ASP.NET CORE Tabanlı Fuar Alanı Yönetimi Projesi

1. Yasin Çetin  *Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi*  
191307055  
191307055[@kocaeli.edu.tr](mailto:numara@kocaeli.edu.tr)

2. Sena Gabbani *Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi*  
181307074  
181307074[@kocaeli.edu.tr](mailto:numara@kocaeli.edu.tr)

3. Taha Rıdvan Öztürk *Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi*  
221307097  
221307097@kocaeli.edu.tr

*Özetç:* *MSSQL veritabanı kullanılan backend olarak ASP.NET CORE ve frontend olarak HTML, CSS ve Javascript kullanılarak web, C# ve SQL içerikli bir proje hazırlanmıştır*

Anahtar Kelimeler: Web, Web Development, Veritabanı, ASP.NET CORE, C#, HTML, CSS, Javascipt

# Giriş

Bu proje raporu, bir fuar alanı yönetim projesinin MSSQL veritabanı kullanarak ASP.NET Core backend ve HTML, CSS ve JavaScript teknolojileri kullanılarak oluşturulması hakkında bilgi verecektir.

Bu proje, bir fuar alanının yönetimini kolaylaştırmayı hedeflemektedir. Bu amaçla, fuar alanında yer alan stantların yönetimi, etkinliklerin planlanması ve katılımcıların kayıt altına alınması gibi işlemler otomatize edilmiştir.

MSSQL veritabanı, proje verilerinin tutulduğu merkezi bir depolama alanı sağlamaktadır. ASP.NET Core backend ise, veritabanıyla etkileşim kurarak işlemlerin gerçekleştirilmesini sağlamaktadır. HTML, CSS ve JavaScript teknolojileri ise, proje arayüzünün oluşturulmasında kullanılmıştır.

Proje, kullanıcı dostu bir arayüz ile tasarlanmıştır. Kullanıcılar, stant ve etkinlikleri kolayca görüntüleyebilir, yeni stantlar ve etkinlikler ekleyebilir ve katılımcı kayıtlarını yönetebilirler. Proje aynı zamanda, farklı kullanıcıların farklı yetkilere sahip olmasına izin vererek, güvenli bir şekilde yönetilebilir.

Bu raporda, proje geliştirme süreci, kullanılan teknolojiler ve proje sonuçları ayrıntılı olarak ele alınacaktır.

# Kullanılan teknolojiler

Web geliştirme, temel olarak iki bileşenden oluşur: sunucu tarafı ve istemci tarafı. Sunucu tarafı, web sunucusunda çalışan ve kullanıcı taleplerini işleyen bir yazılım veya uygulama sunucusu tarafından desteklenirken, istemci tarafı ise kullanıcının tarayıcısı üzerinde çalışan HTML, CSS ve JavaScript kodlarından oluşur. İşte, web geliştirmede kullanılan bazı güncel bilgisayar mimarisi teknolojileri ve kullanımları:

## Back-End Teknolojileri

a. ASP.NET Core: ASP.NET Core, Microsoft tarafından geliştirilen, açık kaynaklı ve platform bağımsız bir web uygulama çatısıdır. .NET Core üzerine inşa edilen ASP.NET Core, çoklu platform desteği, hızlı performansı ve ölçeklenebilirliğiyle web geliştirme alanında popüler bir seçenektir.

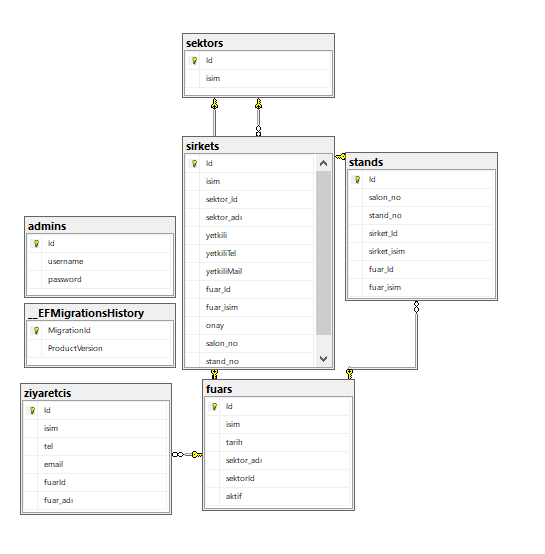
## Front-End Teknolojileri

a. HTML5 ve CSS3: HTML5 ve CSS3, web sayfalarının yapısal ve görsel tasarımını oluşturan temel teknolojilerdir. HTML5, yeni öğeler, API'ler ve multimedya desteği gibi gelişmiş özellikler sunarak web uygulamalarının işlevselliğini artırırken, CSS3, zengin stil ve animasyonlarla daha etkileyici kullanıcı deneyimi sağlar.

