



SOFTITO YAZILIM BİLİŞİM AKADEMİSİ
BACKEND DEVELOPER EĞİTİMİ
BİTİRME PROJESİ RAPORU

GM SEYAHAT ACENTESİ
WEB TABANLI REZERVASYON VE YÖNETİM SİSTEMİ

Hazırlayanlar:

Gamze Türker

Merve Gezginci

Temmuz, 2026

softITO Yazılım Bilişim Akademisi, İstanbul

İÇİNDEKİLER

1. GİRİŞ

1.1. Projenin Amacı

1.2. Proje Kısıtları

2. MATERYAL VE YÖNTEM

2.1. ASP.NET Core MVC Mimarisi

2.2. Katmanlı Mimari Tasarımı (N-Tier Architecture)

2.3. Entity Framework Core ve Dapper Entegrasyonu

2.4. Tasarım Desenleri (Repository ve Unit of Work)

2.5. Önbellekleme (IMemoryCache) ve Raporlama Servisleri (PDFsharp, ClosedXML)

2.6. Gemini Yapay Zeka ve Kural Tabanlı Akıllı Sohbet Servisi

3. SİSTEMİN TASARIMI VE KURULUMU

3.1. Veritabanı Modelleri ve İlişki Yapıları

3.2. Projenin Kurulumu ve Yapılandırılması

3.3. Sistem Özelliklerinin ve İşlevlerinin Kullanımı

3.3.1. Ziyaretçi ve Kullanıcı Arayüzü İşlevleri

3.3.2. Yönetici Kontrol Paneli (Admin Console) İşlevleri

3.3.3. Dinamik PDF Fatura ve Excel Raporlama Çıktıları

4. SONUÇ

5. KAYNAKÇA

1. GİRİŞ

Bu bölümde projenin geliştirilme amacı, hedef kitlesi ve projenin gerçekleştirilmesi sürecinde karşılaşılan veya uygulanan teknik kısıtlar ele alınmıştır.

1.1. Projenin Amacı

Projenin temel amacı; seyahat acentelerinin tur listeleme, rezervasyon alımı, üye yönetimi, kampanya/kupon dağıtımı ve raporlama gibi operasyonel süreçlerini tek bir platform üzerinden yürütebilmelerini sağlayan web tabanlı bir sistem geliştirmektir. "GM Seyahat Acentesi" adı altında tasarlanan bu platform, hem son kullanıcılara (ziyaretçi ve üyeler) kullanıcı dostu ve hızlı bir rezervasyon deneyimi sunmayı hem de acente yöneticilerine (admin) arka plandaki tüm süreçleri denetleyebilecekleri kapsamlı bir yönetim paneli sağlamayı hedeflemektedir.

Sistemin geliştirilmesinde, modern web standartlarına uygunluk, yüksek performans, veri tutarlılığı ve güvenlik ön planda tutulmuştur. Özellikle veri tabanındaki ilişkilerin yönetimi, kapasite kontrollerinin anlık yapılması, admin işlemlerinin denetlenmesi (Audit Log) ve hızlı sayfa yükleme süreleri (caching ve dapper yardımıyla) projenin işlevsel amaçları arasında yer almaktadır.

1.2. Proje Kısıtları

Projenin tasarımı ve uygulanması aşamasında bazı kısıtlayıcı faktörler göz önünde bulundurulmuştur:

- **Teknoloji Kısıtları:** Projenin arka yüzünde (backend) .NET 8.0/9.0 ve C# dilinin kullanılması, veri erişiminde ise Code-First yaklaşımıyla Entity Framework Core'un yanı sıra performans gerektiren okuma işlemlerinde Dapper Micro-ORM'in kullanılması zorunlu kılınmıştır.
- **Veritabanı Kısıtları:** İlişkisel veri tabanı yönetim sistemi olarak Microsoft SQL Server tercih edilmiştir. Tüm ilişkilerin (1-N ve 1-1) yabancı anahtarlarla (foreign key) veri tabanı seviyesinde modellenmesi ve işlemler sırasında veri tutarlılığının korunması sağlanmıştır.
- **Tasarım ve Arayüz Kısıtları:** Arayüzün mobil uyumlu (responsive) olması amacıyla Bootstrap 5.3 kütüphanesi kullanılmıştır. Kullanıcı deneyimini artırmak için dinamik harita gösterimi (Leaflet.js) ve animasyonlu uyarı kutuları (SweetAlert2) entegre edilmiştir.
- **Erişim Yetkilendirmeleri:** Giriş yapan kullanıcıların (User) ve yöneticilerin (Admin) yetki sınırları ASP.NET Core Identity altyapısı ile kesin hatlarla çizilmiş, korumalı sayfalara yetkisiz erişimler engellenmiştir.

2. MATERYAL VE YÖNTEM

Projenin yazılım mimarisi, tercih edilen teknolojiler ve kullanılan yazılım tasarım desenleri bu başlık altında detaylandırılmıştır.

2.1. ASP.NET Core MVC Mimarisi

Projede sunum katmanının tasarımı için Model-View-Controller (MVC) mimari deseni tercih edilmiştir. MVC deseni, uygulamanın iş mantığını (Model), kullanıcıya gösterilen arayüzü (View) ve gelen istekleri yönlendiren kontrol mekanizmasını (Controller) birbirinden ayırarak kodun okunabilirliğini ve bakımını kolaylaştırır. Projede üye işlemleri ve yönetici paneli gibi farklı alanlar, MVC yapısının sunduğu "Areas" özelliği kullanılarak 'Admin' ve 'User' şeklinde modüller olarak ayrılmıştır.

2.2. Katmanlı Mimari Tasarımı (N-Tier Architecture)

Uygulamanın esnek, genişletilebilir ve test edilebilir olması için sorumluluklar üç farklı projeye (katmana) bölünmüştür:

1. **seyahat_projesi (Sunum Katmanı):** Kullanıcının etkileşime girdiği Razor görünümüleri (Views), kullanıcı isteklerini karşılayan denetleyiciler (Controllers), tarayıcı tarafındaki statik dosyalar (wwwroot) ve harici servis entegrasyonları bu katmanda yer alır.

2. **seyahat_projesi.Data (Veri Erişim Katmanı):** Veritabanı bağlantısı ('ApplicationDbContext'), veritabanı şemasını oluşturan göç dosyaları (Migrations), veri tabanı başlatıcıları ve tohum verileri ('DbInitializer', 'DbSeeder') ile veri erişim desenleri (Repositories) bu katmanda konumlandırılmıştır.

