



Restoran Sipariş Yönetim Sistemi İçin ASP .NET MVC ve Entity Framework Kullanarak Code First Yaklaşımının Tasarımı ve Uygulaması

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Özdenur ERASLAN BALI

ORCID 0009-0008-5057-5407

Proje Danışmanı: Doç. Dr. Vahide BULUT

Haziran 2024

Restoran Sipariř Yönetim Sistemi İçin ASP .NET MVC ve Entity Framework Kullanarak Code First Yaklaşımının Tasarımı ve Uygulaması

ÖZ

Bu proje, ASP .NET MVC ve Entity Framework kullanarak Code First yaklaşımını benimseyerek bir restoran sipariř yönetim sistemi geliřtirmeyi amaçlamaktadır. Sistemin temel amacı, müşterilerin restoran menüsünden sipariř vermelerine olanak tanımak, bu sipariřleri işlemek ve yönetmek için restoran personeline araçlar sağlamaktır. Web tabanlı bir platform olarak tasarlanacak olan sistem, kullanıcıların kolayca sipariř vermelerine olanak sağlayacak, aynı zamanda restoran personelinin sipariřlerini etkili bir şekilde yönetmesini sağlayacak bir yönetim arayüzü sunacaktır. Projede, MVC mimarisi kullanılarak kullanıcı arayüzü ve iş mantığı ayrı ayrı tasarlanacak ve Entity Framework Code First yaklaşımı ile veritabanı modellemesi yapılacaktır. Bu proje, modern web teknolojilerinin kullanımını ve veri yönetimi becerilerini sergileyecek, aynı zamanda restoran işletmeciliği için kullanışlı bir çözüm sunacaktır.

Anahtar Sözcükler: ASP .NET, Entity Framework, Code First, MVC, SQL, Web

Design and Implementation of Code First Approach Using ASP .NET MVC and Entity Framework for Restaurant Order Management System

Abstract

This project aims to develop a restaurant order management system by adopting the Code First approach using ASP .NET MVC and Entity Framework. The main purpose of the system is to allow customers to place orders from the restaurant menu and provide restaurant staff with tools to process and manage these orders. The system, which will be designed as a web-based platform, will allow users to easily place, and will also provide a management interface that will enable restaurant staff to manage their orders effectively. In the project, the user interface and business logic will be designed separately using MVC architecture and database modeling will be done with the Entity Framework Code First approach. This project will showcase the use of modern web technologies and data management capabilities, as well as provide a useful solution for restaurant business.

Keywords: ASP .NET, Entity Framework, Code First, MVC, SQL, Web

*Çalıřma sürecinde hiçbir zaman desteęini esirgemeyen sevgili eřime sonsuz
teřekkürlerimi sunarım.*

Teşekkür

Proje çalışmasına katkılarından dolayı sevgili danışman hocam Doç. Dr. Vahide BULUT'a sonsuz teşekkürlerimi sunarım.

İçindekiler

Öz	i
Abstract	ii
Teşekkür	iv
Şekiller Listesi.....	vi
Kısaltmalar Listesi	vii
Semboller Listesi.....	viii
1 Giriş	1
2 Proje İle İlgili Genel Bilgiler ve Kullanılan Teknolojiler	3
2.1 Projenin Amacı ve Kapsamı	3
2.2 ASP .NET MVC Teknolojisi	5
2.3 Entity Framework ve Code First Yaklaşımı	6
2.4 SQL Veritabanı	7
2.5 Geliştirme Aşamasında Kullanılan Diller	8
3 Projenin Tasarımı ve Uygulaması.....	9
3.1 Veritabanı Modelleme	9
3.2 Kullanıcı Arayüzü Tasarımı.....	11
3.3 Sipariş Yönetim Sistemi	12
3.4 Yönetim Arayüzü.....	13
4 Projenin Değerlendirilmesi.....	14
Kaynaklar	25

Şekiller Listesi

Şekil 3.1	MSSQL Veritabanı Diyagramı.....	10
Şekil 3.2	Projede kullanılan DbContext Sınıfı	10
Şekil 3.3	Proje ile MSSQL arasında bağlantıyı sağlayan bağlantı cümlesi	11
Şekil 4.1	Uygulama ana sayfası.....	15
Şekil 4.2	Uygulama hakkımızda sayfası	15
Şekil 4.3	Üyelik oluşturma sayfası.....	16
Şekil 4.4	Üye giriş sayfası	16
Şekil 4.5	Müşteri ana sayfası.....	17
Şekil 4.6	Müşteri sipariş ana sayfası	17
Şekil 4.7	Müşteri sipariş oluşturma sayfası.....	18
Şekil 4.8	Müşteri sipariş onay sayfası	18
Şekil 4.9	Müşteri sipariş ödeme bilgisi sayfası	19
Şekil 4.10	Müşteri siparişlerini görüntüleme sayfası	19
Şekil 4.11	Yönetici giriş sayfası.....	20
Şekil 4.12	Yönetici paneli sayfası	20
Şekil 4.13	Yönetici menü listeleme, düzenleme ve silme sayfası.....	21
Şekil 4.14	Yönetici paneli menü ekleme sayfası.....	21
Şekil 4.15	Yönetici paneli malzeme listeleme, düzenleme ve silme sayfası.....	22
Şekil 4.16	Yönetici paneli malzeme ekleme sayfası	22
Şekil 4.17	Yönetici paneli en çok sipariş verilen malzemeleri görüntüleme sayfası	23
Şekil 4.18	Yönetici paneli en çok sipariş verilen menüleri görüntüleme sayfası.....	23
Şekil 4.19	Yönetici paneli sipariş verilen tüm menüleri görüntüleme sayfası	24
Şekil 4.20	Yönetici paneli kayıtlı kullanıcıları görüntüleme sayfası	24

Kısaltmalar Listesi

ASP	Active Server Pages
.NET	.NET Framework (Microsoft'un yazılım geliştirme çerçevesi)
MVC	Model-View-Controller
C#	C Sharp
EF	Entity Framework
ORM	Object Relational Mapping (Nesne İlişkisel Eşleme)
CF	Code First
SQL	Structured Query Language
RDBMS	Relational Database Management System (İlişkisel Veritabanı Yönetim Sistemi)
MSSQL	Microsoft SQL Server
CRUD	Create, Read, Update, Delete
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
URL	Uniform Resource Locator
JS	JavaScript

Semboller Listesi

Ön işlemci direktifi

Bölüm 1

Giriş

Günümüzün hızlı ve rekabetçi iş dünyasında, restoranlar müşteri memnuniyetini artırmak ve verimliliği artırmak için teknolojiyi etkin bir şekilde kullanmalıdır. Restoran işletmeciliğinde başarı, sadece lezzetli yemekler sunmakla değil; aynı zamanda hızlı ve hatasız sipariş alma, etkili sipariş yönetimi ve müşteri beklentilerini karşılayacak şekilde hizmet sunma gibi unsurlar da büyük önem taşır. Bu bağlamda, restoranların dijital dönüşüm sürecine adapte olarak iş süreçlerini optimize etmeleri gerekmektedir.

Bu proje, ASP .NET MVC ve Entity Framework kullanarak geliştirilecek bir restoran sipariş yönetim sistemi tasarımı ve uygulamasını kapsamaktadır. Projede, Code First yaklaşımı kullanılarak veri tabanı modellemesi yapılacak ve modern web teknolojileri kullanılarak kullanıcı dostu bir arayüz sağlanacaktır. Sistem, müşterilerin restoran menüsünden sipariş vermelerini kolaylaştıracak, siparişlerini görmelerine olanak tanıyacak ve restoran personelinin bu siparişleri etkin bir şekilde yönetmesini sağlayacak araçlar sunacaktır.

Projenin amacı, restoranların dijital altyapılarını güçlendirerek operasyonel verimliliği artırmak ve müşteri deneyimini iyileştirmektir. Bu doğrultuda, MVC (Model-View-Controller) mimarisi kullanılarak sistemin kullanıcı arayüzü ile iş mantığı katmanları birbirinden ayrılacak, böylece hem geliştirme süreci kolaylaşacak hem de sistemin sürdürülebilirliği sağlanacaktır. Entity Framework'ün Code First yaklaşımı ile veri tabanı modellemesi yapılacak, performansın artırılması için uygun veri tabanı indeksleme ve sorgu optimizasyonu teknikleri uygulanacaktır.