b. JavaScript Framework'leri: Angular, React ve Vue.js gibi JavaScript framework'leri, web uygulamalarının karmaşıklığını yönetmek, bileşen tabanlı geliştirme sağlamak ve verimli UI (kullanıcı arayüzü) oluşturmak için kullanılır

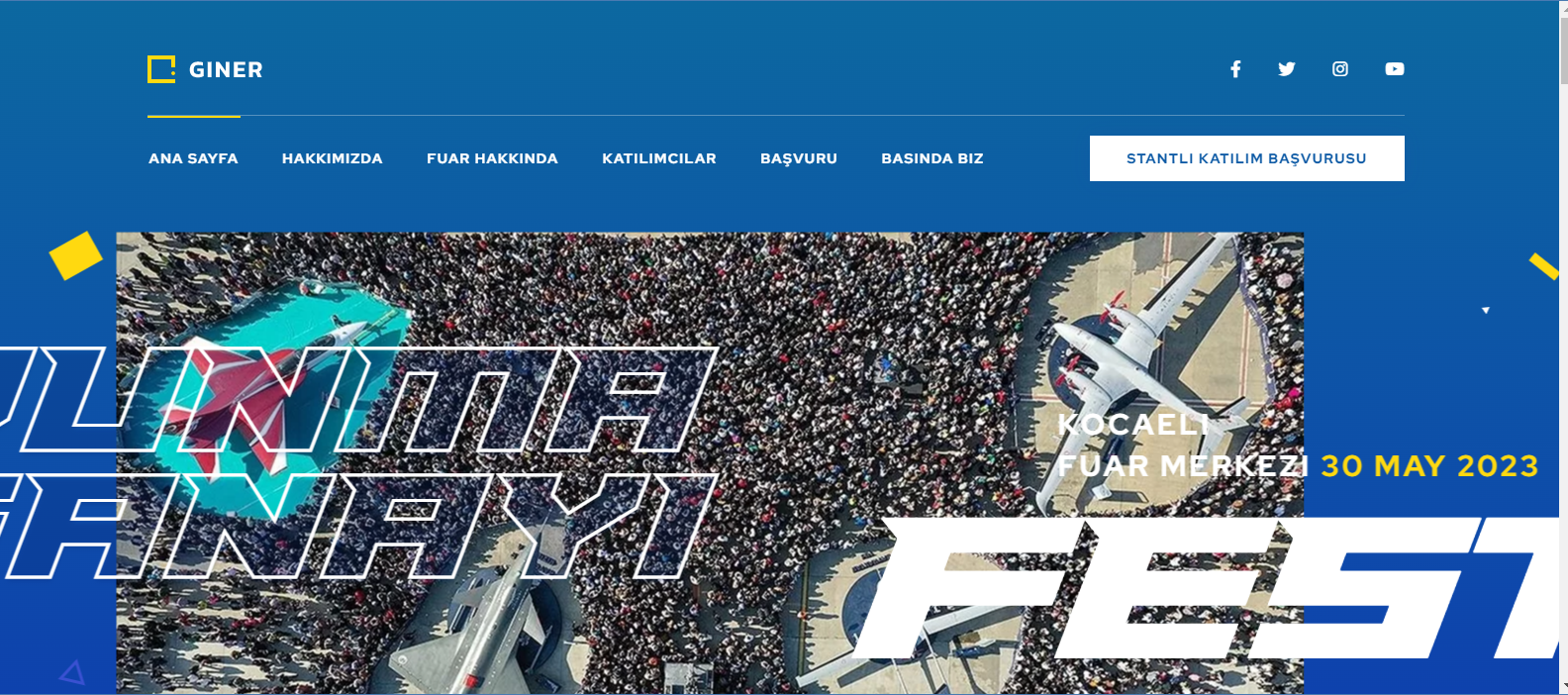
# SQL Veritabanı ve içeriği

Veritabanı adı ihtiyaca yönelik olarak VeritabaniProje şeklinde isimlendirilerek başlanmıştır. Toplantı yapılarak projenin ayrıntıları tartışılmış ve bunun sonucunda “admins”, ”fuars”, ”sektors”, “sirkets”, “stands”, “ziyaretcis” şeklinde table lar oluşturularak ilişkileri eklenmiş ve programa bağlantısı yapılmıştır

