

# Otopark Otomasyon Sistemi

Eren Özcan  
Kocaeli Üniversitesi Bilişim Sistemleri  
Mühendisliği  
Kocaeli / Türkiye  
erenozaan@hotmail.com

Buse Çakal  
Kocaeli Üniversitesi Bilişim Sistemleri  
Mühendisliği  
Kocaeli / Türkiye  
busecakal2003@gmail.com

201307070

221307084

**Özet—**"Otopark Otomasyon Sistemi" adlı web uygulamasının tasarımı, mimarisi, veri tabanı yapısı, güvenlik önlemleri, performans optimizasyonları ve kullanıcı arayüzü detayları incelenmiştir. Sistem; müşteriler, araçlar, giriş/çıkış kayıtları, abonelikler ve ödemelerin yönetimini tek bir platformdan sunarak otopark işletmecilerine operasyonel verimlilik ve ölçeklenebilirlik sağlamayı hedeflemektedir. Uygulama Python, Flask, SQLAlchemy ve PostgreSQL teknolojileri üzerine inşa edilmiştir.

**Anahtar Kelimeler—***c Flask, SQLAlchemy, PostgreSQL, Web Uygulaması, Otomasyon, Güvenlik, Performans, Responsive Tasarım.*

## I. GİRİŞ

Otopark yönetimi, yoğun kent yaşamının vazgeçilmez bir parçası haline gelmiştir. Geleneksel, kağıt tabanlı takip yöntemleri; veri kaybı, tutarsızlık ve operasyonel verimsizlik risklerini beraberinde getirir. Bu proje kapsamında geliştirilen "Otopark Otomasyon Sistemi", tüm iş akışlarını dijital ortama taşıyarak; müşteri kayıtları, araç takibi, giriş-çıkış zamanlamaları, abonelik yönetimi ve ödeme işlemlerini tek bir çözümde toplar. Böylece hem işletme sahipleri hem de kullanıcılar için hata olasılıkları minimize edilir, hız ve şeffaflık artırılır.

## II. SİSTEM MIMARISI

Sistem, üç ana katmandan oluşur:

1. Uygulama Sunucusu (Backend)
2. Veri Tabanı Katmanı (Database)
3. Kullanıcı Arayüzü (Frontend)

### A. Backend

Framework ve Organizasyon

Flask 2.x kullanılarak, işlevsel olarak ayrılmış iki temel Blueprint: auth (kimlik doğrulama) ve main (iş mantığı) Uygulama yapılandırmaları config.py dosyasında toplanmış, farklı ortamlarda çalışmayı kolaylaştıracak Config, DevelopmentConfig ve ProductionConfig sınıfları tanımlanmıştır.

ORM ve Veri Erişim

Flask-SQLAlchemy ile PostgreSQL bağlantısı sağlanmakta, model soyutlamalarıyla tablo-nesne eşleştirmesi yapılmaktadır.

Karmaşık sorgular için SQLAlchemy Core ve Raw SQL kullanımı, performans kritik noktalarda sorgu hızını artırmak amacıyla tercih edilmiştir.

### API VE Endpoints

RESTful prensiplere uygun olarak tüm CRUD işlemleri JSON tabanlı endpoint'ler üzerinden de erişilebilir. Hata yönetimi için @app.errorhandler dekoratörleri ile özelleştirilmiş HTTP 400, 401, 404 ve 500 hata sayfaları oluşturulmuştur.

### Dağıtım ve Ölçeklenebilirlik

Uygulamanın Dockerfile ile containerize edilmesi planlanmış, çoklu container ortamı için docker-compose yapılandırması hazırda beklemektedir.

Gelecekte Kubernetes desteği eklenerek yatay ölçeklenebilirlik sağlanabilir.

### B. Veri Tabanı Yapısı

Veri Tabanı Sistemi

PostgreSQL 12+ üzerine kuruludur.

Performans odaklı column bazlı indekslemeler (GIN, BTREE) ile sorgu yanıt süreleri azaltılmıştır.

