

**ANKARA ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM 4531 PROJE RAPORU**

**SERVİS TAKİP SİSTEMİ**

**WEB TABANLI YÖNETİCİ PANELİ PROJE RAPORU**

**Mustafa Erdem ÖZTÜRK**

**21290480**

**Enver Bağcı**

**Ocak, 2026**

**Video Linki: <https://youtu.be/RdHXjmITZCc>**

## ÖZET

Bu çalışmada, servis taşımacılığı süreçlerinin yönetimini kolaylaştırmak amacıyla geliştirilen “Servis Takip Sistemi”nin web tabanlı yönetici paneli ele alınmıştır. Yönetici paneli; araç, şoför, rota ve yolcu yönetimi gibi operasyonel modülleri tek bir arayüzde toplayarak, sistemin merkezi biçimde yönetilmesini sağlamaktadır. Panel, web tarafında ASP.NET Web Forms mimarisiyle geliştirilmiş olup, arka planda bir API servisinden (REST) veri alarak çalışmaktadır. Kimlik doğrulama tarafında yönetici paneline yalnızca **yönetici rolüne sahip** kullanıcıların erişebilmesi hedeflenmiş; oturum ve yetkilendirme bilgilerinin istemci tarafındaki isteklerde “Bearer Token” üzerinden taşınması yaklaşımı benimsenmiştir.

Geliştirilen arayüz; Dashboard ekranı ile özet metrikleri sunmakta, Canlı Harita modülü ile araçların konumlarını anlık izlemeyi (**SignalR** ile gerçek zamanlı güncellemeler) mümkün kılmakta, Rotalar modülü ile rota yönetimini desteklemekte; rota oluşturma ekranında durak seçimi ve harita üzerinde rota önizlemesi sağlayarak operasyonel planlamayı kolaylaştırmaktadır. Araçlar ve Şoförler modülleri üzerinden temel CRUD işlemleri yapılabilmekte; Yolcular modülünde yolcu listesi görüntülenebilmekte ve yeni yolcu ekleme işlemi desteklenmektedir. Ayarlar bölümünde profil bilgilerinin görüntülenmesi ve şifre değiştirme işlevi için taslak bir arayüz sunulmuştur.

Sonuç olarak, web yönetici paneli; servis operasyonunu yönetmeye yönelik temel ekran ve veri akışlarını sunan, ölçeklenebilir biçimde genişletilebilecek bir yönetim aracı olarak tasarlanmış ve uygulanmıştır.