Bu alıřma, modern web teknolojilerinin ve veri ynetim becerilerinin pratik bir uygulaması olarak restoran iřletmecilięi iin etkin bir zm sunmayı hedeflemektedir.

Bölüm 2

Proje İle İlgili Genel Bilgiler ve Kullanılan Teknolojiler

Bu projeyi geliştirmek için günümüzün en yaygın ve etkili web teknolojileri kullanılmıştır. Proje, Microsoft Visual Studio ortamında geliştirilmiştir. Projenin ana teknolojik bileşenleri ASP .NET MVC, Entity Framework, Code First yaklaşımı ve SQL veritabanıdır. Bu teknolojiler, modern web uygulamaları geliştirmede sağladıkları esneklik ve güçlü özelliklerle öne çıkmaktadır.

2.1 Projenin Amacı ve Kapsamı

Bu projenin temel amacı, restoranların sipariş yönetim süreçlerini dijitalleştirerek daha verimli ve kullanıcı dostu bir sistem sunmaktır. Müşterilerin restoran menüsünden kolayca sipariş verebilmelerini sağlamak, siparişlerini görebilmelerini mümkün kılmak ve restoran personelinin bu siparişleri etkin bir şekilde yönetmesine olanak tanımak bu projenin ana hedefleridir. Projenin spesifik amaçları şu şekilde özetlenebilir:

Müşteri Deneyimini İyileştirmek: Müşterilerin menüye kolay erişim sağlayarak sipariş verme sürecini hızlandırmaktır.

Operasyonel Verimliliği Artırmak: Restoran personelinin siparişleri daha hızlı ve hatasız bir şekilde yönetmelerini sağlamaktır.

Dijital Dönüşümü Desteklemek: Restoranların dijitalleşme sürecine katkıda bulunarak teknolojik altyapılarını güçlendirmek ve rekabet avantajı kazanmalarına yardımcı olmaktır.

Veri Yönetimini Geliştirmek: Sipariş verilerini etkili bir şekilde yöneterek analiz ve raporlama imkanları sunmaktır.

Proje, restoran sipariş yönetim sisteminin geliştirilmesi ve uygulanmasını kapsamaktadır. Proje kapsamında gerçekleştirilecek başlıca faaliyetler ve bileşenler aşağıda belirtilmiştir:

→Sistem Analizi ve Tasarımı:

- Mevcut sipariş yönetim süreçlerinin analizi.
- Gereksinimlerin belirlenmesi ve sistem mimarisinin tasarımı.

→Veritabanı Modelleme:

- Entity Framework kullanılarak veri modellerinin oluşturulması.
- Code First yaklaşımı ile veritabanı yapısının inşa edilmesi.

→Kullanıcı Arayüzü Geliştirme:

- ASP .NET MVC framework'ü kullanılarak kullanıcı arayüzlerinin tasarımı ve geliştirilmesi.
- HTML, CSS ve JavaScript kullanılarak dinamik ve duyarlı web sayfalarının oluşturulması.

→Sipariş Yönetim Modülü:

- Müşterilerin menüden sipariş vermelerini sağlayacak modülün geliştirilmesi.
- Siparişlerin görüntüleneceği müşteri arayüzünün oluşturulması.
- Restoran personeli için sipariş yönetim araçlarının geliştirilmesi.

Projenin sonunda, kullanıcı dostu ve yüksek performanslı bir restoran sipariş yönetim sistemi oluşturulmuş olacaktır.

2.2 ASP .NET MVC Teknolojisi

ASP.NET, olay yönlendirmeli web uygulamaları geliştirilmesini sağlayan .NET platformunun web uygulama geliştirme teknolojisidir (Turan, 2018). Microsoft tarafından geliştirilmiş, Model-View-Controller tasarım desenini temel alır. MVC, karmaşık web uygulamalarını daha yönetilebilir ve test edilebilir hale getiren bir yazılım mimarisidir.

ASP.NET MVC uygulaması üç ana bileşenden oluşur:

- Model: Uygulamanın veri ve iş mantığı katmanıdır. Veritabanı işlemleri, veri doğrulama, iş kuralları ve diğer veri ile ilgili işlemler bu katmanda gerçekleştirilir.
- View: Kullanıcı arayüzünü temsil eder. Kullanıcının gördüğü ve etkileşime geçtiği tüm öğeler bu katmanda bulunur. View, modelden gelen veriyi kullanıcıya sunar ve kullanıcıdan gelen girdileri alır. ASP .NET MVC'de View'ler, HTML, CSS ve JavaScript kullanılarak oluşturulur ve dinamik hale getirilir.
- Controller: Kullanıcı etkileşimlerini yönetir ve model ile view arasındaki iletişimi sağlar. Bir kullanıcının bir sayfaya istek göndermesi, bir formu doldurması veya bir düğmeye tıklaması gibi eylemler Controller tarafından işlenir. Controller, bu eylemleri alır, gerekli iş mantığını uygular ve uygun bir view'i döndürerek yanıt verir.

Bir kullanıcı, bir ASP.NET MVC uygulamasına istek gönderdiğinde, istek önce bir Controller'a gönderilir. Controller, isteği işler ve Model'e erişerek gerekli verileri alır. Daha sonra Controller, verileri bir View'a aktarır ve kullanıcıya sunulan HTML kodunu oluşturur. (<https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>)

ASP.NET MVC, çeşitli web uygulamaları geliştirmek için kullanılabilir, şunlar dahildir:

- E-ticaret web siteleri: Ürün katalogları, alışveriş sepetleri ve ödeme işlemleri gibi e-ticaret işlevlerini yönetmek için kullanılabilir.

- İerik ynetim sistemleri: Bloglar, makaleler ve dięer web ierięini ynetmek iin kullanılabilir.
- Web uygulamaları: Mřteri iliřkileri ynetimi, proje ynetimi ve iř zekası gibi web tabanlı uygulamaları geliřtirmek iin kullanılabilir.

ASP.NET MVC'nin faydaları ise ařaęıda verilmiřtir:

İyi organize edilmiř kod: MVC, uygulamayı modele, view ve controller'a ayırarak kodu daha organize hale getirir. Bu, kodun daha kolay anlaşılmasını, bakımı yapılmasını ve test edilmesini saęlar.

Test edilebilirlik: MVC, birim testleri yazmayı ve uygulamanın farklı bileřenlerini izole ederek test etmeyi kolaylařtırır.

Tekrar kullanılabilirlik: MVC, model, view ve controller bileřenlerini yeniden kullanmayı kolaylařtırarak geliřtirici verimlilięini artırır.

URL ynlendirme: ASP.NET MVC, esnek bir URL ynlendirme sistemi sunar.

Gvenlik: ASP.NET MVC, kimlik doęrulama ve yetkilendirme gibi geliřmiř gvenlik zellikleri ierir.

2.3 Entity Framework ve Code First Yaklařımı

Entity Framework (EF), Microsoft tarafından geliřtirilen ve .NET uygulamaları iin popler bir Object-Relational Mapping (ORM) aracıdır. ORM (Object Relational Mapping), en temel tanımıyla veri tabanı ile nesneye ynelik programlama (OOP) arasındaki iliřkiyi kuran teknolojidir (Balcı, 2018). EF, geliřtiricilere veritabanı iřlemlerini doęrudan C# gibi nesne tabanlı programlama dilleriyle gerekleřtirme olanaęı tanır. Bu sayede SQL sorguları yazmadan, veritabanı tablolarıyla etkileřime gemek mmkndr. (<https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>) Entity Framework'n temel bileřenleri bulunmaktadır:

- DbContext: Veritabanı baęlantısını ve veritabanı iřlemlerini yneten ana bileřendir. Veritabanı tabloları ile nesnelere arasında kpr grevi grr.