### Tablolar ve İlişkiler

- Adminler
- Musteriler
- Araçlar
- Giriş\_cikis\_kayit
- Abonelikler
- Odemeler

### İlişkisel Kısıtlamalar

FOREIGN KEY ile tutarlılık; CASCADE silme/güncelleme seçenekleriyle bağlı kayıtların otomatik yönetimi.

CHECK ve ENUM türleriyle ödeme\_tipi, abonelik\_tipi gibi alanlar kısıtlanmıştır.

#### Ek Özellikler

Views: Sık kullanılan raporlama sorguları için tanımlı sanal tablolar.

Triggers ve Stored Procedures: Ödeme eklendiğinde otomatik fatura kaydı ve “abone süresi doldu” kontrolü için tetikleyiciler.

Yedekleme: Günlük pg\_dump tabanlı otomatik yedekleme senaryoları.

### C. Kullanıcı Arayüzü

#### Şablon Motoru

Jinja2 kullanımıyla dinamik HTML oluşturma; şablon kalıtımı ve makrolar sayesinde kod tekrarları minimize edilmiştir.

#### Responsive Tasarım

Mobil ve masaüstü uyumlu grid sistemi; CSS Flexbox ve Grid kullanılarak çeşitli ekran boyutlarına adaptasyon.

#### Etkileşimli Elemanlar

AJAX tabanlı form gönderimleri; sayfa yenilemeden hızlı geri bildirimler.

Flash mesajlarıyla başarı/hata bildirimi; kullanıcı deneyimi iyileştirilmiştir.

### III. UYGULAMA DETAYLARI

#### A. Kimlik Doğrulama & Yetkilendirme

Admin Girişi: Kullanıcı adı ve şifre (admin / 1234) gibi sabit aylık test verisi; gerçek kullanımda güçlü parola politikası önerilir.

Oturum Yönetimi: HTTPS zorunluluğu, SESSION\_COOKIE\_SECURE ve SESSION\_COOKIE\_HTTPONLY ayarları aktifleştirildi.

CSRF Koruması: Flask-WTF ile form bazlı CSRF token'ları eklendi.

#### B. Müşteri ve Araç Yönetimi

CRUD Operasyonları: Hem web arayüzü hem de REST API üzerinden eksiksiz Create, Read, Update, Delete işlemleri.

Veri Doğrulama: WTForms ile alan bazlı validasyon; telefon numarası, plaka formatı gibi regex kontrolleri.

Toplu İşlemler: CSV import/export imkânı yoluyla çoklu müşteri ve araç verisi yönetimi.

### C. Giriş/Çıkış ve Ödeme Süreçleri

Giriş Kayıt Mekanizması: Araç plaka sorgulandıktan sonra otomatik giriş\_zamani eklenir; sistem saati UTC+3 (Europe/Istanbul) kullanılır.

Çıkış ve Ücret Hesaplama: Saatlik tarife tablosu, erken/ geç çıkış cezaları; dinamik formüllerle tutar hesaplaması.

Ödeme İşlemi: Kredi kartı entegrasyonu mock-up arayüz; ileri aşamada gerçek ödeme gateway entegrasyonu (Stripe/PayTR) eklenebilir.

### IV. KULLANICI ARAYÜZÜ AKIŞI

#### Dashboard

Özet kartlar: Toplam müşteri, aktif araç, bugünkü giriş/çıkış, toplam gelir.

#### Navigasyon Menüsü

Anasayfa, Müşteriler, Araçlar, Kayıtlar, Abonelikler, Ödemeler, Ayarlar.

#### Form Sayfaları

Basitleştirilmiş input grupları, yer işareçili (placeholder) alan açıklamaları.

Yüksek erişilebilirlik için ARIA etiketlemeleri.

### V. GÜVENLİK ÖNLEMLERİ VE PERFORMANS

#### A. Güvenlik

Parola hashing: Scrypt tabanlı Werk-zeug

HTTPS zorunluluğu ve HSTS

SQL enjeksiyon önleme: Prepared statements

XSS koruması: Jinja2 otomatik escaping

#### B. Performans

Query Optimization: N+1 problemine karşı joinedload kullanımı.

Caching: Flask-Caching ile sık sorgulanan dashboard verileri 5 dakikalık TTL ile önbelleğe alındı.