3. **seyahat_projesi.Model (Model Katmanı):** Sistemde kullanılan tüm veritabanı varlıkları (Entities) ile veriyi taşımak ve doğrulamak amacıyla kullanılan yardımcı modeller (ViewModels ve DTO'lar) bu katmandadır.

2.3. Entity Framework Core ve Dapper Entegrasyonu

Projede veri erişim süreçlerinde hibrit bir yaklaşım benimsenmiştir:

- **Entity Framework Core (EF Core):** Ekleme, güncelleme, silme (CRUD) ve ilişkisel tablolar arasındaki karmaşık işlemler (özellikle Transaction yönetimi gerektiren durumlar) EF Core 8.0 üzerinden gerçekleştirilmiştir. Code-First yaklaşımı sayesinde C# sınıfları veri tabanındaki tablolara eşlenmiş ve veri tabanı şeması tamamen kod üzerinden yönetilmiştir.

- **Dapper Micro-ORM:** Yüksek performans gerektiren veri listeleme, arama ve filtreleme süreçlerinde (örneğin ana sayfadaki popüler turların çekilmesi, kategori filtrelemeleri) Dapper tercih edilmiştir. Dapper, ham SQL sorgularını doğrudan çalıştırarak nesne eşleştirmelerini (mapping) milisaniyeler seviyesinde tamamlar ve sunucu kaynaklarının daha verimli kullanılmasını sağlar.

2.4. Tasarım Desenleri (Repository ve Unit of Work)

Veri tabanı işlemlerinin tek bir merkezden yönetilmesi ve sunum katmanının veri erişim teknolojisine doğrudan bağımlı olmaması için şu desenler uygulanmıştır:

- **Repository Pattern:** Veritabanı tabloları üzerinde yapılacak temel CRUD işlemleri soyutlanmış ('IRepository'), ortak metotlar tek bir sınıf ('Repository.cs') üzerinden kalıtım alınarak implemente edilmiştir. Böylece kod tekrarlarının önüne geçilmiştir.
- **Unit of Work Pattern:** Yapılan tüm işlemlerin (birden fazla tablonun etkilendiği durumlar) tek bir veritabanı işlemi (Transaction) olarak ele alınması sağlanmıştır. 'IUnitOfWork' arayüzü sayesinde tüm depolar (Repositories) tek bir bağlam ('DbContext') paylaşır ve 'Save()' metodu tetiklenene kadar değişiklikler fiziksel veri tabanına yazılmaz. Hata oluşması durumunda tüm işlemler geri alınarak veri bütünlüğü korunur.

2.5. Önbellekleme (IMemoryCache) ve Raporlama Servisleri

- **IMemoryCache:** Sık güncellenmeyen ancak sayfalar arası geçişlerde sürekli okunan veriler (örneğin aktif kategoriler ve popüler tur listeleri) RAM üzerinde önbelleğe alınmıştır. Bu sayede veri tabanı sorgu yükü azaltılmış ve sayfa yüklenme süreleri yaklaşık 10 kat hızlandırılmıştır. Tur veya kategori güncellendiğinde önbellek otomatik olarak geçersiz kılınmaktadır.
- **PDFsharp & MyFontResolver:** Kullanıcıların rezervasyonları başarıyla tamamlandığında adlarına düzenlenen faturaları PDF formatında indirebilmesi için PDFsharp kütüphanesi kullanılmıştır. Türkçe karakterlerin ('ğ, ş, ı, ç, ö, ü') düzgün görüntülenebilmesi için işletim sistemindeki yazı tiplerini çözümleyen özel bir 'MyFontResolver' sınıfı yazılmıştır.
- **ClosedXML:** Yönetim panelinde yöneticilerin rezervasyon kayıtlarını ve sistem günlüklerini (Audit Logs) Excel formatında (.xlsx) indirebilmesi sağlanmıştır. Bu servis sayesinde veriler temiz hücre yapılarıyla raporlanabilmektedir.

2.6. Gemini Yapay Zeka ve Kural Tabanlı Akıllı Sohbet Servisi

Ziyaretçilerin sorularını hızlıca yanıtlamak amacıyla 'HomeController.cs' içerisinde entegre bir canlı destek sohbet botu servisi tasarlanmıştır. Bu servis iki aşamalı bir yanıt mekanizması ile çalışmaktadır:

- **Google Gemini Generative AI:** Sistemde tanımlı bir API anahtarı bulunması durumunda sohbet robotu doğrudan Gemini API'sini çağırarak kullanıcı mesajlarına doğal dilde yapay zeka yanıtları üretir (varsayılan olarak 'gemini-3.5-flash' veya 'gemini-3.1-flash-lite' modelleri denenir).
- **C# Kural Tabanlı Fallback Sistemi:** API anahtarı bulunmadığında veya sunucu hatası oluştuğunda, sistem C# tarafında yazılmış kural tabanlı bir mekanizmaya geçer. Bu mekanizma anahtar kelime eşlemeleriyle Sıkça Sorulan Soruları (SSS) veri tabanından çeker, turların detaylarını yanıtlar ve oturum açmış kullanıcılara özel veri tabanından anlık kupon kodlarını çekerek dinamik yanıtlar üretir.

3. SİSTEMİN TASARIMI VE KURULUMU

Bu bölümde uygulamanın ilişkisel yapısı, kurulum adımları ve geliştirilen ekranların işlevsel özellikleri açıklanmıştır.

3.1. Veritabanı Modelleri ve İlişki Yapıları

Sistemde yer alan veritabanı tabloları ve aralarındaki ilişkiler `ApplicationDbContext.cs` üzerinde modellenmiştir:

- **Category (Kategori) ve Tour (Tur):** Bir kategoride (Kültür, Doğa, Yaz vb.) birden fazla tur bulunabilir (1-N İlişki). Tur tablosundaki `CategoryId` alanı `Category` tablosunun birincil anahtarına (Primary Key) bağlıdır.
- **Guide (Rehber) ve Tour (Tur):** Bir tur rehberi birden fazla tura liderlik edebilir (1-N İlişki).
- **ApplicationUser (Kullanıcı) ve Booking (Rezervasyon):** Bir üye sistem üzerinden birden fazla seyahat rezervasyonu yapabilir (1-N İlişki).
- **Booking (Rezervasyon) ve Payment (Ödeme):** Her rezervasyonun yalnızca bir ödeme kaydı bulunur (1-1 İlişki). Rezervasyon silindiğinde ilişkili ödeme kaydı da veri tabanından silinecek şekilde (Cascade) yapılandırılmıştır.
- **Tour (Tur) ve Review (Değerlendirme):** Bir tur için kullanıcılar birden fazla yorum ve puanlama yapabilir (1-N İlişki).
- **SystemLog (Sistem Logu):** Admin paneli üzerinden gerçekleştirilen kritik operasyonlar (CRUD, cache temizleme, rezervasyon iptali vb.) işlemi yapan kullanıcının kimliği, işlem açıklaması, IP adresi ve işlem zamanı ile birlikte kaydedilir.