- DbSet: Belirli bir tabloyu temsil eder ve bu tablo üzerindeki CRUD (Create, Read, Update, Delete) işlemlerini yönetir.
- Entity: Veritabanı tablolarını temsil eden sınıflardır. Bu sınıflar, veritabanı tablolarının sütunlarına karşılık gelen özellikler içerir.

Entity Framework (EF), çeşitli yaklaşımlar sunarak farklı geliştirme senaryolarına uygun esneklik sağlar. Bu yaklaşımlar, veri modeli ve veritabanı arasındaki ilişkiyi nasıl yöneteceğimizi belirler. (<https://coskunkurtuldu.medium.com/net-core-entity-framework-code-first-yakla%C5%9F%C4%B1m%C4%B1-6ac02867b3e5>) EF'nin üç temel yaklaşımı şunlardır:

- Code First Yaklaşımı
- Data First Yaklaşımı
- Model First Yaklaşımı

Proje kapsamında Code First yaklaşımı kullanılmıştır. Code First yaklaşımı, EF'de veritabanı şemasının uygulama kodundan oluşturulmasını sağlayan yöntemdir. Bu yaklaşımda, öncelikle veri tabanını temsil eden nesne sınıfları tanımlanır. Ardından EF, bu nesne sınıflarını kullanarak veritabanı şemasını otomatik olarak oluşturur veya mevcut bir veritabanı şemasını günceller. (<https://medium.com/swlh/entity-framework-core-with-the-code-first-approach-in-net-core-project-f2065b2c28d5>)

2.4 SQL Veritabanı

SQL (Structured Query Language), ilişkisel veritabanlarını yönetmek için kullanılan bir dildir. (https://simple.wikipedia.org/wiki/Structured_Query_Language) Verileri eklemek, sorgulamak, güncellemek ve silmek için kullanılır. SQL, günümüzde en yaygın kullanılan veritabanı dillerinden biridir ve birçok farklı veritabanı sistemi tarafından desteklenmektedir.

MSSQL (Microsoft SQL Server), Microsoft tarafından geliştirilmiş bir ilişkisel veritabanı yönetim sistemidir (Altıntaş, 2019). İşletmeler tarafından büyük ve karmaşık verileri depolamak ve yönetmek için yaygın olarak kullanılır.

2.5 Geliştirme Aşamasında Kullanılan Diller

Proje, Microsoft Visual Studio ortamında .NET 8.0 Framework üzerine inşa edilmiş, ASP.NET Core MVC mimarisi kullanılarak geliştirilmiş bir web uygulamasıdır. Bu uygulama, Model-View-Controller (MVC) tasarım desenini benimseyerek, kullanıcı arayüzü, iş mantığı ve veri erişimini katmanlı bir yapı ile birbirinden ayırır. Bu sayede, uygulamanın bakımı, genişletilebilirliği ve test edilebilirliği artırılmıştır.

Projenin geliştirilme aşamasında C# programlama dili kullanılarak ASP.NET Core framework'ü üzerinde bir taban oluşturulmuştur. Kullanıcı arayüzü için HTML ve CSS dilleri kullanılarak statik ve dinamik içeriklerin oluşturulması sağlanmıştır. Ayrıca, kullanıcı etkileşimini artırmak ve sayfalar arası dinamik içerik akışını sağlamak amacıyla JavaScript dili kullanılmıştır. Bu diller, hem sunucu tarafında hem de istemci tarafında işlevselliği artırmak ve kullanıcı deneyimini artırmak için bir arada kullanılmıştır.

Bölüm 3

Projenin Tasarımı ve Uygulaması

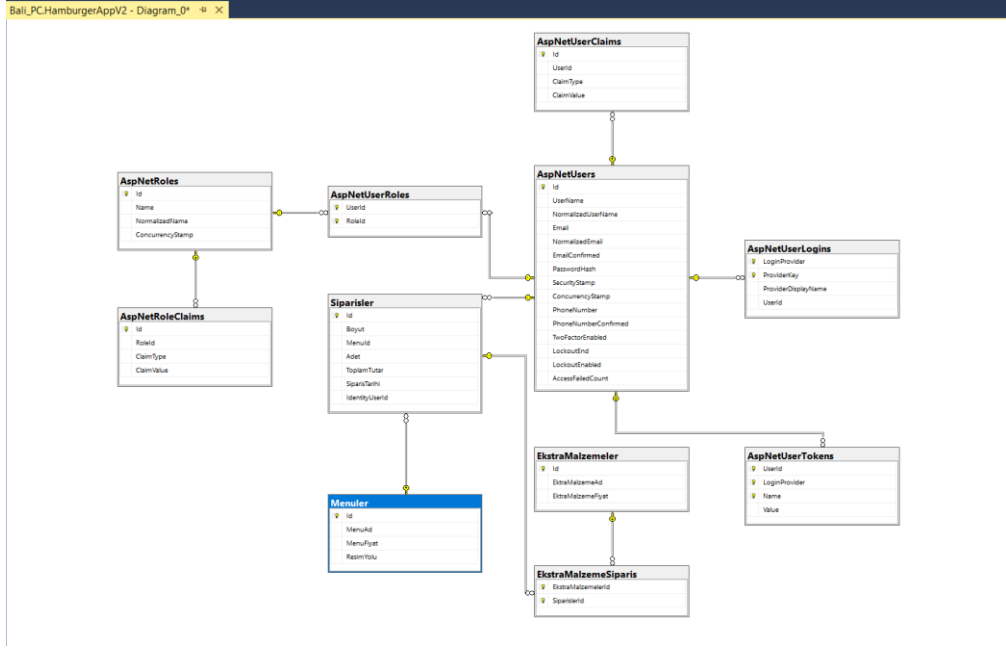
Projenin tasarımı aşamasında, kullanıcı gereksinimleri analiz edilmiş ve bu gereksinimler doğrultusunda bir tasarım planı oluşturulmuştur. Kullanıcı arayüzü için gerekli olan sayfaların ve işlevlerin belirlenmesi, veri modelinin oluşturulması ve veritabanı şemasının tasarlanması gibi adımlar gerçekleştirilmiştir.

Uygulama aşamasında, tasarım planı temel alınarak proje kodlanmış ve geliştirilmiştir. ASP.NET MVC mimarisi kullanılarak, Controller sınıfları oluşturulmuş ve iş mantığı bu sınıflar içinde kodlanmıştır. Model sınıfları, veritabanı ile etkileşim sağlayacak şekilde tasarlanmış ve Entity Framework kullanılarak veri erişimi sağlanmıştır.

Kullanıcı arayüzü, HTML ve CSS kullanılarak statik olarak oluşturulmuş ve JavaScript dili kullanılarak dinamik hale getirilmiştir. Kullanıcıların web sitesinde gezinmesini sağlayan ve kullanıcı etkileşimlerini yöneten JavaScript fonksiyonları yazılmıştır. Son olarak, proje test edilmiş ve kullanıma hazır hale getirilmiştir.

3.1 Veritabanı Modelleme

Projede, Microsoft Visual Studio kullanılarak Entity Framework Code First yaklaşımıyla veritabanı modellemesi yapılmıştır. Bu yaklaşım, veritabanı şemasının C# sınıflarıyla kodlanarak oluşturulmasını sağlar. Bu sayede, veritabanı modeli kod tabanında tanımlanır ve veritabanı oluşturma ve güncelleme işlemleri Entity Framework tarafından yönetilir. Entity Framework, uygulama başlatıldığında, bu DbContext sınıfıyla ilişkilendirilir ve veritabanı bağlantısını otomatik olarak yapar. Veritabanı şeması, bu C# sınıflarına ve ilişkilerine göre otomatik olarak oluşturulur veya güncellenir. Bu şekilde, proje geliştirme sürecinde veritabanı modellemesi kod tabanında yapılarak, daha esnek ve yönetilebilir bir yapı elde edilir.