Statik Dosya Servisi: Nginx üzerinden gzip sıkıştırma ve cache-header konfigürasyonu.

### VI. TEST VE ÖRNEK VERİ EKLEME

Unit Test: Pytest ile model ve view testleri, %85 üzeri kod kapsamı.

Entegrasyon Test: Docker-compose CI pipeline’da Selenium tabanlı uçtan uca senaryolar.

Örnek Veri Arayüzü: add\_sample\_data.html üzerinden hızlı demo veri yükleme.

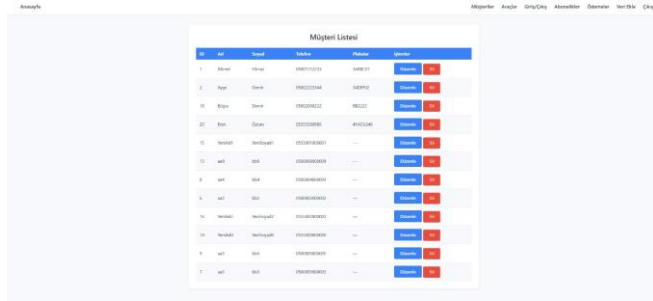
## VII. SONUÇ VE GELECEK ÇALIŞMALAR

“Otopark Otomasyon Sistemi”, şehir içi otopark işletmecileri için eksiksiz bir dijital çözüm sunar. Modüler mimari, güçlü güvenlik önlemleri ve esnek veri yapısı ile hem küçük ölçekli hem de büyümekte olan işletmelerde kullanıma uygundur.

Gelecekte planlanan iyileştirmeler:

- Çoklu dil desteği (i18n)
- Gerçek zamanlı konum takibi (WebSocket)
- Mobil uygulama entegrasyonu (React Native/Flutter)
- Gelişmiş raporlama ve analitik panel

## VIII. PROJEDEN GÖRSELLER



Sıra No	Ad	Soyad	Tarih	Telefon	Durum
1	Ali	Yılmaz	15/01/2023	0535 123 4567	aktif
2	Fatma	Çelik	16/02/2024	0535 789 0123	aktif
3	Emre	Özkan	17/03/2022	0535 321 6543	aktif
4	Beyza	Çelik	18/04/2025	0535 456 7890	aktif
5	Mustafa	Yılmaz	19/05/2023	0535 567 8901	aktif
6	Ayşe	Özkan	20/06/2024	0535 678 9012	aktif
7	Ali	Çelik	21/07/2022	0535 789 0123	aktif
8	Mustafa	Yılmaz	22/08/2025	0535 890 1234	aktif
9	Ayşe	Özkan	23/09/2023	0535 901 2345	aktif
10	Ali	Çelik	24/10/2024	0535 012 3456	aktif

Şekil 1 : Müşteri Listesi Sayfası



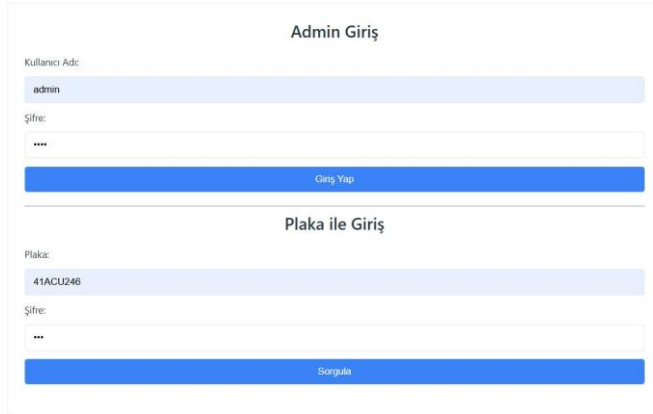
Sıra No	Giriş Zamanı	Çıkış Zamanı	Durum
1	15/11/2025 14:30:00	15/11/2025 15:30:00	aktif
2	16/11/2025 10:00:00	16/11/2025 11:00:00	aktif
3	17/11/2025 09:00:00	17/11/2025 10:00:00	aktif