3.2. Projenin Kurulumu ve Yapılandırılması

Sistemin yerel (lokal) ortamda çalıştırılabilmesi için aşağıdaki adımların takip edilmesi gerekir:

1. **Bağlantı Dizesi (Connection String) Ayarı:** `seyahat\_projesi/appsettings.json` dosyasındaki SQL Server bağlantı dizesi yerel sunucuya göre düzenlenir:

```
```json
"ConnectionStrings": {
 "DefaultConnection":
 "Server=LOKAL_SUNUCU_ADINIZ;Database=SeyahatDb;Trusted_Connection=True;TrustServerCertificate=True"
}
```
```

2. **Paketlerin Geri Yüklenmesi (Dotnet Restore):** Projenin bağımlılıklarının kurulması için terminalde proje kök dizininde şu komut çalıştırılır:

```
```bash
```

dotnet restore

```

3. **Uygulamanın Başlatılması (Dotnet Run):** Sunum katmanı projesini çalıştırmak için şu komut girilir: ``bash dotnet run --project seyahat_projesi

```

**Not:** Uygulama ilk kez çalıştırıldığında `Program.cs` içerisindeki veritabanı göç (migration) kontrol mekanizması devreye girer. Veritabanı yoksa oluşturulur (`MigrateAsync`), ardından sistemin çalışması için gerekli roller, admin hesapları (`DbInitializer.SeedAsync`) ve her tablo için 20'şer adet örnek kayıt (`DbSeeder.Seed20Async`) otomatik olarak veri tabanına yüklenir.

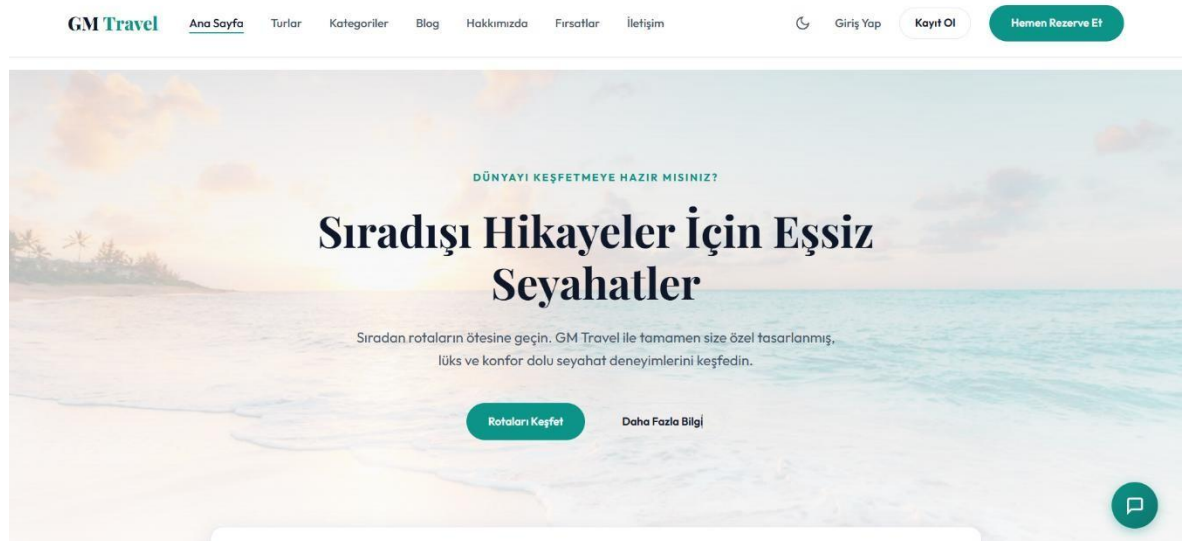
### 3.3. Sistem Özelliklerinin ve İşlevlerinin Kullanımı

Geliştirilen uygulamanın kullanıcı ve yönetici taraflarındaki temel işlevleri ve ekran görüntüleri aşağıda listelenmiştir:

#### 3.3.1. Ziyaretçi ve Kullanıcı Arayüzü İşlevleri

Kullanıcı arayüzü, seyahat etmek isteyen ziyaretçilerin turları kolayca inceleyip satın alabilmesi üzerine tasarlanmıştır:

- **Ana Sayfa:** RAM önbelleğinden (caching) beslenen popüler turları, dinamik kategori slider bileşenini ve duyuru şeritlerini içerir. Arayüzün sağ üst köşesinden açık/koyu (Light/Dark) tema geçişi yapılabilir.

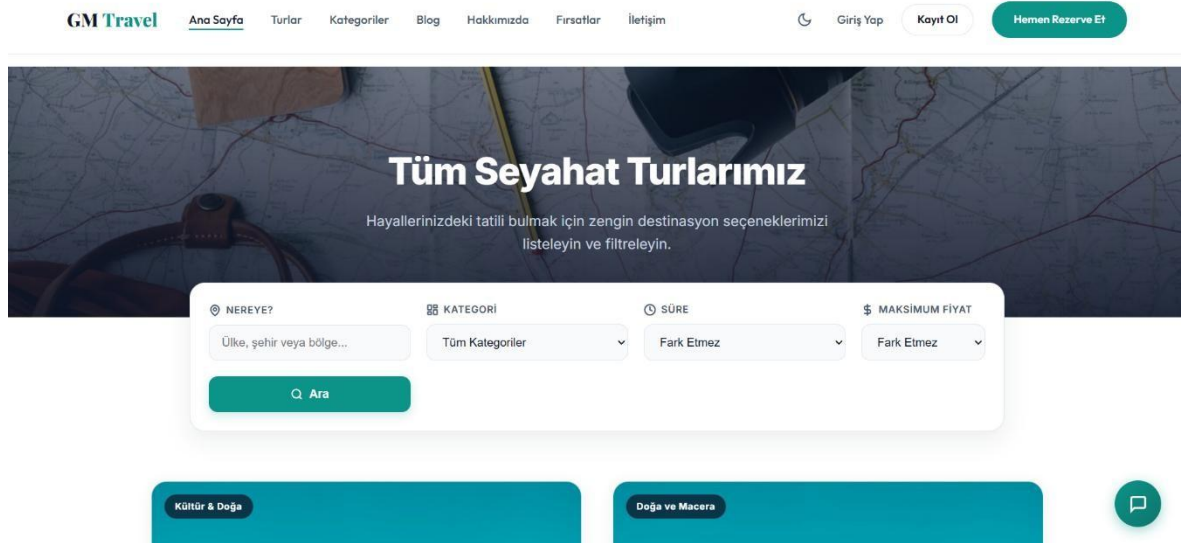


Şekil 1. Ana Sayfa - Açık Tema



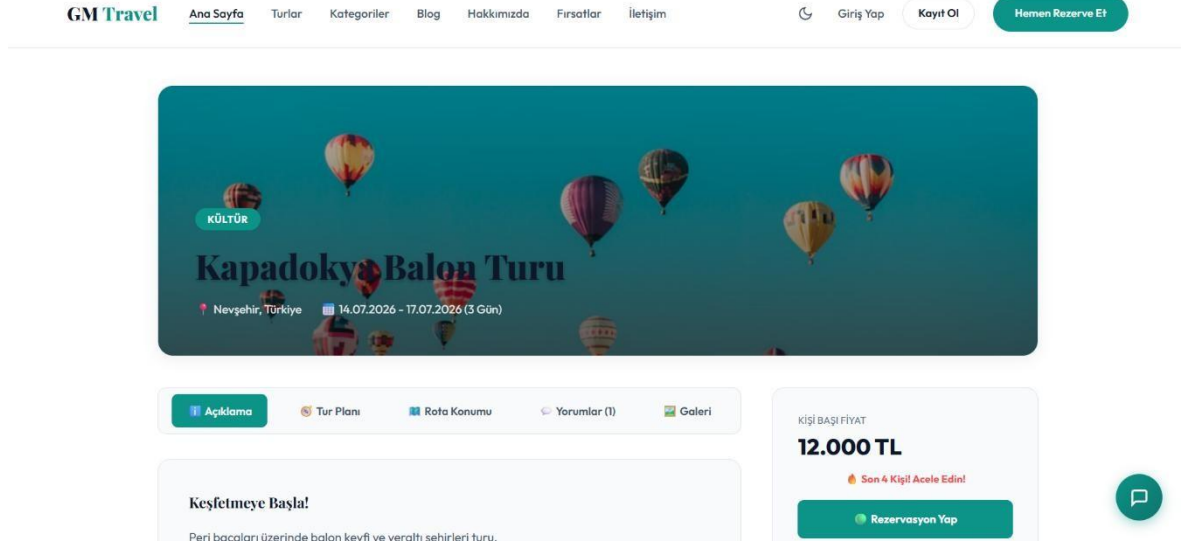
Şekil 2. Ana Sayfa - Koyu Tema

- **Arama ve Filtreleme:** Ziyaretçiler gitmek istedikleri lokasyona, tur kategorisine veya minimum/maksimum fiyat aralığına göre Dapper micro-ORM hızıyla listelenen turları anlık olarak filtreleyebilirler.

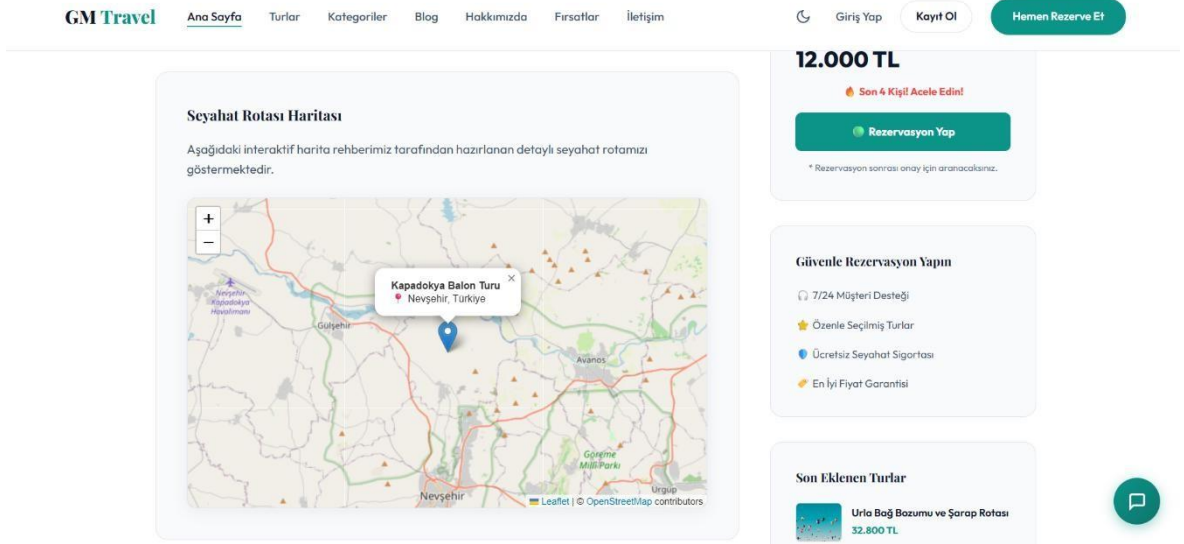


Şekil 3. Turlar ve Arama Ekranı

- **Tur Detay Sayfası:** Turun rotası Leaflet.js entegre edilmiş interaktif bir harita üzerinde gösterilir. Ayrıca tur rehberinin bilgileri, seyahat programı, kontenjan durumu ve tura katılan diğer üyelerin yazdığı yorumlar listelenir.

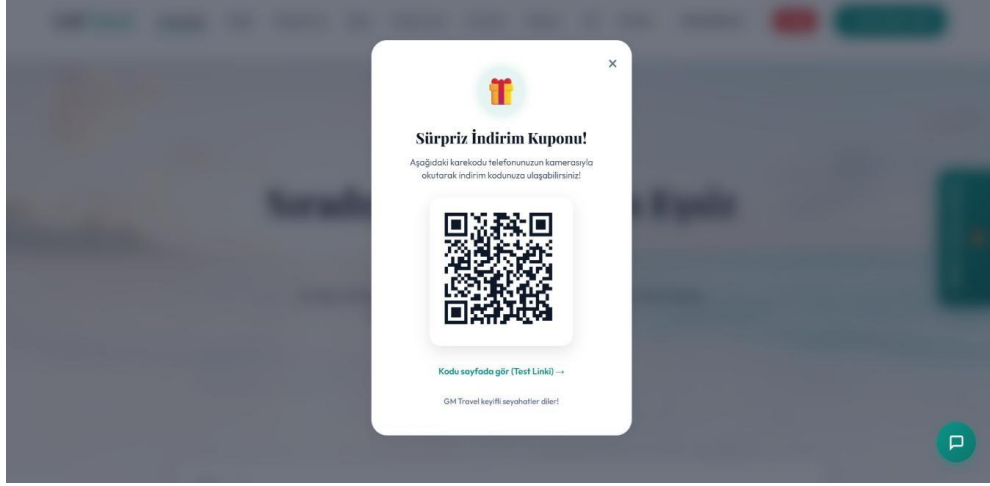


Şekil 4. Tur Detay Bilgileri

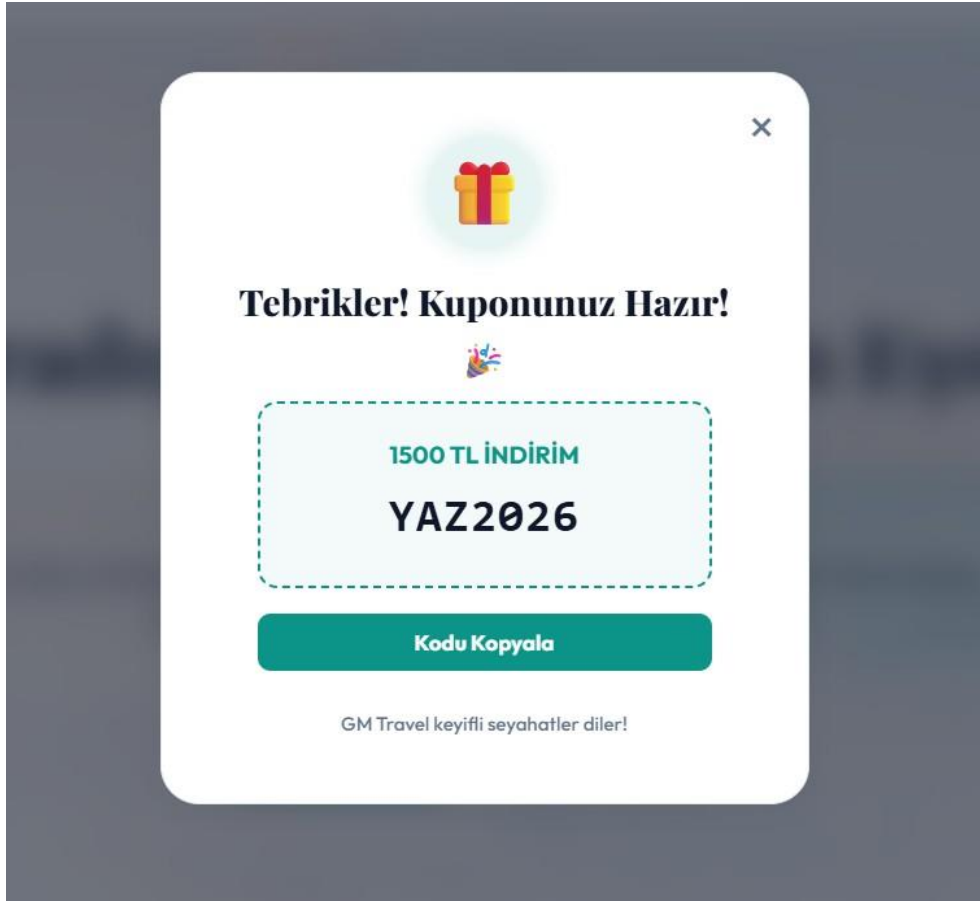


Şekil 5. Seyahat Rotası Haritası

- **Kupon QR Kod Kazanma Modalı:** Kullanıcılar site içerisindeki sürpriz QR kodları taratarak indirim kuponu kazanabilirler. Kazanılan kupon kodları AJAX doğrulamasıyla ödeme ekranında anlık indirim olarak hesaba yansıtılır.



Şekil 6. Sürpriz Kupon QR Kod Modalı



Şekil 7. Kupon Başarı Modalı

- **Rezervasyon ve Satın Alma (Checkout):** Kullanıcı seyahat edecek kişi sayısını girer, varsa kupon kodunu uygular ve ödeme yöntemi simülasyonu ile rezervasyonu tamamlar. İşlem başarılı olduğunda benzersiz bir işlem kodu (TX) üretilir.