## İÇİNDEKİLER

|   |           |
|---|-----------|
| <b>ÖZET.....</b>  | <b>1</b>  |
| <b>İÇİNDEKİLER.....</b>   | <b>2</b>  |
| <b>1. GİRİŞ.....</b>  | <b>1</b>  |
| 1.1 Projenin Amacı.....   | 1         |
| 1.2 Projenin Kapsamı.....   | 1         |
| <b>2. MOTİVASYON VE KURAMSAL ARKA PLAN.....</b>                   | <b>2</b>  |
| 2.1 Proje Motivasyonu ve Operasyonel İhtiyaç.....                 | 2         |
| 2.2 Mevcut Yönetim Yöntemlerinin Yetersizlikleri.....             | 2         |
| 2.3 Web Tabanlı Yönetici Panelinin Rolü ve Katkıları.....         | 2         |
| 2.4 Gerçek Zamanlı Takip ve Anlık Karar Verme Mekanizmaları.....  | 2         |
| 2.5 Proje Kapsamının Operasyonel Planlamaya Entegrasyonu.....     | 3         |
| 2.6 Güvenlik, Yetkilendirme ve Oturum Yönetimi.....               | 3         |
| 2.7 Kuramsal Çerçeve ve Çalışmanın Konumlandırılması.....         | 3         |
| 2.8 Bölümün Genel Değerlendirmesi.....                            | 4         |
| <b>3. KULLANILAN TEKNOLOJİLER VE ARAÇLAR.....</b>                 | <b>4</b>  |
| 3.1 Web Uygulama Teknolojileri.....                               | 4         |
| 3.2 Arayüz ve UI Kütüphaneleri.....                               | 4         |
| 3.3 Harita ve Gerçek Zamanlı İletişim.....                        | 5         |
| <b>4. SİSTEM MİMARİSİ VE WEB PANELİN ROLÜ.....</b>                | <b>5</b>  |
| <b>5. WEB ARAYÜZ TASARIMI (UI/UX).....</b>                        | <b>5</b>  |
| 5.1 Genel Tasarım Yaklaşımı.....                                  | 6         |
| 5.2 Master Layout ve Navigasyon.....                              | 6         |
| 5.3 Kullanılabilirlik.....  | 6         |
| <b>6. MODÜLLER VE İŞLEVLER.....</b>                               | <b>6</b>  |
| 6.1 Giriş (Kimlik Doğrulama).....                                 | 6         |
| 6.2 Dashboard.....  | 7         |
| 6.3 Canlı Harita.....   | 7         |
| 6.4 Araç Yönetimi.....  | 8         |
| 6.5 Şoför Yönetimi.....   | 8         |
| 6.6 Rota Yönetimi.....  | 9         |
| 6.7 Rota Oluşturma ve Düzenleme.....                              | 9         |
| 6.8 Yolcu Yönetimi.....   | 10        |
| 6.9 Ayarlar.....  | 10        |
| <b>7. API ENTEGRASYONU VE VERİ AKIŞI.....</b>                     | <b>10</b> |
| <b>8. GÜVENLİK, YETKILENDİRME VE OTURUM YÖNETİMİ.....</b>         | <b>11</b> |
| 8.1 Oturum Yönetimi.....  | 11        |
| 8.2 Rol Tabanlı Erişim.....                                       | 11        |
| 8.3 İyileştirme Notları.....                                      | 11        |
| <b>9. HATA YÖNETİMİ, PERFORMANS VE İYİLEŞTİRME ÖNERİLERİ.....</b> | <b>11</b> |
| 9.1 Hata Yönetimi.....  | 11        |
| 9.2 Performans.....   | 11        |

|                                 |           |
|---------------------------------|-----------|
| 9.3 Önerilen Geliştirmeler..... | 12        |
| <b>11. SONUÇ.....</b>           | <b>12</b> |
| <b>Kaynaklar.....</b>           | <b>13</b> |

## **1. GİRİŞ**

Günümüzde servis taşımacılığı; okul servisleri, personel servisleri ve benzeri düzenli rotalı taşımacılık senaryolarında yoğun biçimde kullanılmaktadır. Bu süreçlerin etkin yönetimi; araçların takibi, rotaların planlanması, yolcu/şoför eşleştirme ve operasyonel durumun izlenmesi gibi birçok alt bileşeni kapsar. Operasyonun manuel yürütülmesi, bilgi dağınıklığına ve gecikmelere yol açabilmektedir. Bu nedenle dijital bir yönetim arayüzü ile tüm süreçleri tek noktadan yönetmek operasyon verimliliğini artırır.

Bu proje kapsamında geliştirilen Servis Takip Sistemi; mobil istemci(ler), API ve yönetim paneli bileşenleriyle çok katmanlı bir yapı sunmaktadır. Bu rapor, sistemin web yönetici paneli bileşenine odaklanır ve panelin sunduğu işlevleri, teknik altyapayı ve veri akışını detaylandırır

### **1.1 Projenin Amacı**

Web yönetici panelinin temel amacı; servis operasyonunun yönetici rolündeki kullanıcılar tarafından merkezi biçimde yönetilebilmesidir. Yönetici paneli, sahadan gelen konum verilerinin izlenmesi ve operasyonel varlıkların (araç, şoför, rota, yolcu) yönetimi için tasarlanmıştır.

### **1.2 Projenin Kapsamı**

- Yönetici girişi ve oturum yönetimi,
- Dashboard üzerinden özet metrikler ve günlük rota bilgileri,
- Canlı Harita üzerinde anlık araç takibi,
- Araç yönetimi (listeleme/ekleme/güncelleme/silme),
- Şoför yönetimi (listeleme/ekleme/güncelleme/silme + opsyonel araç atama),
- Rota yönetimi ve harita üzerinde rota görüntüleme,
- Harita tabanlı rota oluşturma (durak seçimi ve önizleme),
- Yolcu listeleme ve yeni yolcu ekleme,
- Ayarlar ekranı (profil bilgileri + şifre değiştirme akışı için taslak).