Şekil 3.1: MSSQL Veritabanı Diyagramı

```

namespace HamburgerAppV1.Data
{
    24 references
    public class ApplicationDbContext : IdentityDbContext
    {
        0 references
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }

        18 references
        public DbSet<Menu> Menuler { get; set; }

        9 references
        public DbSet<EkstraMalzeme> EkstraMalzemeler { get; set; }

        11 references
        public DbSet<Siparis> Siparisler { get; set; }

        0 references
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<EkstraMalzeme>()
                .Property(em => em.EkstraMalzemeFiyat)
                .HasColumnType("decimal(18, 2)");

            modelBuilder.Entity<Menu>()
                .Property(m => m.MenuFiyat)
                .HasColumnType("decimal(18, 2)");

            modelBuilder.Entity<Siparis>()
                .Property(s => s.ToplamTutar)
                .HasColumnType("decimal(18, 2)");

            modelBuilder.Entity<Menu>().HasData(
                new Menu() { Id = 1, MenuAd = "Cheeseburger Menü", MenuFiyat = 100, ResimYolu = "cheeseburger-menu.png" },
                new Menu() { Id = 2, MenuAd = "BigKing Menü", MenuFiyat = 120, ResimYolu = "big-king-menu.png" },
                new Menu() { Id = 3, MenuAd = "King Chicken Menü", MenuFiyat = 110, ResimYolu = "king-chicken-menu.png" },
                new Menu() { Id = 4, MenuAd = "Whopper Menü", MenuFiyat = 115, ResimYolu = "whopper-menu.png" },
                new Menu() { Id = 5, MenuAd = "Köfteburger Menü", MenuFiyat = 90, ResimYolu = "kofteburger-menu.png" }
            );

            modelBuilder.Entity<EkstraMalzeme>().HasData(
                new EkstraMalzeme() { Id = 1, EkstraMalzemeAd = "Metçap", EkstraMalzemeFiyat = 2 },
                new EkstraMalzeme() { Id = 2, EkstraMalzemeAd = "Mayonez", EkstraMalzemeFiyat = 2 },
                new EkstraMalzeme() { Id = 3, EkstraMalzemeAd = "Ranch Sos", EkstraMalzemeFiyat = 4 },
                new EkstraMalzeme() { Id = 4, EkstraMalzemeAd = "Buffalo Sos", EkstraMalzemeFiyat = 4 },
                new EkstraMalzeme() { Id = 5, EkstraMalzemeAd = "Barbekü Sos", EkstraMalzemeFiyat = 4 }
            );
        }

        base.OnModelCreating(modelBuilder);
    }
}

```

Şekil 3.2: Projede kullanılan DbContext Sınıfı

```
"ConnectionStrings": {  
  "DefaultConnection": "server=.;Database=HamburgerAppV2;trusted_connection=true;TrustServerCertificate=True"  
}  
,"Logging": {  
  "LogLevel": {  
    "Default": "Information",  
    "Microsoft.AspNetCore": "Warning"  
  }  
}  
,"AllowedHosts": "*"}
```

Şekil 3.3: Proje ile MSSQL arasında bağlantıyı sağlayan bağlantı cümlesi

Veritabanı, müşterilerin siparişlerini ve restoran menüsünü yönetmek için gerekli olan tabloları içermektedir. Temel olarak, müşteri bilgileri, ürünler ve siparişler gibi önemli veri kategorileri yer almaktadır.

Müşteriler Tablosu: Bu tablo, restorana kayıtlı olan müşterilerin bilgilerini içerir.

Ürünler Tablosu: Restoran menüsündeki ürünlerin bilgilerini içeren tablodur. Her ürün için ürün adı, fiyatı, stok durumu gibi bilgiler bulunmaktadır. Ayrıca, her ürüne benzersiz bir ID atanmıştır.

Siparişler Tablosu: Müşterilerin verdiği siparişlerin detaylarını içeren tablodur. Her sipariş, bir müşteriye ve bir veya daha fazla ürüne bağlanır. Sipariş tarihi, miktarı ve toplam fiyat bilgilerini içerir.

Bu tablolar arasında ilişkiler, referans anahtarlar kullanılarak belirlenmiştir. Örneğin, Siparişler Tablosu'ndaki müşteri ID'si, Müşteriler Tablosu'ndaki müşteri ID'sine referans verir ve böylece her sipariş bir müşteriye bağlanmış olur.

3.2 Kullanıcı Arayüzü Tasarımı

Projede, modern ve kullanıcı dostu bir kullanıcı arayüzü tasarımı oluşturulmuştur. Kullanıcı arayüzü, HTML, CSS ve JavaScript kullanılarak geliştirilmiş ve ASP.NET MVC yapılarıyla entegre edilmiştir. Temel amaç, kullanıcıların kolayca sipariş verebilmesi ve siparişlerini takip edebilmesi için deneyim sunmaktır.

Responsive Tasarım: Kullanıcıların farklı cihazlarda (bilgisayar, tablet, mobil) sorunsuz bir şekilde erişebilmesi için responsive tasarım prensipleri takip edilmiştir.

Böylece, kullanıcılar herhangi bir cihazda web uygulamasına erişebilir ve rahatça kullanabilirler.

Sipariş Erişimi: Menü yapıları ve sayfa düzenlemeleri, kullanıcıların web sitesinde kolayca gezinmesini sağlayacak şekilde düzenlenmiştir. Kullanıcılar, menüler aracılığıyla hızlıca istedikleri sayfalara erişebilir ve sipariş sürecini kesintisiz bir şekilde tamamlayabilirler.

Görsellik: Kullanıcı deneyimini artırmak için görsel unsurların (resimler, renkler vb.) kullanılmıştır. Kullanıcı arayüzü tasarımı, kullanıcıların web uygulamasını kullanırken yaşadıkları deneyimi önemli ölçüde etkiler. Bu nedenle, projede kullanıcı dostu ve etkileyici bir arayüz tasarımına önem verilmiş ve modern web tasarım prensipleri doğrultusunda geliştirilmiştir.

3.3 Sipariş Yönetim Sistemi

Bu proje, restoran sipariş yönetim sistemini etkin ve verimli bir şekilde yönetmek için geliştirilmiştir. Sistem, müşterilerin çevrimiçi olarak sipariş vermelerine olanak tanırken, restoran personeline siparişleri yönetmek için gerekli araçları sağlar. Sipariş yönetim sistemi aşağıdaki temel bileşenleri ve özellikleri içerir:

Müşteri Sipariş Sistemi: Müşteriler, restoranın menüsüne erişerek istedikleri ürünleri seçebilir ve siparişlerini oluşturabilirler. Ürünler, kategoriler halinde düzenlenmiş olup, her bir ürünün detayları ve fiyat bilgisi görüntülenebilir.

Restoran Personeli Yönetim Arayüzü: Restoran personeli, yönetim arayüzü üzerinden yeni gelen siparişleri ve/veya tüm siparişleri görüntüleyebilir.

Veritabanı Entegrasyonu: Tüm sipariş ve müşteri bilgileri, güvenli bir şekilde veritabanında saklanır. Entity Framework Code First yaklaşımı kullanılarak, veritabanı ile uygulama arasındaki veri erişimi ve yönetimi sağlanmıştır.

3.4 Yönetim Arayüzü

Proje kapsamında, sistem yöneticisinin operasyonları etkin bir şekilde yönetebilmesi için kapsamlı bir yönetim arayüzü (admin/yönetim paneli) geliştirilmiştir. Bu arayüz, kullanıcı dostu bir tasarıma sahip olup, aşağıdaki ana özellikleri sunar:

→ Menü ve Malzeme Yönetimi:

Restoran menülerinin ve malzemelerin eklenmesi, güncellenmesi ve silinmesi. (CRUD).

Menü fiyatlarının yönetilmesi.

En çok sipariş verilen menülerin listelenmesi.

En çok sipariş edilen malzemelerin listelenmesi.

Toplam cironun görüntülenmesi.

→ Müşteri Yönetimi:

Kayıtlı kullanıcıların görüntülenmesi.

Müşteri sipariş geçmişlerinin görüntülenmesi.

Bu yönetim arayüzü, restoran ile ilgili durumları daha verimli bir şekilde takip etmek amacıyla tasarlanmıştır. Yönetici, kullanıcı dostu ve sade tasarımı sayesinde hızlı bir şekilde sisteme adapte olabilir ve tüm siparişleri ve menüleri kolaylıkla yönetebilir.

