



# **T.C. ERZURUM TEKNİK ÜNİVERSİTESİ**

Mühendislik ve Mimarlık Fakültesi

Bilgisayar Mühendisliği Bölümü

YAZILIM MÜHENDİSLİĞİ DERSİ

*AÇIK ARTIRMA WEB SİTESİ*

HAZIRLAYANLAR

**Mehmet Ali NACAK**

**Emre MERAL**

**Yusuf YÜCEL**

**Ömer ÇELİK**

**Dilara ADIGÜZEL**

**Osman DEMİR**

**Muhammed Cenab KÖROĞLU**

**Serhat Eren ATALAY**

## İçindekiler Tablosu

1.Şekiller Listesi.....	3
2.Giriş.....	5
2.1.Amaç.....	5
2.2.Kapsam.....	5
2.3.Hedef Kitle.....	6
2.4.Problem Tanımı.....	7
2.5.Proje Ekibi.....	7
3.Tanımlar Listesi.....	8
4.Kullanılan Platformlar.....	9
4.1. Proje Yönetim Süreci ve Görev Takibi – Trello.....	9
4.2.Visual Studio – Back-end Geliştirme.....	9
4.3.Visual Studio – Front End Geliştirme.....	9
4.4.Veritabanı Yönetimi – Microsoft SQL Server (MSSQL).....	10
4.5.Sürüm Kontrolü ve İş Birliği -GitKraken .....	11
5.Yazılım Geliştirme ve Katmanlar.....	12
5.1. Katmanlı Mimari.....	12
5.2.API Dokümantasyonu – Swagger Kullanımı.....	13
5.3.Güvenlik Önlemleri.....	13
6.Proje Zaman Çizelgesi ve Sprint.....	14
6.1. Proje Genel Zaman Aralığı.....	14
6.2.Sprint Planlaması ve İçerik.....	14
6.3.Günlük Scrum Toplantılar.....	18
6.4.Backloglar.....	18
6.5.Definiton of Done.....	19

# İÇİNDEKİLER

7.Kullanıcı Senaryoları.....	22
7.1. Senaryo 1.....	22
7.2. Senaryo 2.....	23
7.3. Senaryo 3.....	24
8.Veritabanı Tasarımı ve Kimlik.....	25
8.1. Ana Varlıklar ve Tablolar.....	25
8.2. Kimlik Doğrulama ve Yetkilendirme Yapısı.....	26
8.3. Veri İlişkileri.....	26
8.4. Normalizasyon İlkeleri.....	27
8.5. Use Case Diyagramları.....	28
8.6. Akış Şemaları.....	32
9.Gereksinim Analizi.....	33
9.1. Fonksiyonel Gereksinimler.....	33
9.2. Fonksiyonel Olmayan Gereksinimler .....	34
10.Web Arayüzü.....	35
11.Test.....	41
11.1. Back-end Kısımının Testi.....	41
11.2. Front-end Kısımının Testi .....	43

## 1.ŞEKİLLER LİSTESİ

Görsel-1: Sprint Planlaması.....	16
Görsel-2: User Senaryosuna ait görsel.....	16
Görsel-3: Seller Senaryosuna ait görsel.....	16
Görsel-4: Admin Senaryosuna ait görsel.....	16
Görsel-5: ER Diyagramı.....	16
Görsel-6: Veritabanımıza ait görsel.....	16
Görsel-7 : Kullanıcıya ait use case diyagramı.....	16
Görsel-8 : UML Diyagramı.....	16
Görsel-9 : Seller’a ait use case diyagramı.....	16
Görsel-10 : Admine ait use case diagramı.....	16
Görsel-11: Kullanıcı İşlem Akış Şeması.....	16
Görsel-12: Fonksiyonel Gereksinimlere ait tablo.....	16
Görsel-13: Fonksiyonel Olmayan Gereksinimlere ait tablo.....	16
Görsel-14 : Ana Sayfa görselimiz.....	16
Görsel-15 : Kategoriler Arayüzü.....	
Görsel-16 : Giriş Yap/Kayıt Ol sayfamız.....	
Görsel-17 : Ürünlerin Listelendiği Sayfamız.....	
Görsel-18 : Ürün Detay Sayfamız.....	
Görsel-19 : Ön ödemeye geçiş ekranı .....	
Görsel-20 : Ödeme Ekranı .....	
Görsel-21 : Teklif Geçmişinin Görüntülendiği Ekran.....	
Görsel-22 : Seller Paneli.....	
Görsel-23 : Admin Paneli.....	
Görsel-24 : Uyarı Bildirimlerine ait görsel.....	
Görsel-25: Örnek Backend Testi-1.....	
Görsel-26 : Örnek Backend Testi-2.....	
Görsel-27: Örnek Backend Testi-3.....	
Görsel-28 : Örnek Backend Testi-4 .....	