## **2. MOTİVASYON VE KURAMSAL ARKA PLAN**

### **2.1 Proje Motivasyonu ve Operasyonel İhtiyaç**

Günümüzde servis taşımacılığı operasyonları (okul, personel vb.) dinamik ve çok bileşenli süreçleri yönetmeyi gerektirmektedir. Bu süreçler; araç takibi, rota planlama, şoför-arac-yolcu eşleştirme ve anlık durum izleme gibi kritik görevleri içerir. Bu görevlerin manuel veya dağınık sistemlerle yürütülmesi, operasyonel verimsizliğe, gecikmelere ve hatalı bilgi akışına neden olmaktadır. Projenin temel motivasyonu, tüm bu operasyonel süreçleri dijital, merkezi ve verimli bir web tabanlı yönetici arayüzü ile tek noktadan yönetme ihtiyacından doğmuştur.

### **2.2 Mevcut Yönetim Yöntemlerinin Yetersizlikleri**

Geleneksel servis yönetimi yöntemleri; konum verilerini anlık izleyememe, rota değişikliklerini hızlıca sisteme yansıtamama ve operasyonel varlıkların (araç/şoför) kayıtlarını güncel tutmada zorlanma gibi yetersizliklere sahiptir. Bu durum, özellikle acil durumlarda veya yoğun saatlerde karar alma süreçlerini yavaşlatır. Ayrıca, yetkisiz erişim riskleri barındıran veya tutarsız veri girişine açık olan dağınık kayıt sistemleri, operasyonun güvenilirliğini düşürmektedir. Geliştirilen bu web paneli, bu yetersizlikleri ortadan kaldırmayı ve operasyonun merkezi bir API üzerinden beslenen güvenilir bir arayüz ile yönetilmesini hedeflemiştir.

### **2.3 Web Tabanlı Yönetici Panelinin Rolü ve Katkıları**

Web yönetici paneli, Servis Takip Sistemi'nin çok katmanlı mimarisinde yönetici rolündeki kullanıcılar için kritik bir arayüz görevi görür. Temel rolü, operasyonel varlık yönetimi (Araç, Şoför, Rota, Yolcu - CRUD işlemleri), Canlı Harita ile gerçek zamanlı konum verisi görselleştirmesi ve Dashboard ile özet metriklerin sunulmasıdır. Panel, arka plandaki API servisinden veri alarak çalışır; bu sayede iş kuralları ve veri güvenliği API katmanında merkezileşirken, panel sadece yönetim ve görselleştirme katmanı olarak hizmet eder. Bu yaklaşım, sistemin ölçeklenebilirliğini ve sürdürülebilirliğini artırmaktadır.

### **2.4 Gerçek Zamanlı Takip ve Anlık Karar Verme Mekanizmaları**

Sistemin "Canlı Harita" modülü ve SignalR teknolojisi entegrasyonu, operasyonun en önemli özelliklerindendir. Araç konumlarının anlık olarak harita üzerinde izlenmesi, operasyon yöneticisinin gecikmeleri veya rota dışı sapmaları saniyeler içinde fark etmesini sağlar. Bu anlık geri bildirim mekanizması, yöneticinin gerektiğinde hızlıca müdahale etmesini sağlayarak operasyonun kalitesini ve yolcu memnuniyetini doğrudan etkiler. Rota oluşturma ekranındaki harita tabanlı durak seçimi ve önizleme özellikleri de planlama aşamasında etkin karar vermeyi destekler.

## **2.5 Proje Kapsamının Operasyonel Planlamaya Entegrasyonu**

Proje kapsamında yer alan Rota Yönetimi, Rota Oluşturma, Araç Yönetimi ve Şoför Yönetimi modülleri, servis operasyonunun temel planlama ve yürütme süreçlerini dijitalleştirmektedir. Özellikle harita tabanlı rota oluşturma ekranı, durakların sürükle-bırak ile sıralanabilmesi ve rota önizlemesi sunması sayesinde karmaşık rota planlama sürecini görsel ve kullanıcı dostu hale getirir. Bu entegre yaklaşım, kağıt tabanlı veya manuel yöntemlere kıyasla operasyonel planlama hızını ve doğruluğunu önemli ölçüde artırmaktadır.