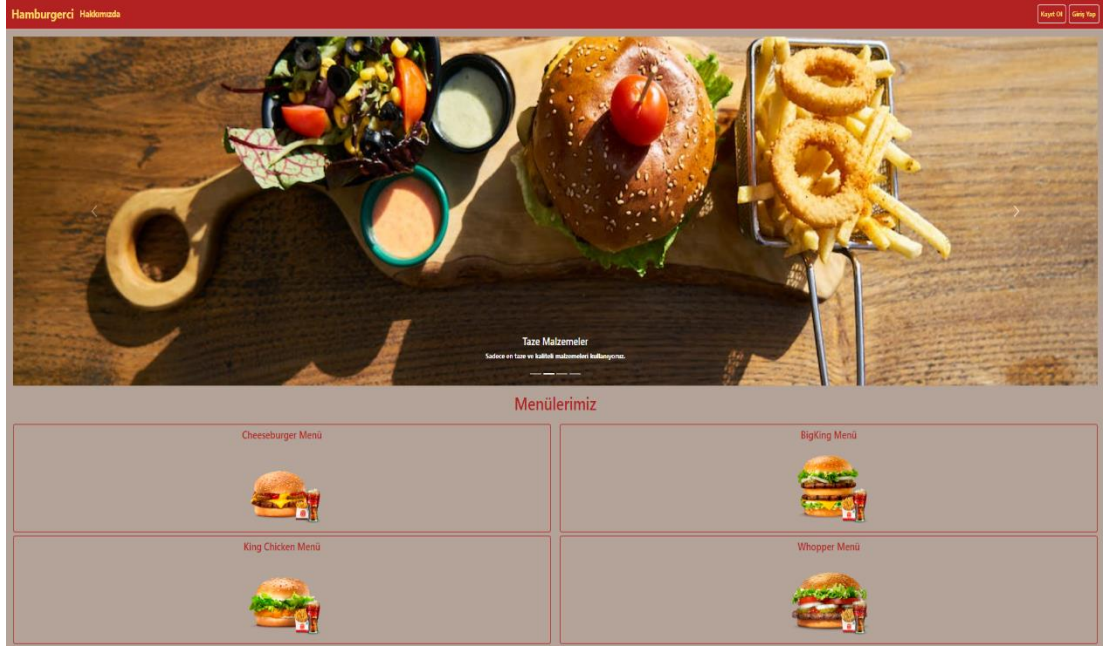
Bölüm 4

Projenin Değerlendirilmesi

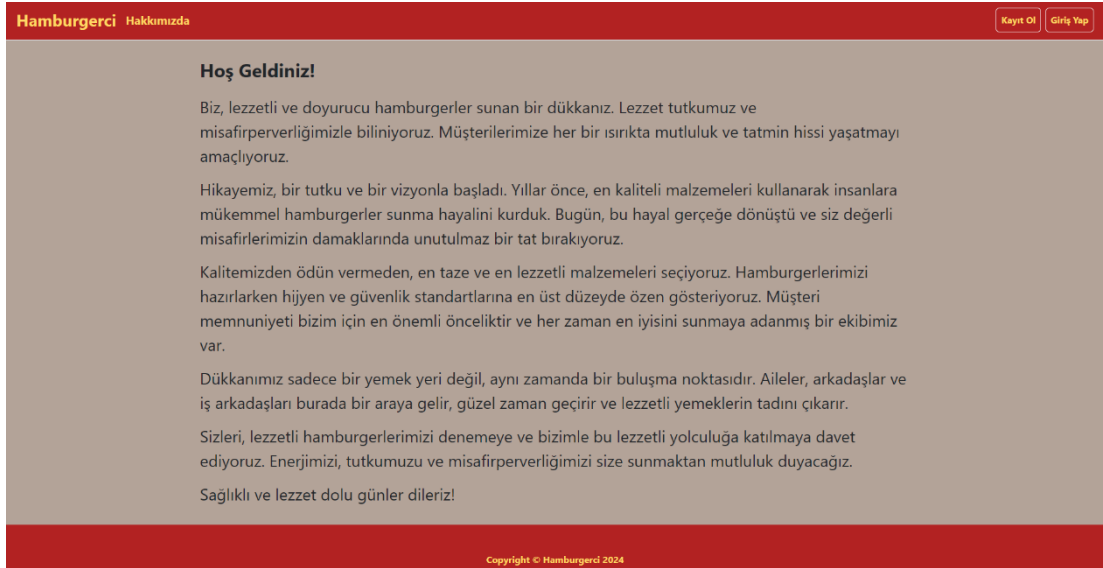
Restoran sipariş yönetim sistemi projesi, belirlenen hedeflere büyük ölçüde ulaşmıştır. Proje, kullanıcı dostu bir arayüz ve yönetim paneli ile hem müşterilere hem de restoran yönetimine kolaylıklar sunmaktadır. Müşterilerin kolayca sipariş verebileceği ve sipariş durumlarını takip edebileceği kullanıcı dostu bir arayüz geliştirilmiştir. Aynı zamanda, yönetim arayüzü restoran personelinin siparişleri ve menüleri etkili bir şekilde yönetmesini sağlamaktadır.

Entity Framework Code First yaklaşımı kullanılarak esnek ve yönetilebilir bir veritabanı modeli oluşturulmuştur. Bu yaklaşım, veritabanının kolayca güncellenebilmesini ve genişletilebilmesini sağlamaktadır. Proje, .NET 8.0 Framework, ASP.NET Core MVC, HTML, CSS ve JavaScript kullanılarak geliştirilmiştir. Ayrıca, proje esnek ve genişletilebilir bir yapıda tasarlanmış olup, gelecekte yapılabilecek geliştirmelere ve entegrasyonlara açıktır.

Projenin tüm işlevselliğini ve kullanıcı arayüzünü göstermek amacıyla, aşağıda projenin çeşitli ekran görüntüleri paylaşılmıştır. Bu ekran görüntüleri, sistemin nasıl çalıştığını ve kullanıcıya nasıl görüldüğünü net bir şekilde ortaya koymaktadır.