Figure 1: Veritabanı diagramı

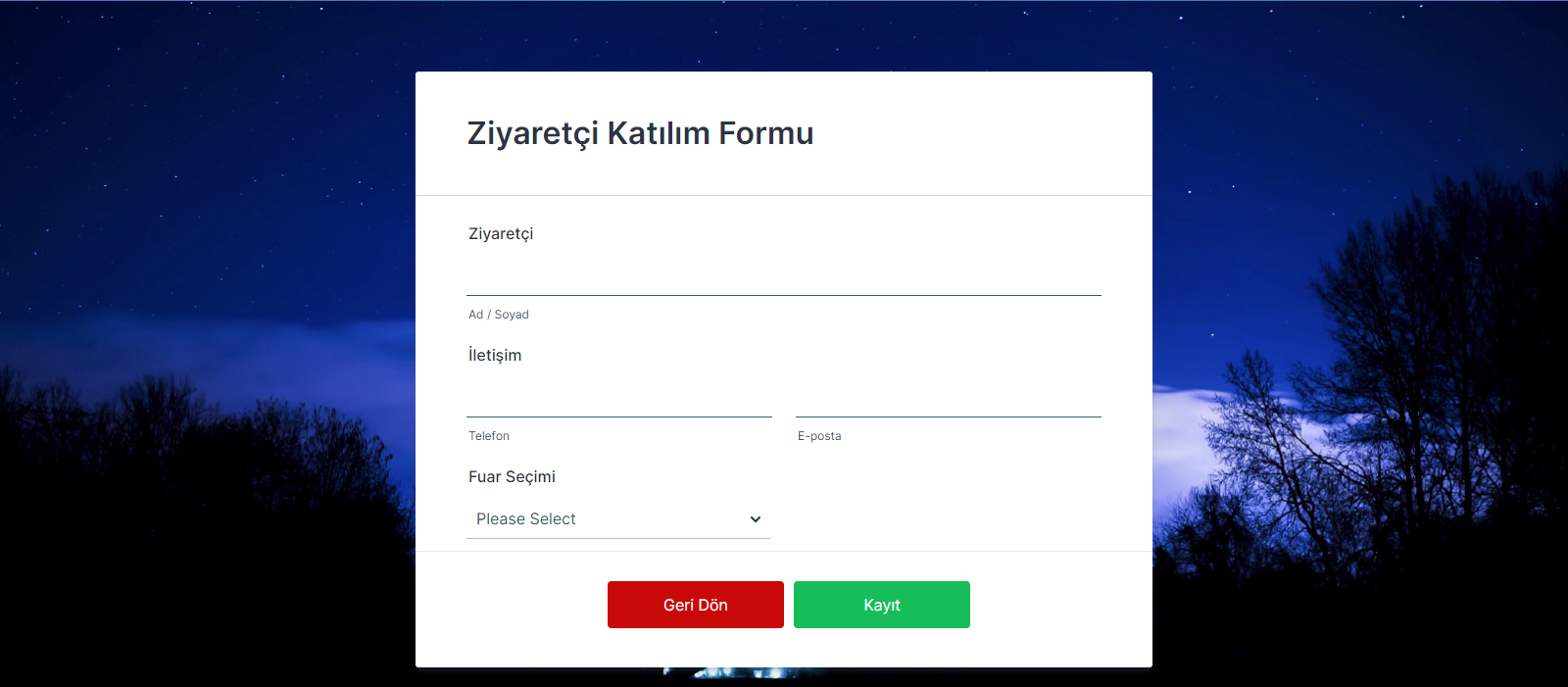
# front-end içeriği ve açıklaması

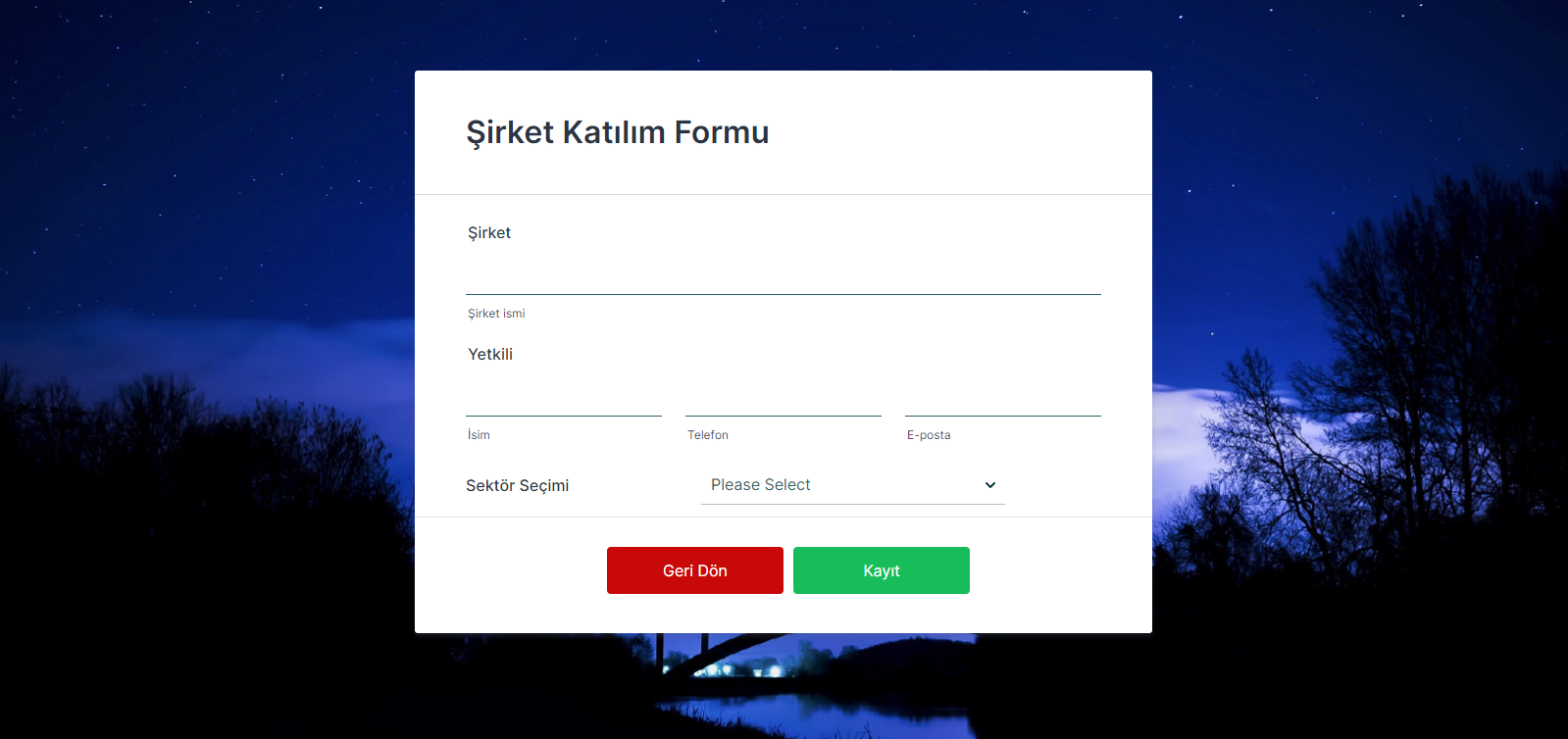
Projenin Web içeriğinin Front-End kısmı farklı template’ler kullanılarak hazırlanmıştır. HTML, CSS ve JS dosyaları Program klasörünün içerisinde “wwwroot” içerisindedir. Bazı sayfalar görsel olarak eklenmiştir.

