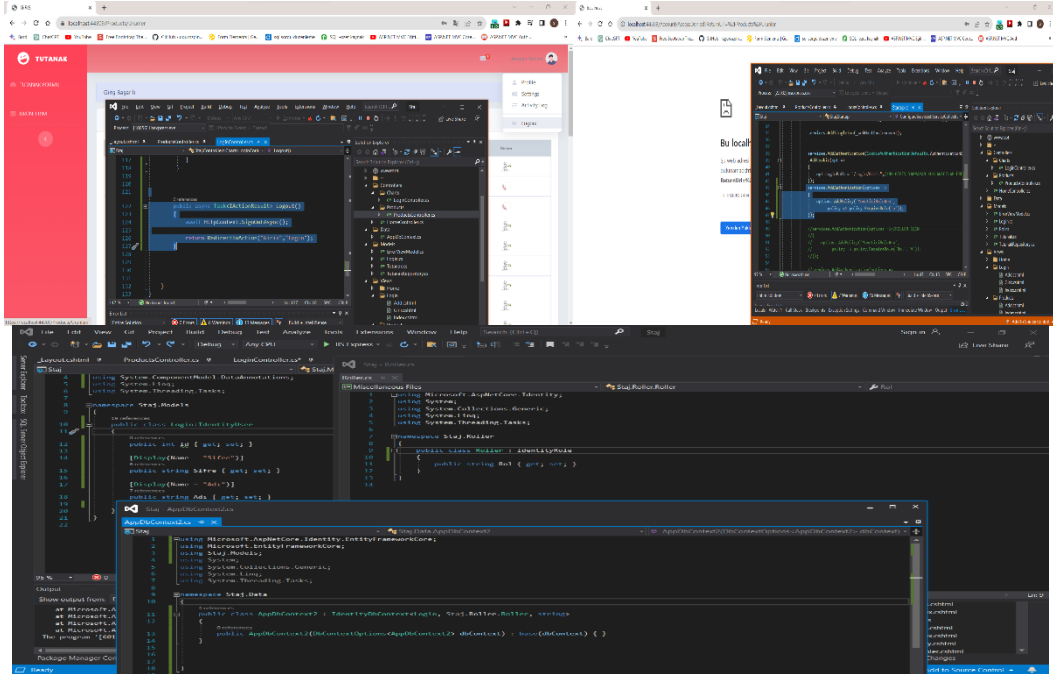


KISIM:

TARİH:05.09.2023

## YAPILAN İŞ: Asp.Net Core MVC5

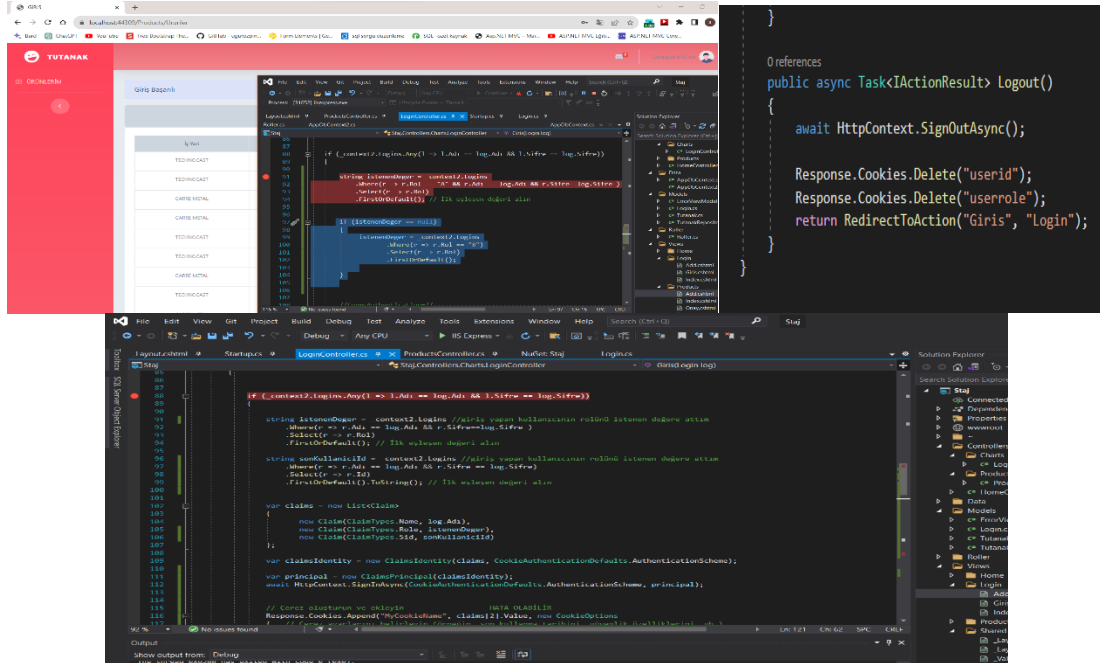
- Çıkış işlemini yaptım. LoginController'da çıkış işlemi için Logout adında yeni bir IActionResult oluşturdum. await HttpContext.SignOutAsync(); ile çıkış işlemi gerçekleştirdim ve Giriş sayfasına yönlendirdim. Logout sayfası oluşturmam bu işlemleri Layout'ta çıkış işlemi için oluşturulan modal'da çıkış butonuna tıkladığında <a class="btn btn-primary" href="@Url.Action("Logout","Login")">Çıkış</a> ile çağırdım.
- Kullanıcılara rol vererek sayfalara erişimi düzenlemeye çalıştım. ProductsController'daki Add sayfasına sadece admin için erişim izni vererek diğer sayfaları tüm kullanıcılara erişim izni vermeye çalıştım. Authorization, kullanıcının belirli bir kaynağa erişimi olup olmadığını belirlemek için kullanılan bir güvenlik mekanizmasıdır. Öncelikle Logins tablosunda kullanıcı rolleri için design diyerek rol alanı oluşturdum ve manuel olarak tanımladım. Startup.cs'de services.AddAuthorization ile AddPolicy'de adını YoneticiRoluAra yaptım ve policy=>policy.RequireRole("a") ile rolü a olanlara erişim izni verdim. LoginController'da rolleri Claim'e ekleyerek roller için modelde ayrı bir class oluşturup ApplicationDbContext bağlantısını da düzenleyerek Identity sınıfını kullanarak bu işlemi yapmayı denedim ancak UserManager sınıfını bulamadı ve Claim'daki roller null geldi. Bu hatayı çözmeye çalıştım.



KONTROL SONUCU:

## YAPILAN İŞ: Asp.Net Core MVC5

- UserManager hatasını çözmeni denedim ancak ilerleme kaydedemedim. Bu yöntemde codefirst ile database bağlantısını kurularak ilerleniyormuş datafirst bağlantısı yaptığım için projede yaptığım değişiklikleri database'e aktaramadım. Farklı bir yol ile kullanıcıların rol bilgisini databaseden almayı denedim. Projede yaptığım değişiklikleri temizledim ve linqery kullanarak ilerledim. Oluşturduğum \_context2 nesnesi ile logins tablosunda giriş yapan kullanıcının bilgilerinde bir kayıt varsa o kullanıcının rolünü aldım ve istenenDeger adında bir değişkene attım ve Claime'e ekledim. Böylece kullanıcıların rollerine ulaşmış oldum ve Authorization işlemini tamamlamış oldum. ProductsController'da Add IActionResults HttpGet'in hemen üzerinde [Authorize(Roles = "A")] tanımladım ve bu sayfaya kullanıvı rolü A olanlara erişim izini vermiş oldum.
- Logins tablosundaki kullanıcı Id'leri ile işlem yapabilmek için Claimleri kullandım. Sisteme son giriş yapan kullanıcının Id'sini alabilmek için linqery kullandım. \_context2 nesnesi yardımıyla giriş yapan kullanıcı adı ve şifresine sahip olan kişinin Id'sini alıp ToString metodu ile string'e çevirip sonKullaniciId değişkeninde tuttum. Bunu Claim'da ClaimTypes.Sid'ye ekledim. Response.Cookies.Append("userid", claims[2].Value); ile çerez oluşturup ekledim. Bu işlemi hem roller hem de adı için yaptım. Oluşturduğum claimleri Logout'ta sildim ve son giriş yapan kullanıcı için hazır hale getirdim.