Şekil 4.1: Uygulama ana sayfası



Şekil 4.2: Uygulama hakkımızda sayfası

Hamburgerci Hakkımızda Kayıt Ol Giriş Yap

Üye Ol

E-mail
deneme5@gmail.com

Şifre

Şifre Tekrar

Üye Ol

Copyright © Hamburgerci 2024

Şekil 4.3: Üyelik oluşturma sayfası

Hamburgerci Hakkımızda Kayıt Ol Giriş Yap

Giriş Yap

E-Posta

Şifre

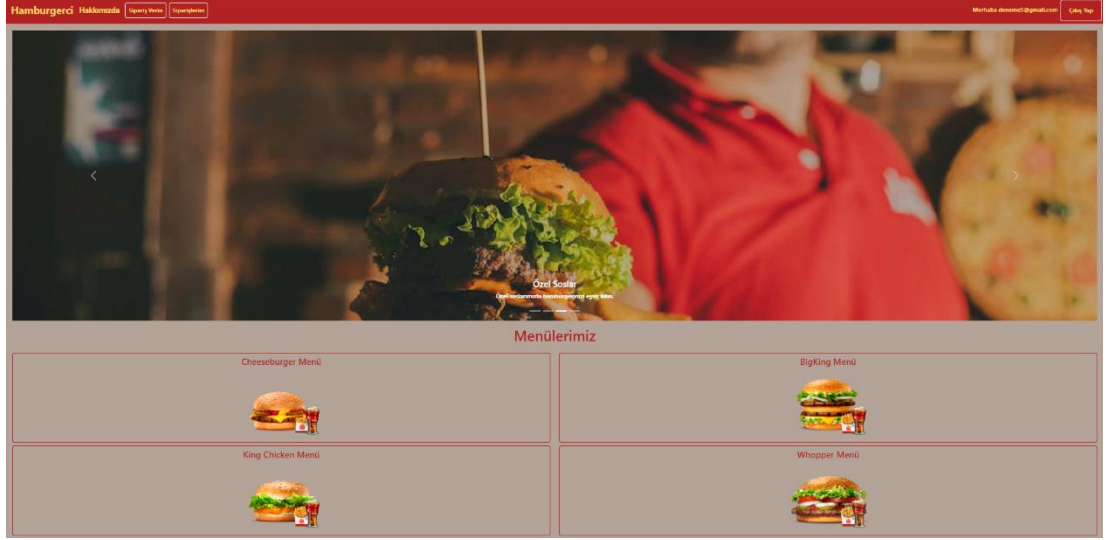
Beni Hatırla

Giriş Yap

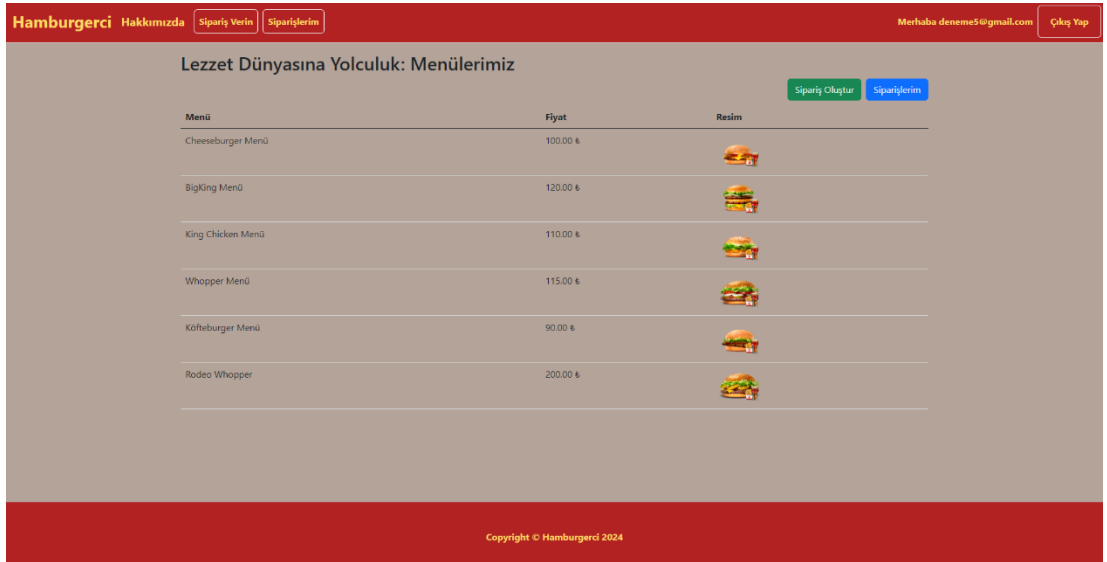
[Şifrenizi mi unuttunuz?](#)
[Yeni Kullanıcı Oluştur](#)

Copyright © Hamburgerci 2024

Şekil 4.4: Üye giriş sayfası



Şekil 4.5: Müşteri ana sayfası



Şekil 4.6: Müşteri sipariş ana sayfası

Hamburgerci Hakkımızda Sipariş Verin Siparişlerim Merhaba deneme5@gmail.com Çıkış Yap

Sipariş Oluştur

Menü Seçiniz
BigKing Menü

Boyut
Orta

Ekstra Malzemeler
 Ketchup
 Mayonez
 Ranch Sos
 Buffalo Sos
 Barbekü Sos

Adet
2

Sipariş Oluştur


Copyright © Hamburgerci 2024

Şekil 4.7: Müşteri sipariş oluşturma sayfası

Hamburgerci Hakkımızda Sipariş Verin Siparişlerim Merhaba deneme5@gmail.com Çıkış Yap

Siparis Bilgileri

Menü: BigKing Menü



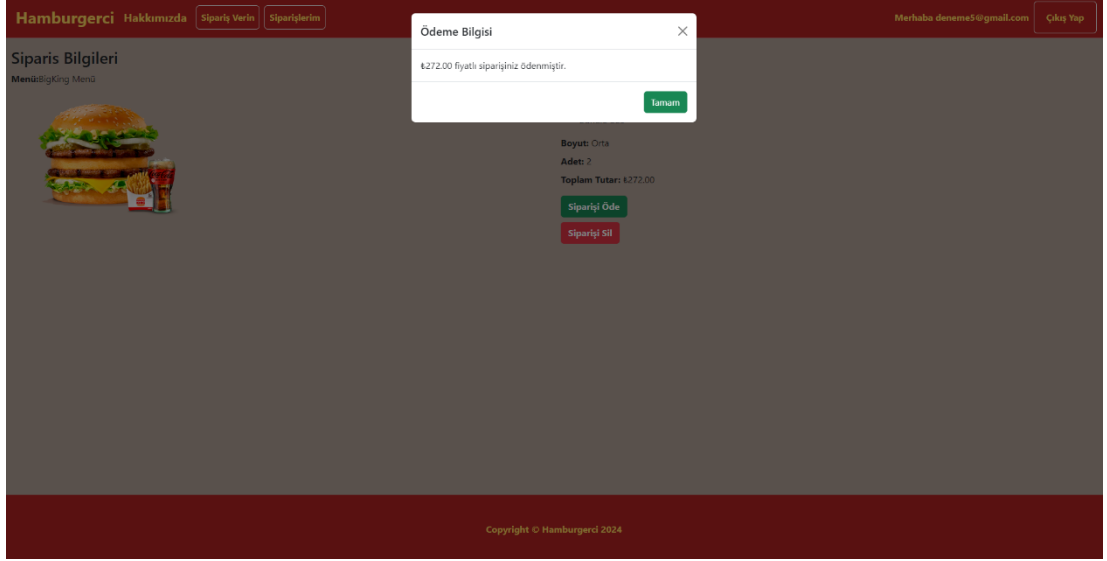
Ekstra Malzemeler:
• Ranch Sos
• Buffalo Sos