Figure 2: Giriş Sayfası

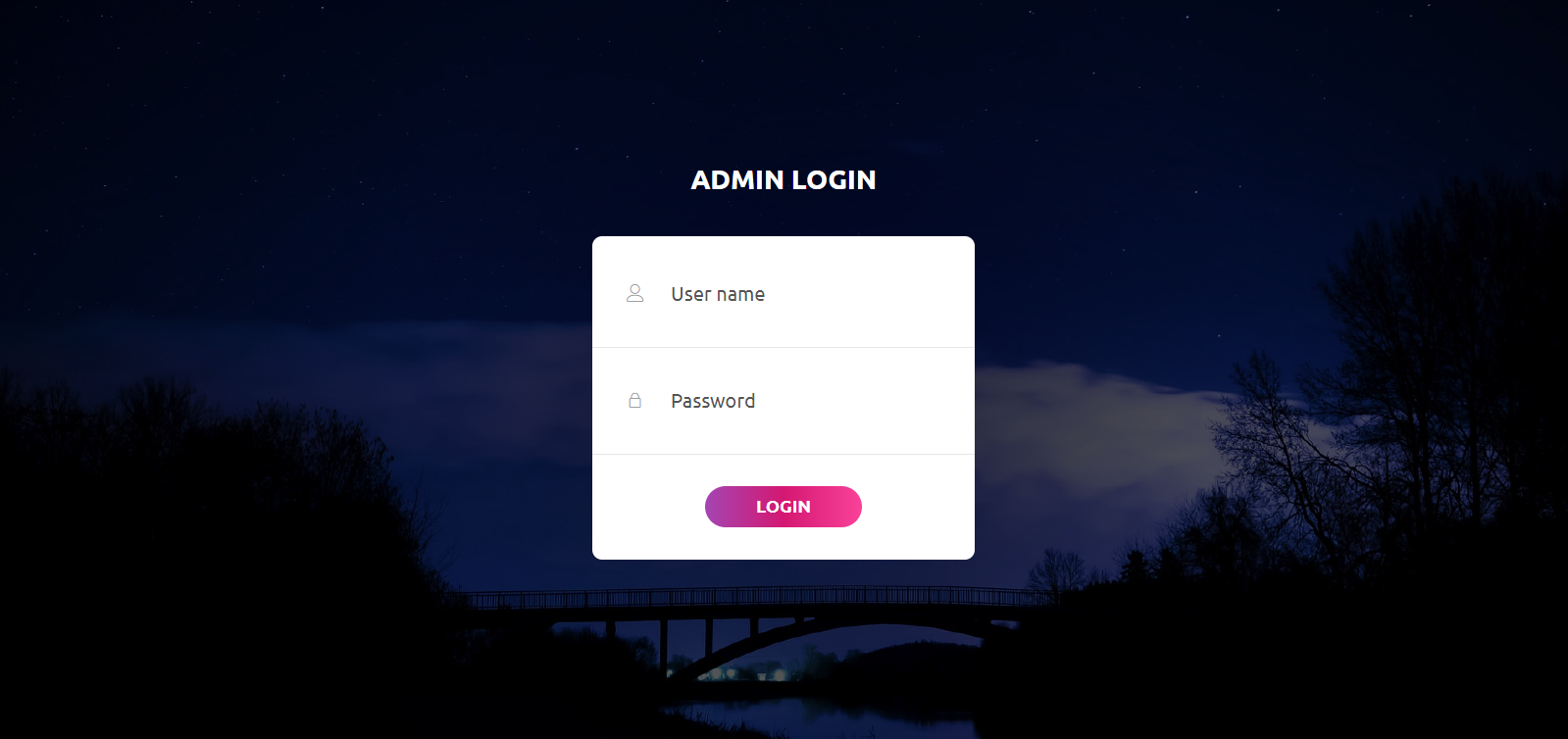
Giriş sayfasında Ana sayfa içeriği, Hakkımızda kısmı, Fuar hakkında bilgilendirme ve Katılımcı şirketler eklenerek gereken genel bilgiler bir “one-page” site halinde verilmiştir.

Ana sayfada “Başvuru” butonuna tıklandığında ziyaretçi kayıt formuna yönlendirilmekte ve burada form üzerinden bilgiler alınarak veritabanında ziyaretcis table’ına girişi yapılmaktadır.