## **2.6 Güvenlik, Yetkilendirme ve Oturum Yönetimi**

Yönetici panelinin güvenliği, operasyon verilerinin hassasiyeti nedeniyle kritik öneme sahiptir. Giriş ekranı üzerinden sağlanan kimlik doğrulama sonrasında, API'den alınan Bearer Token'ın sonraki tüm isteklerde yetkilendirme bilgisi olarak taşınması esastır. Bu yaklaşım ile API ve web paneli arasında güvenli bir iletişim kanalı oluşturulmuştur. Ayrıca, sisteme erişim yalnızca "yönetici rolüne sahip" kullanıcılara sınırlanmış olup, rol tabanlı erişim kontrolü sayesinde operasyonel işlevlere sadece yetkili personelin ulaşması sağlanmıştır. Oturum yönetimi, token'in web oturumunda güvenli bir şekilde tutulmasıyla sağlanır.

## **2.7 Kuramsal Çerçeve ve Çalışmanın Konumlandırılması**

Bu çalışma, Bilgi Sistemleri (Yönetici Paneli), Dağıtık Sistemler (Mobil İstemci, API ve Web Panelden oluşan çok katmanlı mimari) ve Gerçek Zamanlı Veri İletişimi (SignalR) alanlarında konumlanmaktadır. Web Yönetici Paneli, bir REST API İstemcisi olarak görev yaparak Veritabanı ve İş Kuralları katmanından ayrılmıştır. Bu

mimari, sistemin her bir bileşeninin bağımsız olarak geliştirilebilmesi, bakımı yapılabilmesi ve ölçeklenebilmesi için modern yazılım mühendisliği prensiplerine uygun bir çerçeve sunmaktadır.

## **2.8 Bölümün Genel Değerlendirmesi**

Bu bölümde ele alınan motivasyon ve kuramsal arka plan, web tabanlı yönetici panelinin, Servis Takip Sistemi'nin merkezi bir yönetim aracı olarak operasyonel verimliliği ve güvenilirliği artırmayı amaçladığını göstermektedir. Proje, sadece bir arayüz geliştirmekle kalmamış, aynı zamanda servis taşımacılığı operasyonlarının zorluklarına modern teknolojilerle (ASP.NET Web Forms, Google Maps API, SignalR) çözüm sunan bütüncül bir yönetim katmanı olarak tasarlanmıştır.

# **3. KULLANILAN TEKNOLOJİLER VE ARAÇLAR**

## **3.1 Web Uygulama Teknolojileri**

- **ASP.NET Web Forms (.NET Framework 4.8):** Yönetici panelinin temel geliştirme çatısıdır. Master Page yaklaşımıyla ortak layout ve menü yapısı sağlanmıştır.
- **C#:** Sunucu tarafı code-behind geliştirmeleri.
- **HTTP Client:** API isteklerini gerçekleştirmek için kullanılmıştır.
- **JSON (Newtonsoft.Json):** API'den gelen/verilen JSON içeriklerin serileştirilmesi/ayrıştırılması.

## **3.2 Arayüz ve UI Kütüphaneleri**

- **Bootstrap 5:** Responsive tasarım ve bileşenler (tablo, modal, buton vb.).
- **Bootstrap Icons:** Arayüz ikonları
- **SortableJS:** Rota oluşturma ekranında seçilen durakların sürükle-bırak sıralanması.

### **3.3 Harita ve Gerçek Zamanlı İletişim**

- **Google Maps JavaScript API:** Harita üzerinde durak/araç marker'ları ve rota çizimleri.
- **SignalR JavaScript Client:** Canlı Harita sayfasında konum güncellemelerini gerçek zamanlı alabilmek için.