Boyut: Orta
Adet: 2
Toplam Tutar: 8272,00

Siparişi Öde
Siparişi Sil

Copyright © Hamburgerci 2024

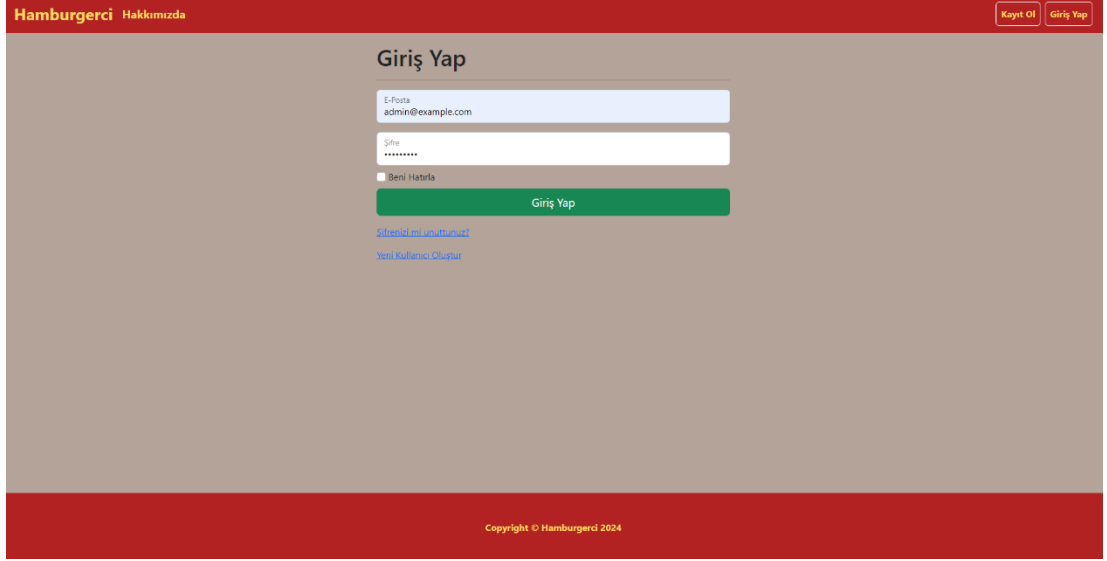
Şekil 4.8: Müşteri sipariş onay sayfası



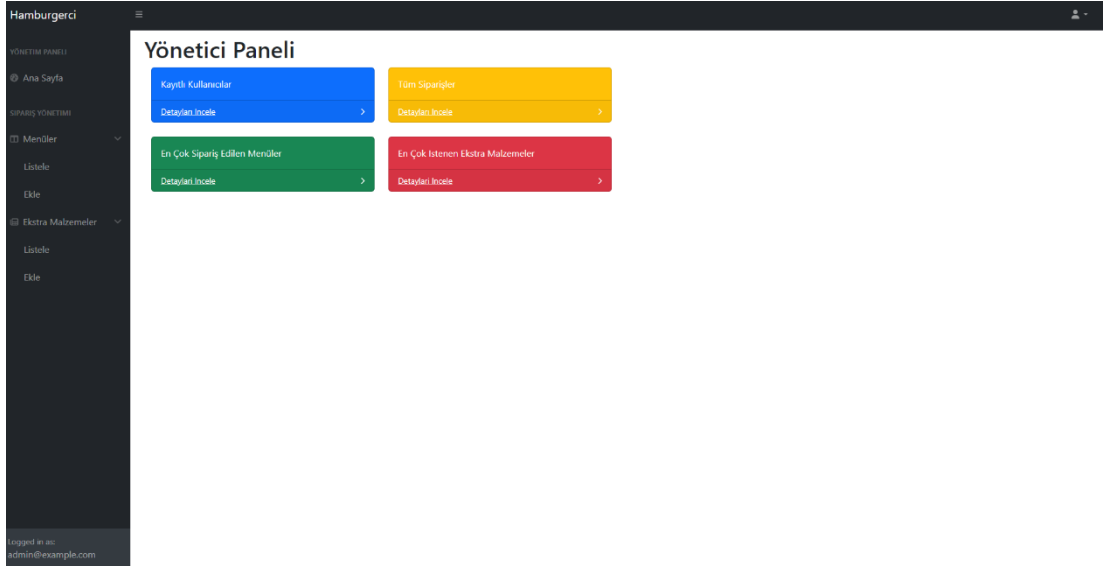
Şekil 4.9: Müşteri sipariş ödeme bilgisi sayfası



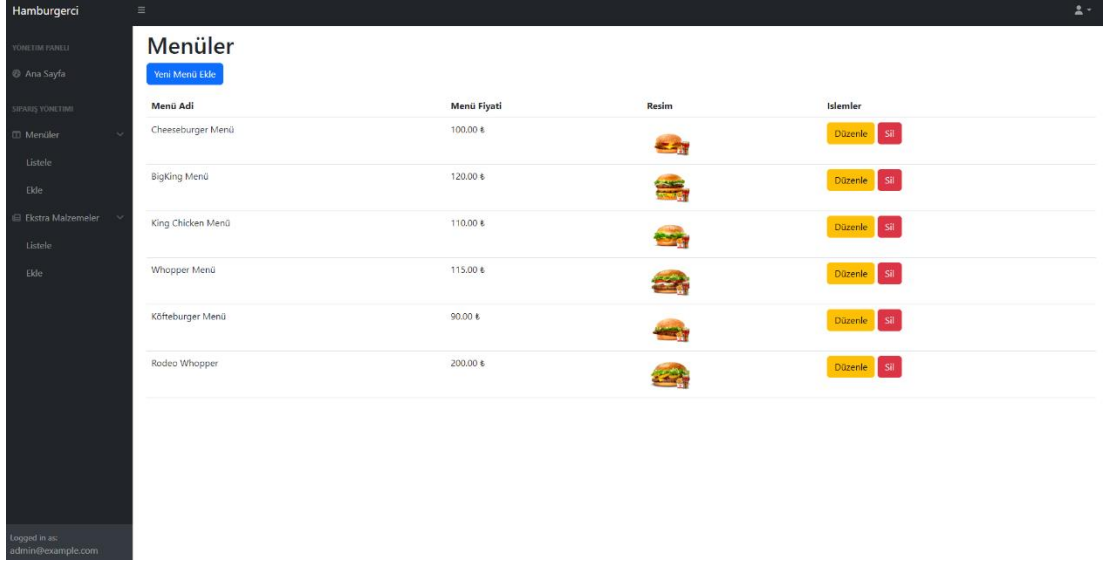
Şekil 4.10: Müşteri siparişlerini görüntüleme sayfası