**GM Travel**[Ana Sayfa](#)[Turlar](#)[Kategoriler](#)[Blog](#)[Hakkımızda](#)[Fırsatlar](#)[İletişim](#)[Profilim](#)[Çıkış](#)[Hemen Rezerve Et](#)

 **Kişisel Bilgiler**

Ad Soyad

MERVE GEZGİNCİ

E-posta

merve@gmail.com

 **Tur ve Tarih Seçimi**

Seçilen Rota

Fırtına Deresi Rize Yaylaları

Gidiş Tarihi

26.07.2026

Kişi Sayısı

4



**DOĞA VE MACERA**  
**Fırtına Deresi Rize Yaylaları**  
Rize, Türkiye

Müşteri: MERVE GEZGİNCİ

Kişi Sayısı: 4 Kişi

Kişi Başı Fiyat: 11.200 TL

Toplam Tutar:

**44.800 TL**

**REZERVASYONUNU TAMAMLA**

**Sürpriz Kupon Kazan!**



Şekil 8. Rezervasyon Sayfası

## Rezervasyonunuz Alındı!

Ödeme işleminiz başarıyla tamamlandı ve onaylandı.

**Rezervasyonunuz ve ödeme işleminiz başarıyla tamamlandı!**

**Seyahat Detayları**

 **Rota:** Fırtına Deresi Rize Yaylaları

 **Konum:** Rize, Türkiye

 **Gidiş Tarihi:** 26.07.2026

 **Kişi Sayısı:** 4 Kişi

 **Ödeme Yöntemi:** Kredi / Banka Kartı

 **İşlem Kodu (TX):** TXN-7A59A8EF

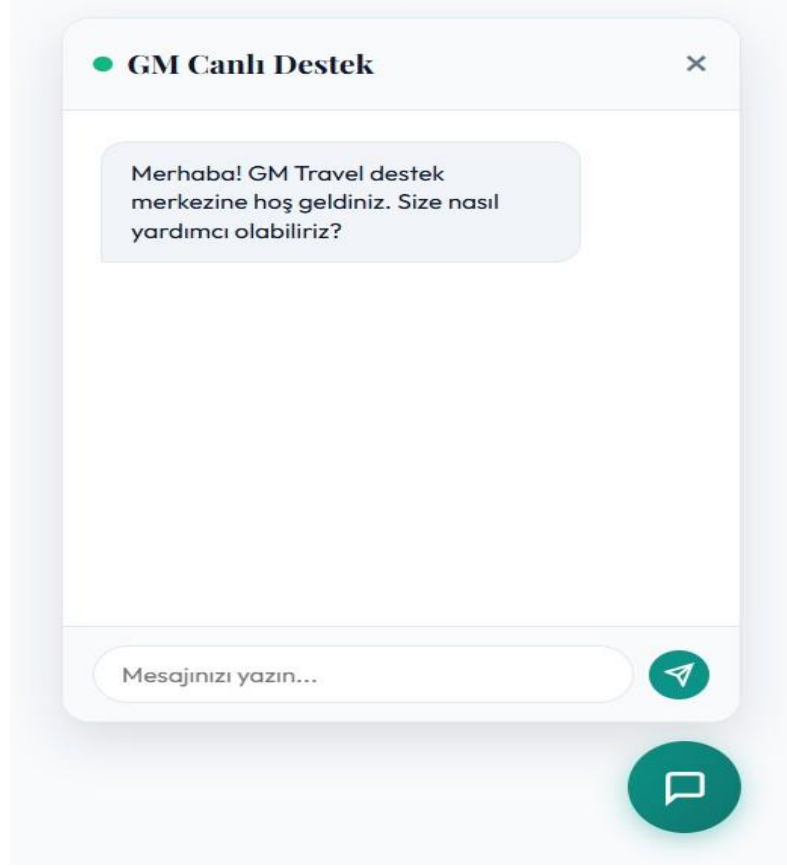
**Ödenen Toplam:**

**44.800 TL**

\* Rezervasyon sonrasında onay ve bilgilendirme için temsilcilerimiz (Gamze veya Merve) tarafından aranacaksınız.

Şekil 9. Rezervasyon Başarılı Ekranı

- Canlı Destek (Chatbot) Modülü:** Web sayfalarının sağ alt kısmında yer alan sohbet arayüzü sayesinde, kullanıcılar sistemle anlık yazışabilmektedir. Gemini entegrasyonu veya kural tabanlı veri tabanı eşleşmeleri ile kullanıcılara akıllı yönlendirmeler yapılır.

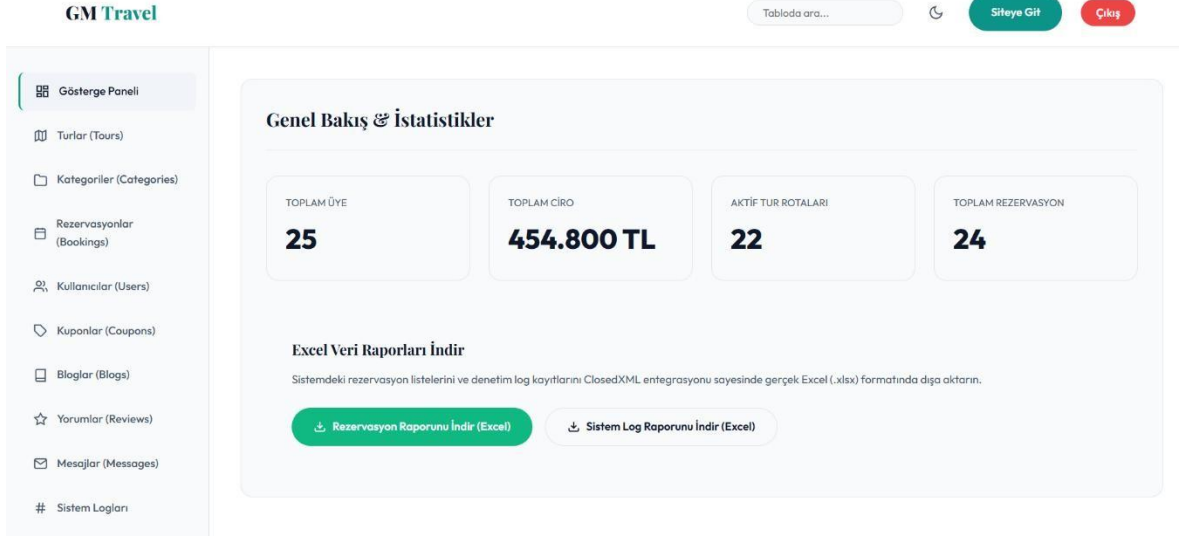


Şekil 10. Canlı Destek Arayüzü