Görsel-29 : Örnek Front-end Testi-1.....
Görsel-30 : Örnek Front-end Testi-2.....

## 2.GİRİŞ

### 2.1. AMAÇ

Bu projenin temel amacı, klasik sabit fiyatlı e-ticaret modellerinden farklı olarak, dinamik teklif verme sistemine dayalı bir açık artırma platformu geliştirerek hem satıcı hem de alıcı açısından değer odaklı bir alışveriş ortamı sunmaktır. Satıcılar, ürünlerini serbest piyasa koşullarında rekabetçi bir şekilde sunarak maksimum gelir elde etme fırsatı bulurken; alıcılar ise gerçek zamanlı ve şeffaf bir süreçte piyasa değerine uygun teklifler vererek daha adil fiyatlarla alışveriş yapabilmektedir. Proje, kullanıcı dostu bir arayüz, güçlü bir dijital altyapı, anlık işlem takibi ve güvenli işlem ortamı sağlayarak; bireysel kullanıcılar, küçük işletmeler ve koleksiyon meraklıları gibi farklı kullanıcı profillerine hitap eden esnek bir açık artırma sistemi oluşturmayı hedeflemektedir.

### 2.2. KAPSAM

Platform, bireysel kullanıcıların ve işletmelerin kolaylıkla ürün ekleyebileceği kullanıcı dostu bir arayüz sunar. Kullanıcılar, diledikleri ürünleri sistem üzerinden açık artırmaya çıkarabilir ve bu ürünler, belirlenen süre boyunca platformda yayımlanır. Açık artırma süreci gerçek zamanlı olarak tüm kullanıcılar tarafından izlenebilir; bu da şeffaf ve rekabetçi bir alışveriş deneyimi oluşturur.

Teklif süresi sona erdiğinde, sistem otomatik olarak en yüksek teklif veren kullanıcıyı kazanan olarak belirlemekte, böylece manuel müdahale olmaksızın güvenilir bir sonuca ulaşılmaktadır. Bu yapı, işlem sürekliliğini artırırken kullanıcı deneyimini de optimize etmektedir.

Sistem tamamen çevrim içi bir dijital altyapı üzerinde çalışmakta olup; ürün listeleme, teklif verme ve işlem kayıtlarının görüntülenmesi gibi tüm işlemler anlık olarak yansıtılmaktadır. Bu sayede platform, tüm kullanıcılar için hızlı, şeffaf ve eşitlikçi bir işlem akışı sunar.

Güvenlik, sistem mimarisinin merkezinde yer almaktadır. Kullanıcı bilgileri, işlem geçmişleri ve teklif verileri modern güvenlik protokolleri ile

korunmakta; dolandırıcılık ve yetkisiz erişim risklerine karşı güçlü bir altyapı tesis edilmiştir.