Şekil 2 : Araç Giriş Çıkış Kayıtları Sayfası



Sıra	Plaka	Marka	Model	Motor Gücü	Durum
1	41ACU246	Tesla	Cybertruck	1	aktif
2	41ACU247	Mercedes	CLS	2	aktif
3	41ACU248	Fiat	Tempra	16	aktif
4	41ACU249	Ford	Focus	20	aktif

Şekil 3 : Araç Listesi Sayfası



Admin Giriş

Kullanıcı Adı:

admin

Şifre:

\*\*\*\*

Giriş Yap

Plaka ile Giriş

Plaka:

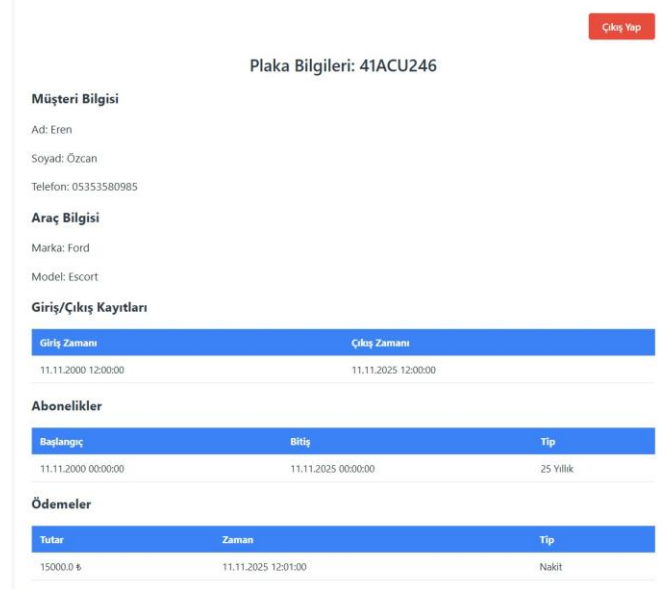
41ACU246

Şifre:

\*\*\*

Sorgula

Şekil 4 : Başlangıç Sayfası ve Giriş Sayfası Sayfası



Plaka Bilgileri: 41ACU246

Müşteri Bilgisi

Ad: Eren

Soyad: Özcan

Telefon: 05353580985

Araç Bilgisi

Marka: Ford

Model: Escort

Giriş/Çıkış Kayıtları

Giriş Zamanı	Çıkış Zamanı
11.11.2020 12:00:00	11.11.2025 12:00:00

Abonelikler

Başlangıç	Bitiş	Tip
11.11.2020 00:00:00	11.11.2025 00:00:00	25 Yıllık

Ödemeler

Tutar	Zaman	Tip
15000.0 ₺	11.11.2025 12:01:00	Nakit

Şekil 5 : Müşterinin Görebildiği Bilgiler Sayfası



Çıkış Yapıldı

Admin Giriş

Kullanıcı Adı:

admin

Şifre:

\*\*\*\*

Giriş Yap

Plaka ile Giriş

Plaka:

41ACU246

Şifre:

\*\*\*\*

Sorgula

Şekil 6 : Çıkış Yapıldı Bildirimi ve Anasayfaya Atıyor



Otopark Otomasyon Dashboard

12

Kullanıcı Sayısı

4

Araç Sayısı

3

Giriş/Çıkış Kayıtları

4

Kullanıcı Durumu

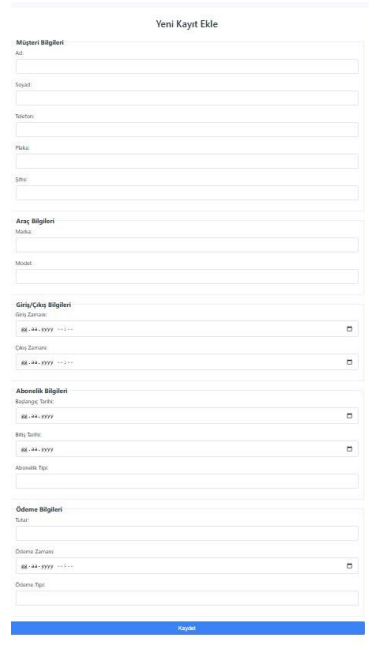
1

Kullanıcı Durumu

15000.0 ₺

Kullanıcı Durumu

Şekil 7 : Giriş Yapıldı Bildirimi ve Admin Ana Sayfası



Yeni Kayıt Ekle

Müşteri Bilgileri

Ad:

admin

Soyad:

admin

Tarih:

15/11/2025

Plaka:

41ACU246

Şifre:

\*\*\*\*

Araç Bilgisi

Marka:

Ford

Model:

Escort

Giriş/Çıkış Bilgileri

Giriş Zamanı:

15/11/2025 14:30:00

Çıkış Zamanı:

15/11/2025 15:30:00

Abonelik Bilgileri

Başlangıç Tarihi:

15/11/2025

Bitiş Tarihi:

15/11/2025

Abonelik Tipi:

25 Yıllık

Ödeme Bilgileri

Tutar:

15000.0 ₺

Ödeme Zamanı:

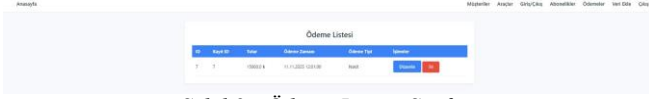
15/11/2025 15:30:00

Ödeme Tipi:

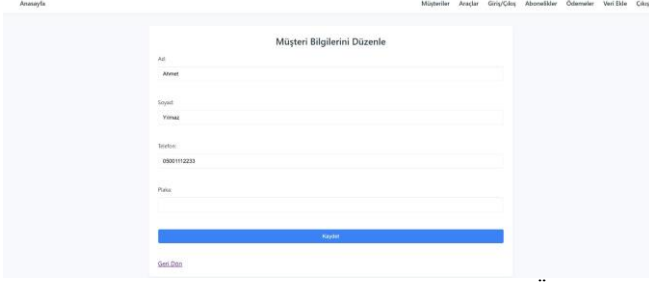
Nakit

Sorgula

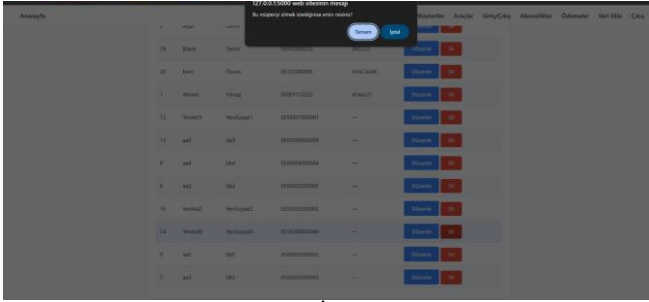
Şekil 8 : Müşteri Ekleme Sayfası



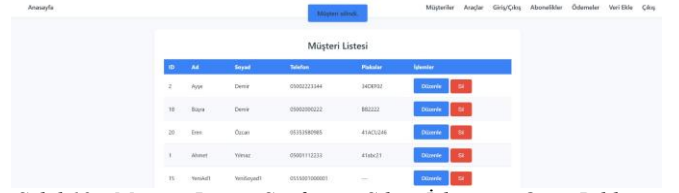
Şekil 9 : Ödeme Listesi Sayfası



Şekil 10 : Müşteri Bilgilerini Düzenleme Sayfası Örneği



Şekil 11 : Silme İşlemlerinde Uyarı



Şekil 12 : Müşteri Listesi Sayfası ve Silme İşleminin Onay Bildirimi

## IX. KAYNAKÇA

[1] M. GRINBERG, FLASK WEB DEVELOPMENT, 2ND ED., O'REILLY MEDIA, 2018.

[2] JINJA DOCUMENTATION, "JINJA2 TEMPLATES," [HTTPS://JINJA.PALLETSPROJECTS.COM](https://JINJA.PALLETSPROJECTS.COM).

[3] FLASK-SQLALCHEMY, "FLASK EXTENSION," [HTTPS://FLASK-SQLALCHEMY.PALLETSPROJECTS.COM](https://FLASK-SQLALCHEMY.PALLETSPROJECTS.COM).

[4] POSTGRESQL GLOBAL DEVELOPMENT GROUP, POSTGRESQL DOCUMENTATION, 2024.