## **4. SİSTEM MİMARİSİ VE WEB PANELİN ROLÜ**

Servis Takip Sistemi çok bileşenli bir yapıda kurgulanmıştır:

- **API Katmanı (Backend):** İş kuralları, veritabanı işlemleri ve güvenlik kontrollerinin merkezi noktasıdır. Web panel API'yi tüketir.
- **Web Yönetici Paneli:** Yönetici kullanıcıların sisteme giriş yapıp operasyonu yönetmesini sağlar. Panelde yapılan işlemler API'ye HTTP istekleri ile iletılır.
- **Düzenleyici:** Mobil/web istemci(ler) veya sürücü uygulaması gibi bileşenler, konum ve operasyon verisi üretebilir.

Bu mimaride web panelin görevi; yönetimsel işlemleri bir arayüz üzerinden kolaylaştırmak ve API verilerini görselleştirmektir. Panel, veriyi saklayan/şüreklendiren ana yer değil; API ile entegre çalışan bir yönetim katmanıdır

## **5. WEB ARAYÜZ TASARIMI (UI/UX)**

## **5.1 Genel Tasarım Yaklaşımı**

Panelin arayüzü, yönetim uygulamalarında sık kullanılan “sol menü + ana içerik” yaklaşımıyla tasarlanmıştır. Sol menüde modüller listelenmiş, sağ tarafta seçilen modülün içeriği görüntülenmiştir. Arayüzde Bootstrap kullanılarak modern bir görünüm ve responsive uyum hedeflenmiştir.

## **5.2 Master Layout ve Navigasyon**

Panelde ortak bileşenler (sidebar, kullanıcı alanı, çıkış butonu) tek bir Master Page üzerinden yönetilmiştir. Böylece her sayfada tekrar eden HTML/CSS kodları minimize edilmiş, tutarlı bir kullanıcı deneyimi sağlanmıştır.

## **5.3 Kullanılabilirlik**

- Tablo tabanlı listelerde hızlı tarama ve işlem butonları (düzenle/sil) sunulmuştur.
- Boş liste durumlarında kullanıcıya açıklayıcı mesaj ve ikon gösterimi yapılmıştır.
- Canlı Harita sayfasında araç listesi ile harita etkileşimi desteklenerek operatörün hızlı takip yapabilmesi amaçlanmıştır.

# **6. MODÜLLER VE İŞLEVLER**

Bu bölümde, sistem tasarıımı aşamasında belirlenen mimari yapı ve bileşenlerin yazılım ortamında nasıl hayatı geçirildiği açıklanmaktadır. Gerçekleştirme süreci, modüler ve sürdürülebilir bir yazılım geliştirme yaklaşımı ile yürütülmüştür.

## **6.1 Giriş (Kimlik Doğrulama)**

Panel erişimi, bir giriş ekranı üzerinden sağlanır. Kullanıcı e-posta ve şifre ile giriş yapar; sistem API'ye giriş isteği gönderir. API başarılı dönerse token ve kullanıcı bilgileri oturumda saklanır. Ek olarak rol kontrolü yapılarak yalnızca “yönetici” rolündeki kullanıcıların panele erişmesine izin verilir.

Bu yaklaşım ile:

- Yönetici paneli ayrı bir güvenlik sınırı oluşturur,

- API'den alınan token ile sonraki isteklerde yetkilendirme sağlanır,
- Kullanıcı bilgileri sidebar'da görüntülenebilir.

## **6.2 Dashboard**

Dashboard ekranı; anlık operasyon görünürlüğü sağlamak için özet metrikler ve “bugünün rotaları” tablosu içerir. Örnek metrikler:

- Aktif araç sayısı,
- Gecikmeli araç sayısı,
- Tamamlanan rotalar,
- Doluluk oranı (örnek/placeholder değerler).

Dashboard, API'den “bugünün rotaları” bilgisini çekerek tabloya basar. API'ye erişilemezse örnek veriyle sayfa boş kalmayacak şekilde bir fallback yaklaşımı uygulanmıştır.

## **6.3 Canlı Harita**

Canlı Harita modülü, sistemin en kritik görselleştirme bileşenlerinden biridir. Bu sayfa:

- Harita üzerinde araç konumlarını marker ile gösterir,
- Sağ tarafta aktif araçların listesini, durumlarını ve son güncellenme zamanlarını listeler,
- SignalR üzerinden gerçek zamanlı konum güncellemelerini dinler,
- Araç offline olduğunda marker'ı kaldırabilir,
- (Varsa) belirli bir rota için “routeld” parametresi ile rota/durak bilgisini haritada göstermeye yönelik bir akış barındırır.