## 2.3. HEDEF KİTLE

Geliştirilen açık artırma tabanlı e-ticaret platformu, geniş bir kullanıcı yelpazesine hizmet vermeyi amaçlamakta olup, farklı ihtiyaç ve kullanım senaryolarına göre şekillendirilmiş modüller içermektedir. Hedef kitlenin temel bileşenleri aşağıda açıklanmıştır:

- **Bireysel Kullanıcılar:**  
Evlerinde kullanılmayan eşyaları, ikinci el ürünleri veya el emeği hobi ürünlerini değerlendirmek isteyen bireyler için platform; kolay erişim, hızlı ürün ekleme ve minimum teknik bilgi gerektiren kullanıcı dostu arayüz sunar. Bu grup, hem ürünlerini değerlendirmek hem de bütçesine uygun ürünleri açık artırmayla temin etmek isteyen kişilerden oluşur.
- **Küçük ve Orta Ölçekli İşletmeler (KOBİ'ler):**  
Stok fazlası, sezon sonu veya sınırlı sayıdaki ürünlerini değerlendirmek isteyen KOBİ'ler, platform aracılığıyla ürünlerini doğrudan son kullanıcıya ulaştırabilir. Böylece, geleneksel satış kanallarındaki aracı maliyetlerinden tasarruf sağlarken, ürünler dinamik fiyatlandırma sayesinde rekabetçi bir ortamda değerine ulaşır.
- **Koleksiyon Meraklıları ve Nadir Ürün Satıcıları:**  
Sanat eserleri, antikalar, sınırlı sayıda üretilmiş ürünler ya da koleksiyonluk parçalar gibi nadir veya özel nitelikteki ürünleri satmak ya da edinmek isteyen kullanıcılar için platform; şeffaf, güvenli ve zaman bazlı rekabet ortamı sunarak koleksiyon değeri taşıyan nesnelerin gerçek değerini bulmasına olanak tanır.

## 2.4. PROBLEM TANIMI

Mevcut e-ticaret sistemleri, sabit fiyatlı satış modeline dayandığı için hem satıcıların ürünlerini gerçek değerinde satmalarını zorlaştırmakta hem de alıcıların fiyat üzerinde etkide bulunmasına imkân vermemektedir. Ayrıca, stok fazlası ürünlerin değerlendirilmesi, ikinci el ve özel ürünlerin satışında güvenli ve dinamik çözümler sunamamaktadır. Gerçek zamanlı teklif verme, şeffaflık ve güvenlik gibi temel ihtiyaçlara yanıt veremeyen bu sistemler, hem bireysel kullanıcılar hem de işletmeler için yetersiz kalmaktadır.

Geliştirilen açık artırma platformu, bu eksikleri gidermek amacıyla; dinamik fiyatlandırma, gerçek zamanlı takip, güvenli işlem altyapısı ve kullanıcı etkileşimini esas alan bir yapı sunmaktadır.

## 2.5. PROJE EKİBİ

Proje, aşağıdaki uzmanlık alanlarına sahip ekip üyeleri tarafından geliştirilmiştir:

- **Mehmet Ali NACAK** – Grup Lideri / Back-end Developer
- **Emre MERAL** – Back-end Developer
- **Yusuf YÜCEL** – Back-end Developer
- **Ömer ÇELİK** – Front-end Developer
- **Dilara ADIGÜZEL** – Full Stack Developer
- **Osman DEMİR** – Front-end Developer
- **Muhammed Cenab KÖROĞLU** – Test Mühendisi
- **Serhat Eren ATALAY** – Test Mühendisi



## 3.TANIMLAR

- Tanımlar Listesi:

Terim	Tanım
API	Application Programming Interface - Uygulama Programlama Arayüzü
MSSQL	Microsoft SQL Server
JWT	JSON Web Token
CRUD	Create, Read, Update, Delete
HOC	Higher-Order Component
JSON	JavaScript Object Notation
ER	Entity Relation
HTML	Hyper Text Markup Language
UML	Unified Modeling Language
REST	Representational State Transfer
UI	User Interface
UX	User Experience

## 4.KULLANILAN PLATFORMLAR

### 4.1. Proje Yönetim Süreci ve Görev Takibi - Trello

Trello, proje sürecinin planlanmasında ve ekip içi koordinasyonun sağlanmasında aktif bir biçimde kullanılmıştır. Görevler “Yapılacak”, “Yapılıyor” ve “Tamamlandı” başlıkları altında kategorize edilerek süreçlerin anlık olarak izlenmesi mümkün kılınmıştır. Her görev kartında sorumlu kişiler tanımlanmış, açıklamalar ve belgeler eklenerek takip kolaylaştırılmıştır. Bu yapı, ekip içi iletişimi güçlendirmiş, zaman yönetimini optimize etmiş ve teslim sürelerine uyumu artırmıştır.