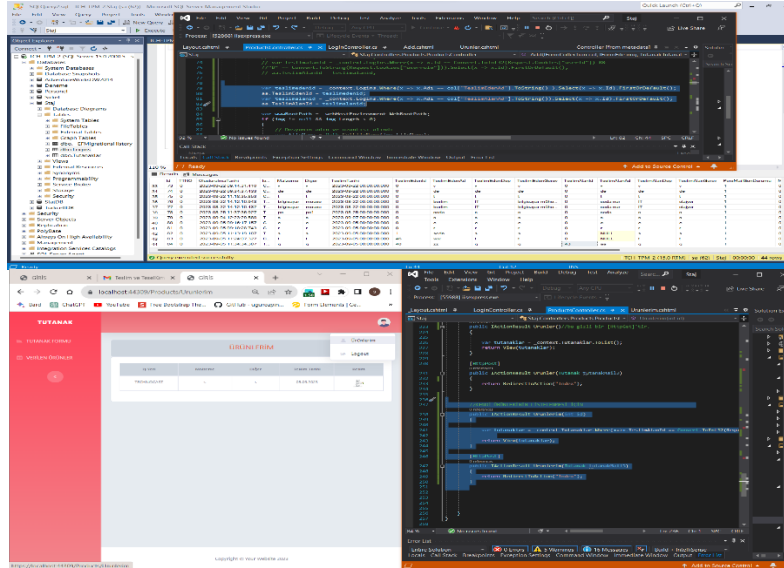
KONTROL SONUCU:

KISIM:

TARİH:07.09.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Teslim eden ve teslim alan kişilerin Id bilgisini linqquery ile adını logins tablosunda bulup o isimdeki kişinin Id'sini bir değişkenlerde tutup bunları ProductsController'ın Add sayfasında database'e kaydettim. Her form için oluşturulan TTNO bilgisini düzenledim. String değişkende TT ile başlatıp yıl bilgisinin son iki rakamını sonrasında ay bilgisini ve tutanak sayısının bir fazlasını yazdırarak bunu database'e kaydettim.
- Her kullanıcıya özel aldığı ürünleri görebilmesi için bir sayfa oluşturdum. Urunlerim sayfasında oluşturduğum view'i Urunler sayfası yaparak yeni işlemleri Urunlerim sayfasında yaptım. Öncelikle Urunler sayfasını düzenledim. Eski sayfaya ek olarak teslim tarihini ve mail onay durumunu da ekledim. Urunlerim sayfasında ise iş yeri, malzeme, diğer, teslim tarihi, ve resim bilgilerini getirdim. Urunlerim sayfasını Layout'taki dropdown kısmında href ile ekledim. Bu sayfayı giriş yapan kişinin Id sine göre dinamik bir hale getirdim. Bunu linqquery sorgusu ile yaptım. Cookilerde tutulan userid ile teslim alan id'si eşitse ToList metodu ile bir değişkende saklayarak return View'e aktardım.
- Onay sayfası için maildeki linke tıklayınca login sayfasına atıyor ve giriş yaptıktan sonra Urunler sayfasına atıyordu. Giriş yapınca onay sayfasına iletmesini sağladım. Begin formda url alınmıyordu, begin form yerine form kullanarak url'yi aldım ve LoginController'ın Giriş View'inde kullandım. Form sayfasında verileri otomatik doldurmak için Login tablosunda departman ve görev bilgilerini manuel ekledim.