Bu sayede operasyon yöneticisi, araçların sahadaki hareketini canlı izleyebilir; gecikme veya rota dışı durumları daha hızlı fark edebilir.

#### **6.4 Araç Yönetimi**

Araçlar modülü; araçların listelenmesi ve temel CRUD işlemlerini sağlar:

- Araç ekleme,
- Araç güncelleme,
- Araç silme,
- Araç aktif/pasif durumu yönetimi.

Arayüzde modal form kullanılarak yeni araç ekleme ve güncelleme işlemi tek ekrandan yürütülür. Form; plaka, marka-model birleşimi, kapasite ve aktiflik gibi alanlar içerir.

#### **6.5 Şoför Yönetimi**

Şoförler modülü, şoför kayıtlarının yönetimini sağlar:

- Şoför listesi,
- Şoför ekleme/güncelleme/silme,
- Opsiyonel olarak araca atama (dropdown ile araç seçimi).

Bu modül, API'den hem şoför listesini hem de araç listesini alarak “atanacak araç” seçeneğini kullanıcıya sunar. Böylece yönetici, şoför–araç eşleştirmesini arayüzden yönetebilir.

#### **6.6 Rota Yönetimi**

Rotalar modülü; rota listesini ve rota durum bilgisini sunar. Yönetici bu ekrandan:

- Yeni rota oluşturma ekranına gidebilir,

- Bir rotayı harita üzerinde görüntüleyebilir,
- Rota düzenleme sayfasına yönlenebilir,
- Rota silme işlemini başlatabilir.

Liste ekranında rota adı, araç, şoför, başlangıç/bitiş saatleri, durak sayısı ve aktiflik durumu gibi alanlar gösterilerek operasyonun hızlıca izlenmesi hedeflenmiştir.

### **6.7 Rota Oluşturma ve Düzenleme**

Rota oluşturma ekranı, harita tabanlı bir planlama deneyimi sunar:

- Sol tarafta tüm duraklar listelenir ve arama yapılabilir,
- Orta alanda harita üzerinde duraklar ve seçilen duraklar görselleştirilir,
- Sağ tarafta seçilen duraklar listelenir ve sürükle-bırak ile sıralanabilir,
- Araç seçimi ve başlangıç saatı gibi rota meta verileri girilir,
- Rota önizlemesi haritada çizilir; çoklu durak için çizgisel rota gösterimi yapılır,
- Directions API ile gerçek yol çizimi (rota çizimi) desteklenen yapı mevcuttur.

Bu modül, rota planlamanın en kritik kısmını görsel hale getirerek kullanıcı hatalarını azaltır ve planlama hızını artırır.

### **6.8 Yolcu Yönetimi**

Yolcular modülü; yolcu listesini görüntüler ve yeni yolcu ekleme işlemi sunar. Yolcu ekleme formunda ad-soyad, telefon, e-posta ve adres gibi alanlar hedeflenmiştir. Sunucu tarafında yolcu ekleme isteği API'ye iletılır.

Not: Uygulama mantığında yolcu rolü ayrı bir role işaret eder ve yeni yolcu oluşturma isteğinde rol bilgisi bu doğrultuda gönderilir.

### **6.9 Ayarlar**

Ayarlar ekranı, yönetici profil bilgilerini (ad, e-posta, rol) gösterir. Ayrıca şifre değiştirme formu bulunmaktadır.

## 7. API ENTEGRASYONU VE VERİ AKIŞI

Web panelde temel veri akışı şu şekildedir:

1. Yönetici giriş yapar → API'ye login isteği gider → token alınır.
2. Token, web oturumunda saklanır.
3. Panel sayfaları açıldığında API'ye token ile istek atılır:
  - Dashboard: günün rotaları ve metrik hesapları
  - Canlı Harita: aktif konumlar ve gerçek zamanlı güncellemeler
  - Araç/Şoför/Rota/Yolcu ekranları: listeleme ve CRUD
4. Bazı sayfalarda istemci tarafı fetch istekleri de kullanılır (özellikle rota oluşturma gibi dinamik ekranlarda).