### 3.3.2. Yönetici Kontrol Paneli (Admin Console) İşlevleri

Yalnızca 'Admin' rolüne sahip kullanıcıların erişebildiği kontrol paneli, sistemin tüm veri akışını yönetir:

- **Yönetici Dashboard:** Toplam ciro, aktif üye sayısı, tamamlanan ve iptal edilen rezervasyon miktarları gibi özet istatistikleri grafikler ve kartlar eşliğinde gösterir.



Şekil 11. Yönetici Dashboard Ekranı

- Tur, Kategori ve Rehber CRUD:** Yöneticiler yeni tur rotaları ekleyebilir, mevcut olanları düzenleyebilir veya pasife alabilirler. Görseller sunucuya yüklenerek kayıt altına alınır.

GM Travel

Tabloda ara...

Siteye Git

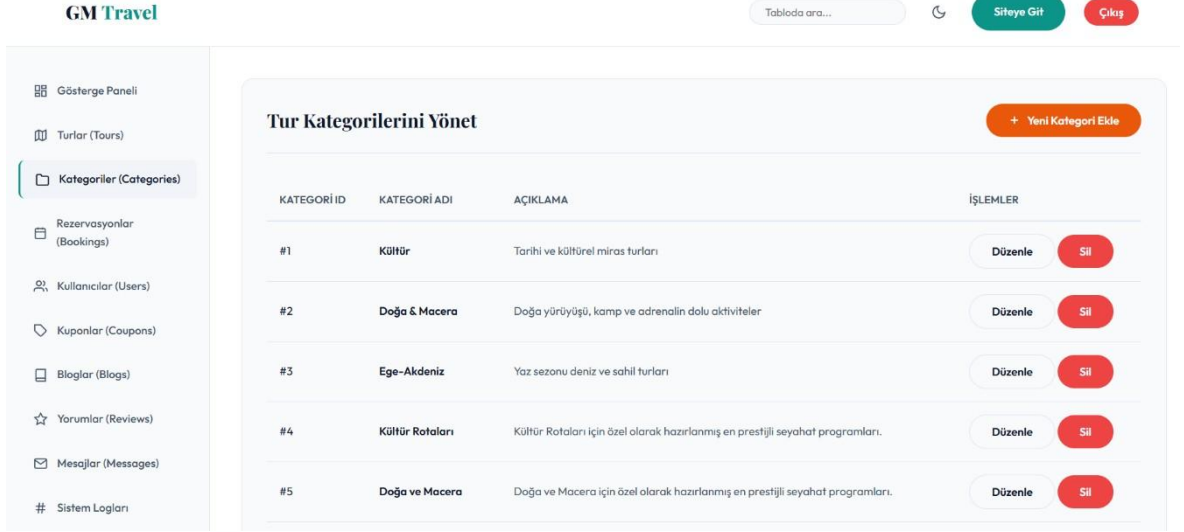
Çıkış

### Seyahat Rotalarını/Turlarını Yönet

+ Yeni Rota/Tur Ekle

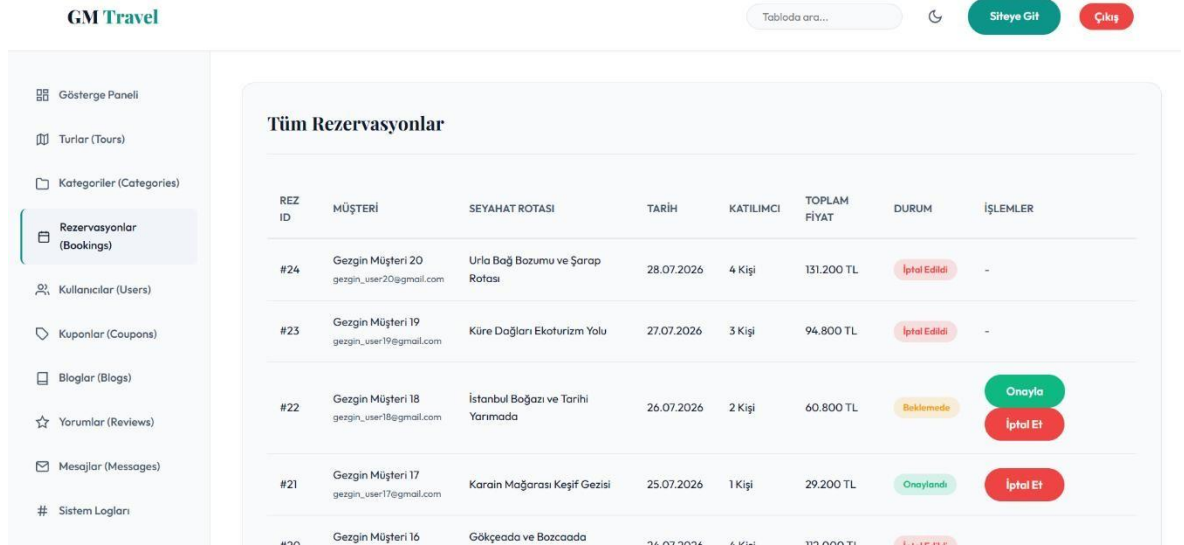
TUR ID	GÖRSEL	BAŞLIK	KONUM	FİYAT	KATEGORİ	REHBER	KONTENJAN	SÜRE	İŞLEMLER
#1		Kapatokya Balon Turu	Nevşehir, Türkiye	12.000 TL	Kültür	Caner Şen	4 Kişi	3 Gün	Düzenle Sil
#2		Likya Yolu Doğa Yürüyüşü	Muğla, Türkiye	18.000 TL	Doğa & Macera	Elif Kaya	14 Kişi	5 Gün	Düzenle Sil
#3		Göbeklitepe ve Şanlıurfa Kültür Turu	Şanlıurfa, Türkiye	10.000 TL	Kültür Rotaları	Mehmet Şahin	10 Kişi	3 Gün	Düzenle Sil
#4		Fırtına Deresi Rize Yaylaları	Rize, Türkiye	11.200 TL	Doğa ve Macera	Ayşe Çelik	11 Kişi	4 Gün	Düzenle Sil
#5		Marmaris Mavi Kayalar Yolculuğu	Muğla, Türkiye	12.400 TL	Mavi Yolculuk	Hakan Öztürk	12 Kişi	5 Gün	Düzenle Sil

Şekil 12. Tur Yönetim Paneli



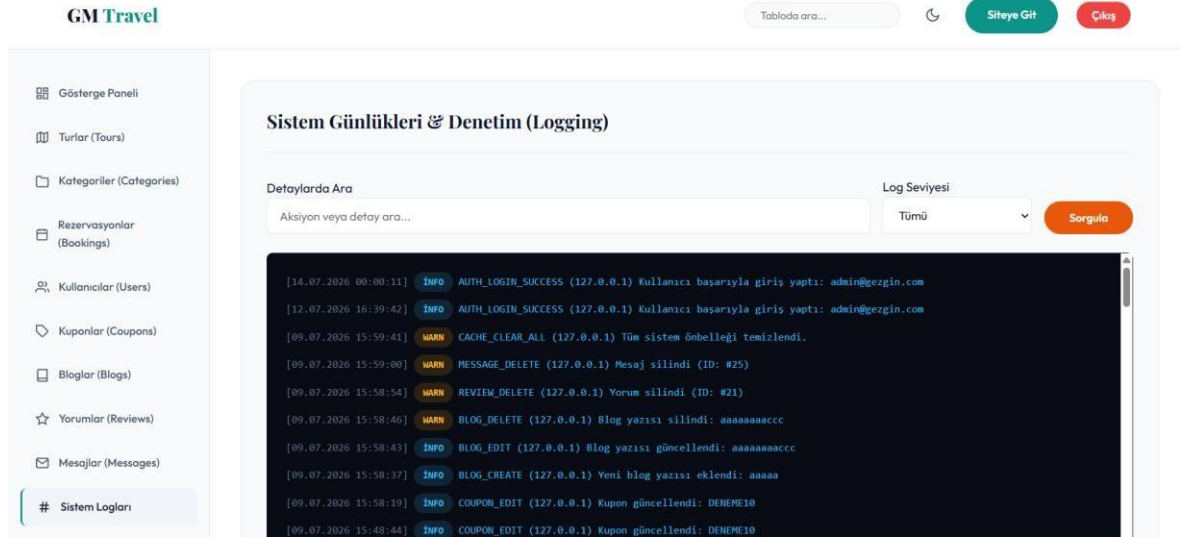
Şekil 13. Kategori Yönetim Paneli

- **Rezervasyon Moderasyonu:** Gelen seyahat rezervasyonları onaylanabilir veya iptal edilebilir. Rezervasyon iptal edildiğinde, sistem otomatik olarak tur kontenjanını iptal edilen kişi sayısı kadar artırarak kapasiteyi günceller.