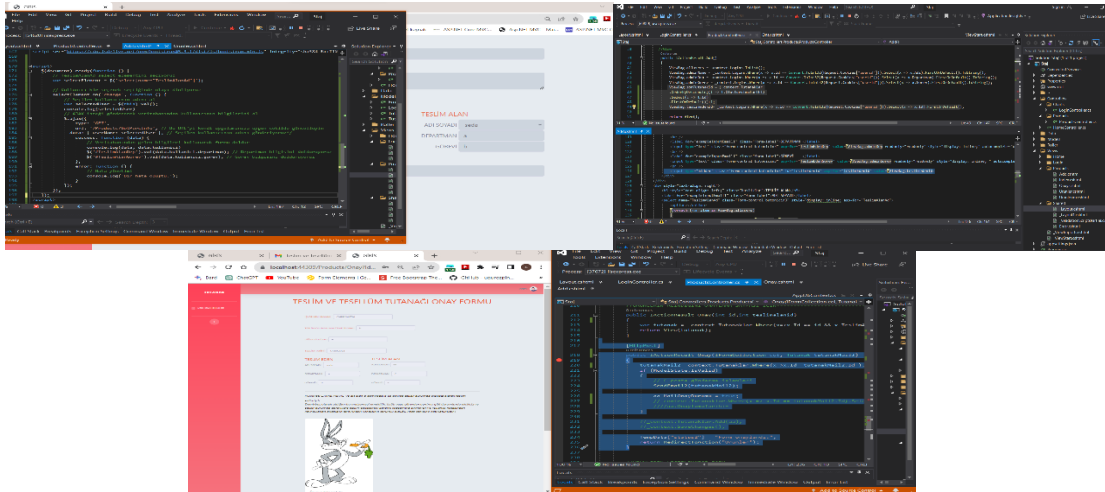
KONTROL SONUCU:

KISIM:

TARİH:08.09.2023

YAPILAN İŞ: Asp.Net Core MVC5

- ProductsController'ın Add View'inde teslim eden bilgilerini otomatik doldurma işlemini yaptım. HttpGet View'inde linquery kullanarak Cookie'lerdeki userid Logins tablosundaki Id'lerden birisiyle eşitse otomatik dolduracağım alanların bilgisini seçtim ve Viewbag'te saklayarak bu bilgileri View sayfasında inputlar içerisinde kullandım. Teslim alan kişi seçildiği zaman o kişinin bilgilerinin herhangi bir butona tıklanmadan otomatik doldurulması istendi. Bu işlem için ajax kullandım. Ajax post etmeden anlık veri değişimini yapmayı sağladığımı öğrendim. Bunun için öncelikle GetPersInfo adında bir IActionResult oluşturdum. string username olarak linquery sorgusu oluşturdum. Username adında bir kullanıcı Logins tablosunda varsa bu kişinin bilgilerini bir değişkende tuttum ve return Json ile ilettim. View sayfasının sonuna ise javascript kodu ekledim. Bu kısımda TeslimAlanAd select elementini seçtim ve değişkene attım. Kullanıcı bir seçenek seçtiğinde oluşturduğum değişken ile olayı dinledim. Ajax isteği göndererek veritabanından kullanıcı bilgilerini aldım ve view sayfasındaki alanları doldurdum.
- ProductsController'da atılan maillerin doğru form ve kişi ile atılmasını Id'leri eşleştirerek sağladım. İlk atılan mailde onay maili için Add sayfasında TeslimAlanId gönderdim. Ajax ile select'den teslim alan seçildiğinde id otomatik geliyor. İlk atılan mailde onay maili için Add sayfasında Id gönderdim. Form onaylandığındaki Id'deki veri için mail attırdım. Teslim alan formu onaylayınca ikinci mail için Id eşleştirerek mail atmasını sağladım. Onay sayfasında label olarak tanımladığım kısımları input yapıp name ve id bilgilerini düzenledim ve bu verileri HttpPost'ta kullandım.



KONTROL SONUCU:

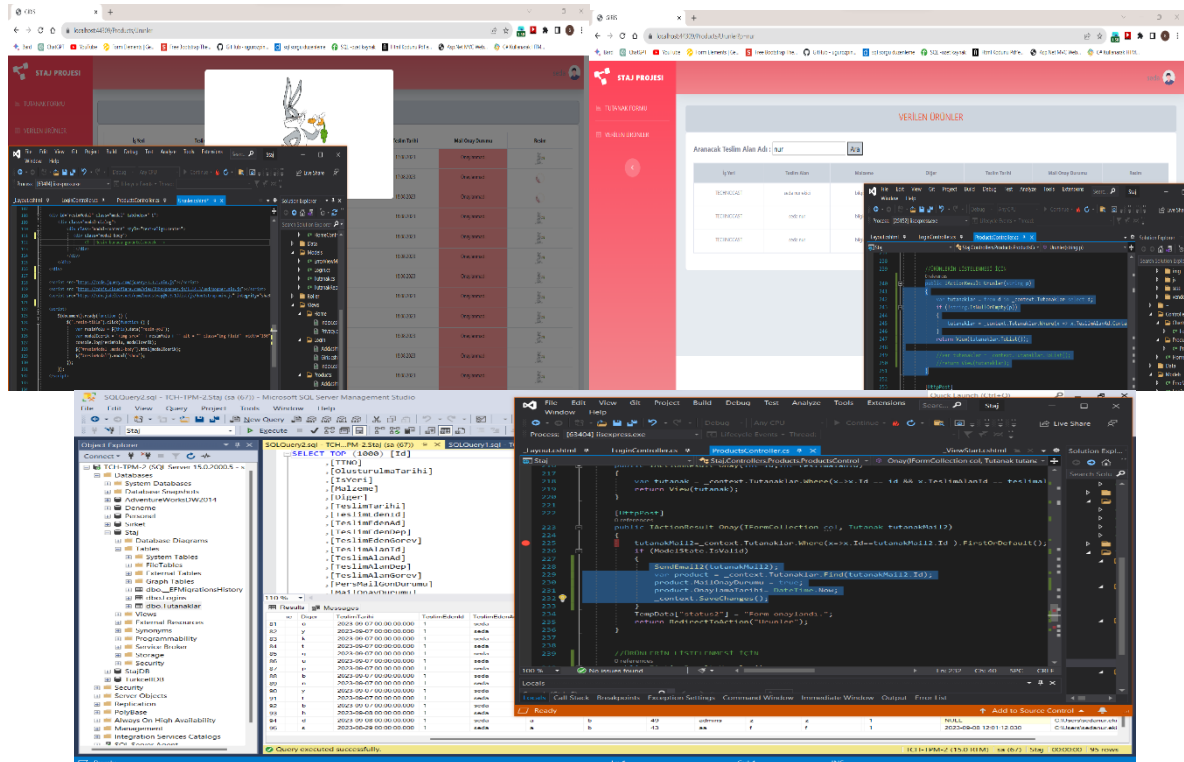


KISIM:

TARİH:11.09.2023

## YAPILAN İŞ: Asp.Net Core MVC5

- Projede Urunler ve Urunlerim sayfasındaki resimlerin üzerine tıklıldığında resmin büyütülmesi işlemini yaptım. View sayfalarında bir modal oluşturarak resmin gösterileceği alanı belirledim. Javascript kodu kullanarak click ile resmin yolunu aldım ve bu yolu düzenleyip View'de oluşturduğum alana ilettim ve tamamlamış oldum. Projede ek istenen bir özellik ile çalışmaya devam ettim. Verilen ürünler sayfasında search yapabilmeyi istediler. Urunler IActionResult'ta bir string nesnesi aldım ve tutanaklardan bu nesneyi seçip değışkende sakladım. Eğer boş değilse tutanaklar = \_context.Tutanaklar.Where(x => x.TeslimAlanAd.Contains(p) || x.Diger.Contains(p)||x.IsYeri.Contains(p)||x.Malzeme.Contains(p)); ile return View'e tutanakları liste halinde gönderdim. Böylece arama yapıldığında aranan kelimeye dair verilerin sayfada görünmesini sağladım.
- ProductsController'da Onay IActionResult'ta mail atıldığında MailOnayDurumu'nu true yaparak database'e kaydettim ve product.OnaylamaTarihi= DateTime.Now; ile onaylanma tarihini database'e kaydettim. Böylece projede yapmam gerekenleri tamamlamış oldum. Proje sunumumu yaparak oluşturduğum sayfaların nasıl kullanılacağını anlattım ve projeyi teslim ettim. Staj süresince beraber çalıştığım ekipte kilerle vedalaştım ve stajımı tamamladım.



KONTROL SONUCU:

Kocaeli Üniversitesi Mühendislik Fakültesi