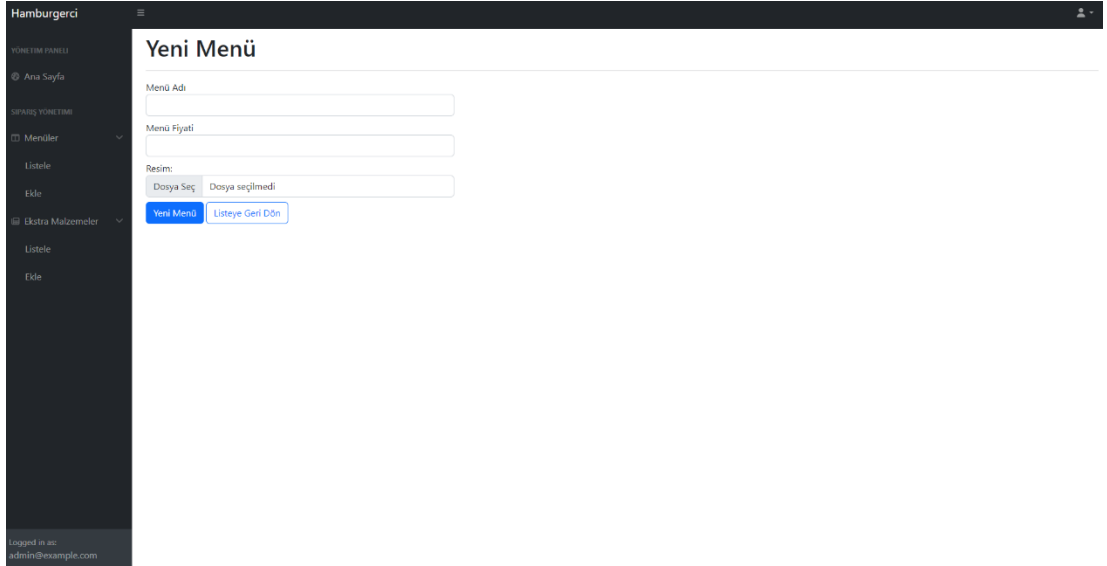
Şekil 4.11: Yönetici giriş sayfası



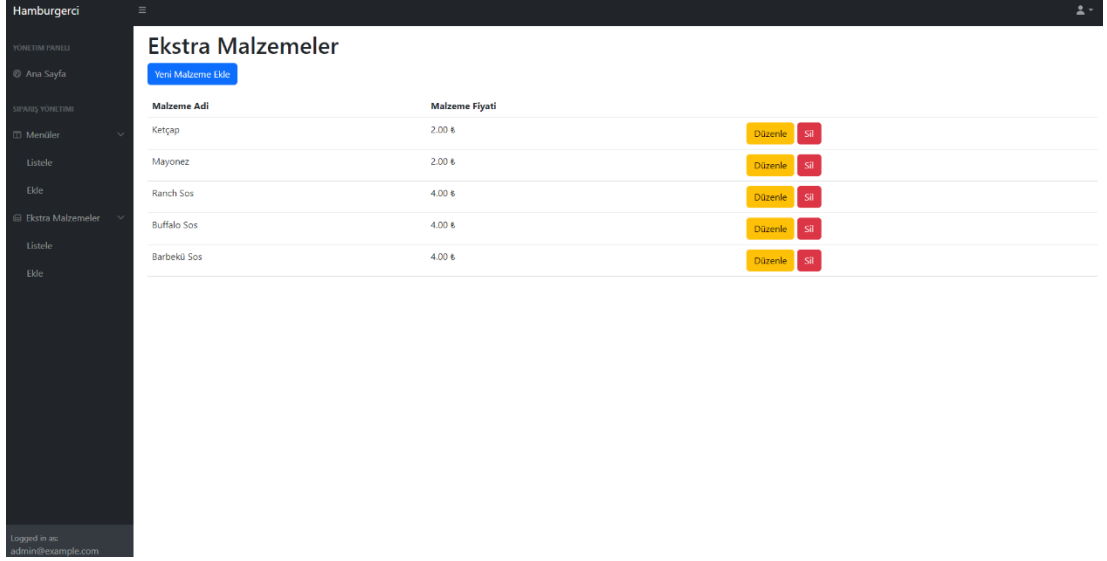
Şekil 4.12: Yönetici paneli sayfası



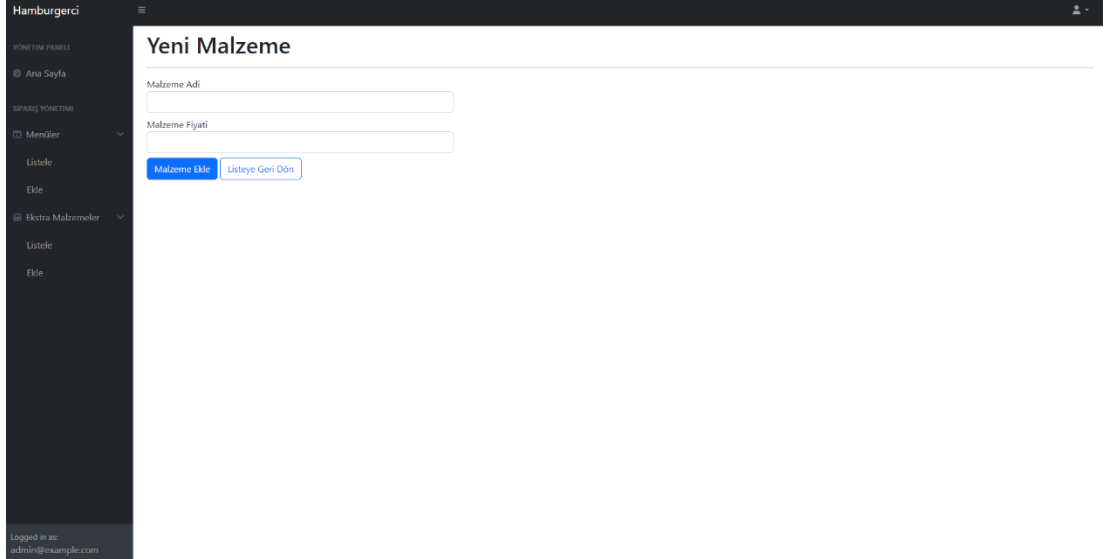
Şekil 4.13: Yönetici menü listeleme, düzenleme ve silme sayfası



Şekil 4.14: Yönetici paneli menü ekleme sayfası



Şekil 4.15: Yönetici paneli malzeme listeleme, düzenleme ve silme sayfası



Şekil 4.16: Yönetici paneli malzeme ekleme sayfası

Malzeme Adı	Kullanılma Sayısı
Ketçap	13
Mayonez	8
Ranch Sos	4
Barbekü Sos	1
Buffalo Sos	1

Şekil 4.17: Yönetici paneli en çok sipariş verilen malzemeleri görüntüleme sayfası

Menu Adı	Sayısı
BigKing Menü	9
Cheeseburger Menü	3
King Chicken Menü	3
Köfteburger Menü	3
Rodeo Whopper	1
Whopper Menü	1

Şekil 4.18: Yönetici paneli en çok sipariş verilen menüleri görüntüleme sayfası

Hamburgerci					
YÖNETİM PANELİ					
Tüm Siparişler					
Kullanıcı Adı	Menü	Ekstra Malzemeler	Sipariş Tarihi	Toplam Tutar	
deneme@gmail.com	BigKing Menü	Ketçap,Mayonez	14 Şubat 2024	400.00 ₺	
deneme@gmail.com	BigKing Menü	Ketçap,Mayonez,Ranch Sos	14 Şubat 2024	140.00 ₺	
deneme@gmail.com	King Chicken Menü	Ketçap,Mayonez	14 Şubat 2024	114.00 ₺	
deneme@gmail.com	BigKing Menü	Mayonez,Ranch Sos	14 Şubat 2024	270.00 ₺	
deneme@gmail.com	King Chicken Menü	Ketçap	14 Şubat 2024	244.00 ₺	
deneme@gmail.com	BigKing Menü	Ketçap	14 Şubat 2024	290.00 ₺	
deneme@gmail.com	Köfteburger Menü	Ketçap	14 Şubat 2024	299.00 ₺	
deneme@gmail.com	Cheeseburger Menü	Ketçap,Mayonez	14 Şubat 2024	224.00 ₺	
deneme@gmail.com	BigKing Menü	Ketçap	14 Şubat 2024	122.00 ₺	
deneme@gmail.com	BigKing Menü	Barbekü Sos	14 Şubat 2024	292.00 ₺	
deneme2@gmail.com	Köfteburger Menü	Ketçap,Mayonez	15 Şubat 2024	544.00 ₺	
deneme2@gmail.com	Whopper Menü	Ketçap	15 Şubat 2024	232.00 ₺	
deneme@gmail.com	Cheeseburger Menü	Ranch Sos	15 Şubat 2024	114.00 ₺	
deneme3@gmail.com	Köfteburger Menü		15 Şubat 2024	99.00 ₺	
deneme4@gmail.com	Cheeseburger Menü	Ketçap,Mayonez	16 Şubat 2024	224.00 ₺	
deneme@gmail.com	BigKing Menü	Ketçap	16 Şubat 2024	122.00 ₺	
deneme@gmail.com	BigKing Menü	Ketçap	16 Şubat 2024	398.00 ₺	
deneme@gmail.com	King Chicken Menü		16 Şubat 2024	121.00 ₺	
deneme@gmail.com	Rodeo Whopper	Mayonez	18 Şubat 2024	242.00 ₺	
deneme5@gmail.com	BigKing Menü	Ranch Sos,Buffalo Sos	18 Mayıs 2024	272.00 ₺	
				Toplam Ciro: 4763.00 ₺	

Şekil 4.19: Yönetici paneli sipariş verilen tüm menüleri görüntüleme sayfası

Hamburgerci		
YÖNETİM PANELİ		
Kayıtlı Kullanıcılar		
Kullanıcı Adı	E-Posta	
deneme@gmail.com	deneme@gmail.com	
deneme2@gmail.com	deneme2@gmail.com	
deneme5@gmail.com	deneme5@gmail.com	
deneme3@gmail.com	deneme3@gmail.com	
deneme4@gmail.com	deneme4@gmail.com	

Şekil 4.20: Yönetici paneli kayıtlı kullanıcıları görüntüleme sayfası

Bu proje, restoranların dijitalleşmesine katkı sağlayarak operasyonel verimliliği artırmayı ve müşteri memnuniyetini en üst düzeye çıkarmayı hedeflemektedir. Projenin başarıyla tamamlanması, bu hedeflere ulaşmada önemli bir adım olmuştur.

Kaynaklar

Turan, F. (2018). ASP. NET MVC 4 teknolojilerini kullanarak bir e-ticaret sitesi uygulamasının geliştirilmesi (Yayın No. 490547) [Yüksek lisans tezi, Beykent Üniversitesi]. YÖK Ulusal Merkezi. <https://tez.yok.gov.tr/UlusalTezMerkezi/>

Balcı, T. (2018). Entity framework'ün farklı veri tabanlarındaki performans analizi (Yayın No. 514187) [Yüksek lisans tezi, Kırıkkale Üniversitesi]. YÖK Ulusal Merkezi. <https://tez.yok.gov.tr/UlusalTezMerkezi/>

Altıntaş, B. (2019). A security comparison of Oracle, SQL Server and MySQL database management systems against SQL injection attack vulnerabilities (Yayın No. 599900) [Yüksek lisans tezi, Yaşar Üniversitesi]. YÖK Ulusal Merkezi. <https://tez.yok.gov.tr/UlusalTezMerkezi/>

<https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>

<https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>

<https://medium.com/swlh/entity-framework-core-with-the-code-first-approach-in-net-core-project-f2065b2c28d5>

<https://coskunkurtuldu.medium.com/net-core-entity-framework-code-first-yakla%C5%9F%C4%B1m%C4%B1-6ac02867b3e5>

https://simple.wikipedia.org/wiki/Structured_Query_Language