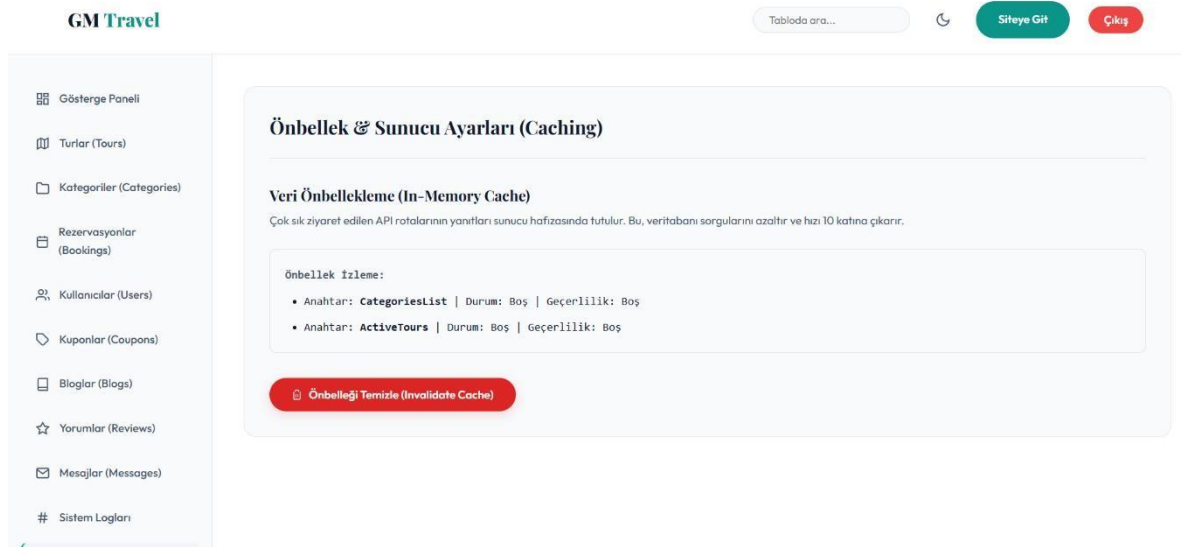
Şekil 14. Rezervasyon Moderasyon Paneli

- **Sistem Günlükleri (Audit Log):** Hangi yöneticinin hangi IP adresinden, hangi saatte ne tür bir işlem yaptığı veri tabanına kalıcı olarak kaydedilir ve Excel formatında dışa aktarılabilir.



Şekil 15. Sistem Denetim Günlükleri

- **Cache Yönetim Sayfası:** Admin panelindeki cache yönetim sekmesinden RAM üzerindeki tüm önbellek verileri tek tıklamayla silinerek veri tabanı ile anlık senkronizasyon zorlanabilir.



Şekil 16. Önbellek (Cache) Yönetim Paneli

### 3.3.3. Dinamik PDF Fatura ve Excel Raporlama Çıktıları

Sistem, kullanıcıların ve yöneticilerin veri çıktılarını alabilmesi için güçlü raporlama servisleri sunar:

- **PDF Fatura:** Kullanıcılar satın aldıkları rezervasyonların faturalarını profil ekranından PDF olarak indirebilirler. PDFsharp altyapısıyla Türkçe karakter hatası olmaksızın çizilen faturalar; seyahat detaylarını, katılımcı sayısını, ödeme tutarını ve doğrulama bilgilerini içerir.

- **Excel Raporu:** Yöneticiler tek bir tıklama ile tüm rezervasyon geçmişini veya sistem günlüklerini ClosedXML kütüphanesi yardımıyla oluşturulan gerçek `.xlsx` Excel tabloları olarak bilgisayarlarına indirebilirler.

#### 4. SONUÇ

Bu çalışma kapsamında, seyahat acentelerinin dijitalleşme ihtiyaçlarını karşılayan, modern mimari prensiplere uygun bir seyahat rezervasyon sistemi geliştirilmiştir. Projede kullanılan .NET 8.0/9.0 platformunun gücü, ilişkisel veri yönetiminde Entity Framework Core'un sağladığı kolaylıklar ve Dapper Micro-ORM'in yüksek performanslı okuma kabiliyeti ile birleştirilerek optimum bir veri erişim katmanı (hibrit yaklaşım) oluşturulmuştur.