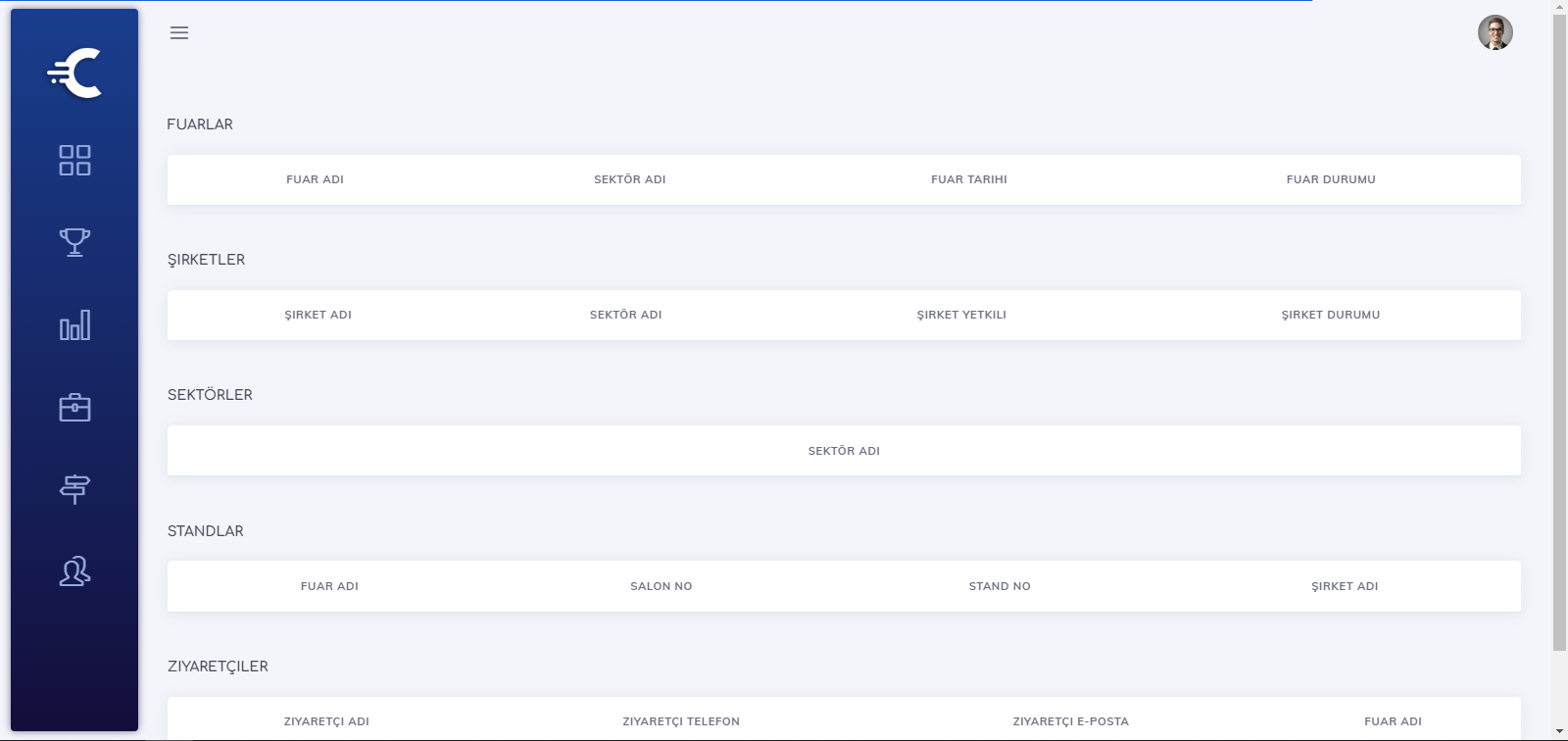
Figure 3: Ziyaretçi Kayıt Formu

Figure 4: Stantlı Katılım Formu

Ana sayfada “Stantlı Katılım Başvurusu” butonuna tıklandığında şirket kayıt formuna yönlendirilmekte ve burada form üzerinden bilgiler alınarak veritabanında sirkets ve sektors table’larına girişi yapılmaktadır.

Figure 5: Admin Giriş Paneli

Admin giriş panelinde önceden veritabanına eklenmiş olan kullanıcılar kullanıcı adı ve şifre ile giriş yaparak sistemde kayıt yapılmış verilere erişebilirler.

Figure 6: Admin Paneli

Admin paneli üzerinde fuarlar, fuarlara katılan şirketler ve tekil ziyaretçiler görüntülenebilmektedir ve isteğe göre filtre yapılabilir şekildedir.

# back-end içeriği ve açıklaması

* Startup.cs

uygulama ayağa kalktığında yapılacak işlemleri belirtir. Ayrıca, veritabanı işlemleri gerçekleştirir.

Kodun içerisindeki ConfigureServices() metodu, servislerin yapılandırılması için kullanılır. Burada, uygulamanın kullanacağı servislerin eklenmesi gerçekleştirilir. AddControllersWithViews() metodu, uygulamanın Controller ve View özelliklerini etkinleştirir. AddMvc() metodu ise, uygulamanın MVC (Model-View-Controller) mimarisini kullanması için gerekli olan servisleri etkinleştirir.

services.AddAuthentication() metodu, kullanıcı kimlik doğrulama servislerini eklemek için kullanılır. Bu kodda, CookieAuthenticationDefaults.AuthenticationScheme kullanılarak cookie tabanlı kimlik doğrulama sağlanmaktadır. LoginPath özelliği, kullanıcıların kimlik doğrulama gerektiren sayfalara erişimlerinde kullanılacak yönlendirme adresini belirtmektedir.