### 4.2. Visual Studio – Back-end Geliştirme

Projede sunucu taraflı işlemlerin geliştirilmesi için Visual Studio IDE’si tercih edilmiştir. C# dili ve ASP.NET mimarisi kullanılarak; kullanıcı yönetimi, teklif işlemleri, zamanlayıcılar ve veri erişim katmanları oluşturulmuştur. Visual Studio’nun hata ayıklama (debugging) araçları ve .NET uyumu, geliştirme sürecinde verimliliği artırmış, kodun modüler ve sürdürülebilir şekilde yazılmasına olanak sağlamıştır.

#### **Avantajları:**

- Güçlü hata ayıklama ve otomatik tamamlama özellikleri
- .NET Framework ile tam uyum
- Visual tasarım araçları sayesinde hızlı form geliştirme

#### **Dezavantajları:**

- Kaynak tüketimi yüksektir (özellikle düşük donanımlı bilgisayarlarda yavaş çalışabilir)

### 4.3. Visual Studio Code – Front-end Geliştirme

Kullanıcı arayüzü geliştirmeleri için hafif ve genişletilebilir bir editör olan Visual Studio Code kullanılmıştır. HTML, CSS, JavaScript ve ilgili framework’ler ile sayfa tasarımları, kullanıcı etkileşimleri ve dinamik

içerikler oluşturulmuştur. Eklenti desteği, sürüm kontrol sistemleriyle entegrasyon kolaylığı ve özelleştirilebilir yapısı nedeniyle tercih edilmiştir.

**Avantajları:**

- Hafif ve hızlı çalışma ortamı
- Geniş eklenti desteği
- Çapraz platform uyumluluğu (Windows, Linux, Mac)

**Dezavantajları:**

- Yerleşik hata ayıklama yetenekleri sınırlıdır
- Büyük projelerde yapılandırma karmaşık hale gelebilir

#### 4.4. Veritabanı Yönetimi - Microsoft SQL Server (MSSQL)

Veri saklama, işleme ve ilişkilendirme işlemleri için Microsoft SQL Server kullanılmıştır. Ürün bilgileri, kullanıcı verileri, açık artırma kayıtları ve teklif geçmişi gibi tüm veri kümeleri ilişkisel veritabanı yapısı ile modellenmiştir:

**Avantajları:**

- Güçlü veri bütünlüğü ve güvenlik desteği
- Büyük veri kümeleriyle verimli çalışma yeteneği
- SQL Server Management Studio (SSMS) ile kolay yönetim ve sorgu desteği

**Dezavantajları:**

- Lisans maliyeti yüksektir (özellikle kurumsal sürümler için)
- Sadece Windows ortamında tam performans sağlar

## 4.5. Sürüm Kontrolü ve İş Birliği – GitKraken

Proje geliştirme sürecinde kaynak kodlarının yönetimi, sürüm takibi ve ekip içi kod iş birliği için GitKraken aracı tercih edilmiştir. GitKraken'in sunduğu kullanıcı dostu arayüz sayesinde dallanma (branch), birleştirme (merge), çatallanma (fork) ve commit işlemleri grafiksel olarak kolayca yönetilmiştir. Bu sayede ekip üyeleri aynı anda farklı modüllerde çalışabilmiş, kod çatışmaları minimuma indirilmiştir.

GitKraken, Git ile tam uyumlu yapısı sayesinde tüm kod geçmişinin şeffaf ve güvenli biçimde takip edilmesine olanak tanımış; ekip içi senkronizasyonun ve proje bütünlüğünün korunmasında önemli rol oynamıştır.