Bu yaklaşım, web panelin API'ye bağımlı bir “yönetim istemcisi” olduğunu gösterir. Dolayısıyla API erişilebilirliği ve yetkilendirme doğruluğu, panelin doğru çalışmasının temel şartıdır.

## 8. GÜVENLİK, YETKİLENDİRME VE OTURUM YÖNETİMİ

### 8.1 Oturum Yönetimi

Panelde kullanıcı oturumu, giriş sonrası token'ın session'da tutulması ile yönetilmektedir. Master layout seviyesinde oturum kontrolü yapılmakta ve giriş yapılmamışsa login ekranına yönlendirme uygulanmaktadır.

### 8.2 Rol Tabanlı Erişim

Login sonrası kullanıcı rolü kontrol edilerek yalnızca yönetici rolüne sahip kullanıcıların panele erişimi hedeflenmiştir. Böylece normal kullanıcıların yönetici paneline erişmesi engellenir.

### **8.3 İyileştirme Notları**

Geliştirme sürecinde sık görülen bazı güvenlik ve sürdürülebilirlik notları:

- API adresinin bazı sayfalarda sabit (hard-coded) kullanılması yerine tek bir konfigürasyondan okunması tercih edilmelidir.
- Harita anahtarı gibi hassas anahtarlar raporda gösterilmemeli, ortam değişkeni/secret yönetimi ile korunmalıdır.
- Şifre değiştirme ve yolcu oluşturma gibi işlemler, gerçek sisteme mutlaka güçlü doğrulama ve güvenlik politikaları ile desteklenmelidir (ör. varsayılan şifre yaklaşımı yerine davet/şifre oluşturma akışı).

## **9. HATA YÖNETİMİ, PERFORMANS VE İYILEŞTİRME ÖNERİLERİ**

### **9.1 Hata Yönetimi**

Bazı modüllerde API çağrıları try/catch ile çevrelenmiş ve hata durumunda kullanıcı deneyimini bozmayacak “boş liste” veya “örnek veri” gösterimi uygulanmıştır. Bu yaklaşım demo/sunum için avantajlıdır; ancak üretimde kullanıcıya daha anlamlı hata mesajları ve loglama mekanizması eklenmelidir.

### **9.2 Performans**

- Liste ekranlarında sayfalama/filtreleme henüz bulunmamaktadır; veri büyündüğünde performans etkilenebilir.
- Canlı Harita tarafında marker güncelleme ve DOM güncellemeleri sık gerçekleşebileceğ için, yoğun senaryolarda optimizasyon (throttling/debouncing) gerekebilir.

### **9.3 Önerilen Geliştirmeler**

- Tüm API URL yönetimini tek merkezden yapmak,
- Ortak bir API istemci katmanı yazıp tekrarları azaltmak,

- Yetkilendirme hatalarında otomatik logout/yeniden login yönlendirmesi,
- Yolcu modülüne düzenleme/silme işlemlerini eklemek,
- Loglama ve izleme (ör. merkezi log) eklemek.

## **11. SONUÇ**

Bu raporda Servis Takip Sistemi'nin web tabanlı yönetici paneli incelenmiştir. Panel; operasyon yönetimi için gerekli temel modülleri sağlamaktadır: yönetici giriş akışı, dashboard özetleri, canlı harita ile anlık takip, araç/şoför/rota/yolcu yönetimi ve rota oluşturma. API tabanlı mimari sayesinde web panel, veri yönetimi ve iş kurallarını backend servise bırakarak daha sürdürülebilir ve genişletilebilir bir yapı sunmaktadır.

## **Kaynaklar**

1. Microsoft Docs – ASP.NET Web Forms
2. Microsoft Docs – HttpClient Kullanımı
3. Microsoft Docs – SignalR (JavaScript Client)
4. Google Maps Platform – Maps JavaScript API
5. Bootstrap 5 Dokümantasyonu
6. Newtonsoft.Json Dokümantasyonu