Configure() metodu, uygulamanın HTTP isteklerine nasıl cevap vereceğini belirleyen bir HTTP Pipeline oluşturur. Bu metot, istekleri sırayla işleyen middleware bileşenlerini ekler. UseAuthentication() metodu, kimlik doğrulama özelliğinin kullanılabileceğini belirtir.

HandleExpiredFairs() metodu ise, uygulama ayağa kalktığında çalışacak özel bir metottur. Bu metod, veritabanında tarihleri geçmiş olan fuarların kaydını tutan ve ilgili diğer tablolardaki verileri de silen bir fonksiyondur. Bu fonksiyon, son kullanıcının erişimine açık olmayan bir arka plan işlemi olarak çalıştırılabilir ve veritabanı işlemlerini gerçekleştirerek uygulamanın performansını artırabilir.

* Program.cs

Main() yöntemi, CreateHostBuilder() yöntemini çağırır ve döndürülen IHostBuilder nesnesini kullanarak bir web sunucusu oluşturur ve çalıştırır.

CreateHostBuilder() yöntemi, uygulamanın yapılandırılmasını sağlar. Bu yöntem, bir IHostBuilder nesnesi oluşturur ve bu nesnenin yapılandırılmasını sağlar. Bu nesne, uygulamanın yapılandırılması için bir dizi seçenek sağlar.

CreateDefaultBuilder() yöntemi, bir IHostBuilder nesnesi oluşturur ve varsayılan yapılandırmaları uygular. ConfigureWebHostDefaults() yöntemi, web sunucusu yapılandırmasını sağlar ve Startup sınıfını kullanarak uygulamanın başlangıcını yapılandırır.

* Models Klasörü

Models klasörü içerisinde gerekli sınıflar tanımlanmıştır.

* Migrations Klasörü

first.cs dosyası get\_frequency() adlı bir fonksiyon tanımlar. Bu fonksiyon, bir dizeyi girdi olarak alır ve her karakterin frekansını içeren bir sözlük döndürür. Fonksiyon öncelikle boş bir sözlük olan freq\_dict'i başlatır. Ardından, girdi dizesindeki her karakteri dolaşır ve karakter zaten sözlükte varsa, ilgili değeri 1 artırarak freq\_dict'teki karşılık gelen değeri günceller ya da karakter sözlükte yoksa yeni bir anahtar-değer çifti ekler ve değeri 1 olarak ayarlar. Son olarak, fonksiyon freq\_dict'i döndürür.

Update-sirket.cs dosyası bir sözlük alıp en yüksek değere sahip anahtarı döndüren get\_mode() adlı bir fonksiyon tanımlar. Fonksiyon öncelikle mode\_key ve mode\_value adlı iki değişkeni başlatır. Ardından, girdi sözlüğündeki her anahtar-değer çiftini dolaşır ve mevcut değer mode\_value'dan daha büyükse mode\_key ve mode\_value'yu günceller. Son olarak, fonksiyon mode\_key'i döndürür.

ContextModelSnapshot.cs dosyası ise kullanıcıdan bir dize girdisi alır, get\_frequency() fonksiyonunu girdi dizesiyle birlikte çağırır ve ardından elde edilen sözlükle get\_mode() fonksiyonunu çağırır. Daha sonra, girdi dizisinde en sık geçen karakteri gösteren sonucu yazdırır.

* Controllers Klasörü

Bu klasörde bulunan kod blokları proje üzerindeki kontrolleri sağlar ve veritabanı tarafını da kontrol ederek işlem yapar.

CompanyController.cs dosyası CompanyController adlı bir denetleyici sınıfını tanımlar. Bu denetleyici, Sirket adlı bir modeli kullanarak şirketlerle ilgili işlemleri yönetir.

Index() adlı bir eylem, sektörleri veritabanından alır ve bunları ViewData["Sektorler"] özniteliğine atar. Daha sonra, görünümü döndürür.

GoBack() adlı bir eylem, Stand denetleyicisine bir yönlendirme yapar.