### **Avantajları:**

- Grafiksel arayüz ile kolay branch ve commit takibi
- Git işlemlerinde hata yapma riskini azaltan kullanıcı dostu yapı
- Ekip içi iş birliğini kolaylaştıran görsel proje geçmişi

### **Dezavantajları:**

- Ücretsiz sürümde bazı gelişmiş özellikler kısıtlıdır
- Yüksek RAM kullanımı, düşük donanımlı cihazlarda performans sorunlarına yol açabilir

## 5.Yazılım Mimarisi ve Katmanlar

Bu projede, sürdürülebilirlik, yeniden kullanılabilirlik ve güvenlik ilkeleri doğrultusunda **katmanlı mimari** yaklaşımı benimsenmiştir. Ayrıca proje süresince geliştirici deneyimini artırmak ve iş akışını optimize etmek amacıyla çeşitli araç ve teknolojiler entegre edilmiştir.

### 5.1 Katmanlı Mimari

Proje, beş temel katmandan oluşan katmanlı mimari yapısına sahiptir:

- **Client (Kullanıcı Arayüzü):**  
Kullanıcıların sistemle doğrudan etkileşim kurduğu bu katman, React kütüphanesi kullanılarak geliştirilmiştir. Tek sayfa uygulama (SPA) mimarisi ve bileşen tabanlı yapısıyla hızlı ve dinamik bir kullanıcı deneyimi sunmaktadır. Kullanıcıdan gelen işlemler (örneğin ürün listeleme, teklif verme) HTTP istekleri aracılığıyla API katmanına iletilmektedir.
- **API (Uygulama Katmanı):**  
Kullanıcı arayüzünden gelen HTTP isteklerini karşılar. Bu katmanda yer alan Controller yapıları, Business katmanı ile iletişim kurarak gerekli veri akışını sağlar. API, dış dünya ile sistem arasındaki güvenli geçiş katmanıdır.
- **Business Katmanı (İş Mantığı):**  
Uygulamanın iş kuralları ve mantıksal süreçleri bu katmanda yürütülmektedir. Örneğin, bir ürüne verilen teklifin geçerli olup olmadığı veya açık artırmanın süresinin dolup dolmadığı gibi kontroller bu bölümde gerçekleştirilir.  
MyGalaxy\_Auction\_Business isimli projede servis sınıfları, arayüzler ve somut uygulamalar mantıksal bütünlük içinde yapılandırılmıştır.
- **Core Katmanı (Temel Yapılar):**  
Tüm katmanlarda ortak olarak kullanılabilecek veri modelleri, yardımcı sınıflar ve sabit yapılar bu katmanda yer almaktadır. Bu yapı, kod tekrarını önleyerek bağımlılıkları azaltmakta ve sistemin sürdürülebilirliğini desteklemektedir.

MyGalaxy\_Auction\_Core projesinde Models, Common, MailHelper gibi klasör yapıları altında organize edilmiştir.

- **Data Access Katmanı (Veri Katmanı):**

Veritabanı işlemleri bu katman üzerinden yürütülmektedir. Microsoft SQL Server (MSSQL) veritabanı ile birlikte Entity Framework Core kullanılmış; tüm CRUD işlemleri ApplicationDbContext sınıfı üzerinden gerçekleştirilmiştir.

MyGalaxy\_Auction\_Data\_Access projesi, Context, Models, Domain ve Migrations gibi klasörler aracılığıyla yapılandırılmıştır.

## 5.2 API Dokümantasyonu – Swagger Kullanımı

Tüm API uç noktaları, Swagger arayüzü aracılığıyla açık biçimde dokümante edilmiştir. Swagger, sistemin dış geliştiriciler ve test mühendisleri tarafından erişilebilirliğini ve test edilebilirliğini artırmaktadır. Bu sayede API entegrasyonu kolaylaşmış, hata tespiti ve giderilmesi hızlanmıştır.

## 5.3 Güvenlik Önlemleri

Projenin güvenlik politikaları çerçevesinde çeşitli modern yöntemler uygulanmıştır:

- **JWT (JSON Web Token):**

Kimlik doğrulama ve kullanıcı yetkilendirme işlemleri için JWT tabanlı güvenlik yapısı kullanılmaktadır. Her kullanıcının işlemleri, token doğrulaması üzerinden kontrol edilerek güvenli bir işlem akışı sağlanmıştır.