Önbellekleme stratejileri (IMemoryCache) sayesinde sunucu üzerindeki sorgu yükü minimize edilmiş, PDFsharp ile Türkçe karakter uyumlu dinamik faturalandırılması ve ClosedXML ile Excel tabanlı raporlamalar yapılarak ticari bir yazılımın gereksinim duyduğu tüm çıktı modülleri başarıyla implemente edilmiştir. Güvenlik tarafında ASP.NET Core Identity altyapısı, anlık Gemini API destekli akıllı sohbet botu ve denetlenebilirlik tarafında geliştirilen Audit Log modülü, uygulamanın kurumsal standartlarda çalışmasını sağlamaktadır.

Sonuç olarak, geliştirilen "GM Seyahat Acentesi" projesi; esnek mimarisi, katmanlı yapısı ve kullanılan tasarım desenleri sayesinde gelecekteki yeni özelliklerin (örneğin gerçek bir ödeme geçidi entegrasyonu veya e-posta bildirim servisleri) kolayca eklenebileceği, kararlı ve sürdürülebilir bir altyapıya kavuşturulmuştur.

## 5. KAYNAKÇA

- [1] Microsoft Docs, "ASP.NET Core MVC Overview",  
<https://docs.microsoft.com/aspnet/core/mvc>
- [2] Microsoft Docs, "Entity Framework Core", <https://docs.microsoft.com/ef/core>
- [3] Dapper Tutorial, "Dapper Micro-ORM Documentation",  
<https://github.com/DapperLib/Dapper>
- [4] PDFsharp Documentation, "PDFsharp & MigraDoc Foundation",  
<https://www.pdfsharp.com>
- [5] ClosedXML GitHub, "ClosedXML - Easy Excel generation in .NET", <https://github.com/ClosedXML/ClosedXML>
- [6] Bootstrap Docs, "Bootstrap v5.3 Framework Reference", <https://getbootstrap.com>