CreateCompany() adlı bir HTTP Post eylemi, kullanıcının bir şirket oluşturmasına izin verir. Bu eylem, bir Sirket nesnesi ve bir sektorId parametresi alır. İlk olarak, veritabanından sektorId parametresine göre sektörü alır ve bir fuar nesnesi ile eşleştirir. Daha sonra, Sirket nesnesinin sektor\_ ve sektor\_adı özniteliklerine sektörü, fuar\_ ve fuar\_isim özniteliklerine de fuarı atar. Son olarak, oluşturulan Sirket nesnesini veritabanına ekler ve değişiklikleri kaydeder. Eylem, Stand denetleyicisindeki Index() eylemine yönlendirme yapar.

HomeController.cs dosyası HTTP isteklerini işlemek için çeşitli işlem yöntemleri olan HomeController adlı bir denetleyici tanımlar.

Index işlemi, ana sayfa için bir görünüm döndürür.

login işlemi, giriş formundan gelen POST isteğini işler ve veritabanındaki girilen kimlik bilgilerini kontrol ederek yönetici kullanıcısını doğrular. Kullanıcı doğrulandığında, işlem, bir iddia kimliği oluşturur ve SignInAsync yöntemini kullanarak kullanıcıyı oturum açar.

LogOut işlemi, kullanıcıyı oturumdan çıkarır ve ana sayfaya yönlendirir.

Dashboard işlemi, [Authorize] özniteliğiyle donatılmıştır, böylece yalnızca doğrulanmış kullanıcılar erişebilir. İşlem, verileri veritabanından alır ve ViewData sözlüğünü kullanarak görünüme aktarır. Çıkış parametresi sorgu dizesinde mevcut ve true değerine sahipse, işlem ana sayfaya yönlendirir.

Fuarlar, Şirketler, Standlar, Ziyaretçiler ve Sektörler işlemleri, verileri veritabanından alır ve ilgili görünümlere aktarır.

StandEkle işlemi, yeni bir standı veritabanına eklemek için formdan gelen POST isteğini işler. İlgili fuar için son eklenen standı alır ve yeni stand için salon ve stand numaralarını hesaplar. Boş bir stand mevcutsa, işlem yeni standı veritabanına ekler ve ilgili şirket kaydını günceller. Aksi takdirde, Sirketler işlemine bir hata mesajı ile yönlendirir.

StandController.cs dosyası

Index eylemi, etkin olan fuarları veritabanından sorgulayarak ve bunları bir liste olarak döndürerek görünümünü (view) oluşturur ve döndürür.

Bu eylem bir asenkron metottur (async Task), bu nedenle await anahtar kelimesiyle veritabanı sorgusunun tamamlanmasını bekler ve ardından sonucu döndürür.

c değişkeni, veritabanına erişim sağlamak için kullanılan Context sınıfından bir nesne oluşturur. Context sınıfı, Entity Framework Core tarafından sağlanan bir sınıftır ve veritabanı işlemlerini gerçekleştirir.

ToList() metodu, sorgudan dönen sonuçları bir liste olarak almak için kullanılır. Bu sayede sorgu sonucu daha sonra kullanılmak üzere bellekte saklanır.

Sonuç olarak, bu kod bloğu, aktif olan fuarları veritabanından alarak Index eylemine gönderir ve görünümü oluşturmak için kullanır.

VisitorController.cs dosyası Index yöntemi, etkin fuarları listelemek için veritabanından veri alır ve bu verileri görünüme ViewData sözlüğü aracılığıyla aktarır.

GoBack yöntemi, Stand/Index eylemine yönlendirir ve bu sayfada geri dönüş işlemini gerçekleştirir.

CreateVisitor yöntemi, ziyaretçi ekleme formu tarafından gönderilen POST isteğini karşılar. Bu yöntem, fuarın kimliğine göre fuarı alır, ziyaretçiyi bu fuara bağlar, ziyaretçiyi veritabanına kaydeder ve Stand denetleyicisinin Index yöntemine yönlendirir.

##### referanslar

* https://themeselection.com/item/cryptodash-free-cryptocurrency-dashboard-template/
* https://themeforest.net/item/giner-multiconcept-event-html-template/24631142
* https://www.jotform.com/tr/form-templates/?classic
* https://learn.microsoft.com/tr-tr/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0
* https://www.w3schools.com/
* https://frontendmasters.com/