- **HOC (Higher-Order Component):**

React tarafında, oturum bazlı erişim kontrolü için HOC yapısı kullanılmıştır. Belirli sayfalara yalnızca yetkili kullanıcıların erişmesine izin verilerek arayüz düzeyinde güvenlik artırılmıştır.

## 6.Proje Zaman Çizelgesi ve Sprint

### 6.1 Proje Genel Zaman Aralığı

- **Başlangıç Tarihi:** 15 Şubat 2025
- **Bitiş Tarihi:** 1 Mayıs 2025
- **Toplam Süre:** Yaklaşık 10 hafta + 5 gün
- **Toplam Sprint Sayısı:** 5 Sprint + 1 son değerlendirme

Proje süreci, çevik yazılım geliştirme metodolojisi (Agile) esas alınarak planlanmış olup, her sprint belirli hedef ve görev kümeleri çerçevesinde yönetilmiştir. Sprintler iki haftalık zaman aralıklarına bölünmüş ve her biri sonunda yapılacak işlerin teslimi ile ilerleme sağlanmıştır.

### 6.2 Sprint Planlaması ve İçerik

Sprint toplantıları, yazılım geliştirme sürecinde ekip içi koordinasyonu sağlamak, ilerlemeyi değerlendirmek ve sonraki adımları planlamak amacıyla düzenli olarak gerçekleştirilmiştir. Projemiz, 15 Şubat 2025 tarihinde başlamış ve 1 Mayıs 2025 tarihinde başarıyla tamamlanmıştır. Bu süreç boyunca toplam **beş sprint** planlanmış ve her sprint, belirli modüllerin geliştirilmesini hedefleyen bir yol haritası çerçevesinde yürütülmüştür.

#### **Sprint 1: 15 Şubat – 28 Şubat 2025**

**Hedef:** Temel gereksinimlerin toplanması ve proje altyapısının oluşturulması

#### **Gerçekleştirilen Görevler:**

- Kullanıcı rolleri ve sistem ihtiyaçlarının analiz edilmesi
- Kullanıcı Hikayeleri (User Stories) yazımı
- Veritabanı tasarımı (ER Diyagramı, tabloların tanımlanması)
- Kullanılacak teknolojilerin belirlenmesi (React, SQL Server)

- Proje sürüm kontrol sistemi kurulumu (GitHub, GitKraken)
- Temel React arayüzü ve backend çatısının kurulumu

### **Sprint 2: 1 Mart – 14 Mart 2025**

**Hedef:** Kullanıcı işlemleri ve temel sistem işlevlerinin geliştirilmesi

#### **Gerçekleştirilen Görevler:**

- Kullanıcı kayıt ve giriş sistemi (JWT ile kimlik doğrulama)
- Roller bazlı sistem ayrımı: Kullanıcı / Satıcı / Yönetici (Admin)
- Ürün listeleme ve detay sayfalarının geliştirilmesi

### **Sprint 3: 15 Mart – 28 Mart 2025**

**Hedef:** Açık artırma ve teklif süreçlerinin uygulanması

#### **Gerçekleştirilen Görevler:**

- Satıcıların ürün ekleyebilmesi ve güncelleyebilmesi
- Açık artırma başlatma ve teklif sürecinin yönetimi
- Teklif verme ve teklif geçerlilik kontrollerinin yazımı
- Teklif geçmişi görüntüleme modülünün eklenmesi
- Ön ödeme kısmının yapılandırılması
- Akış diyagramları ve sistem flowchart'larının hazırlanması

### **Sprint 4: 29 Mart – 11 Nisan 2025**

**Hedef:** Yönetici yetkilerinin tanımlanması ve sistemsel denetimlerin sağlanması

#### **Gerçekleştirilen Görevler:**

- Admin panelinin oluşturulması: kullanıcı ve ürün yönetimi
- Raporlama ve loglama altyapısının geliştirilmesi



- Yetki kontrollerinin ve erişim doğrulamalarının yazılması
- Geliştirilmiş hata yönetimi ve bildirim mekanizmalarının entegrasyonu

### **Sprint 5: 12 Nisan – 25 Nisan 2025**

**Hedef:** Test süreçleri, hata düzeltmeleri ve kullanıcı deneyimi iyileştirmeleri

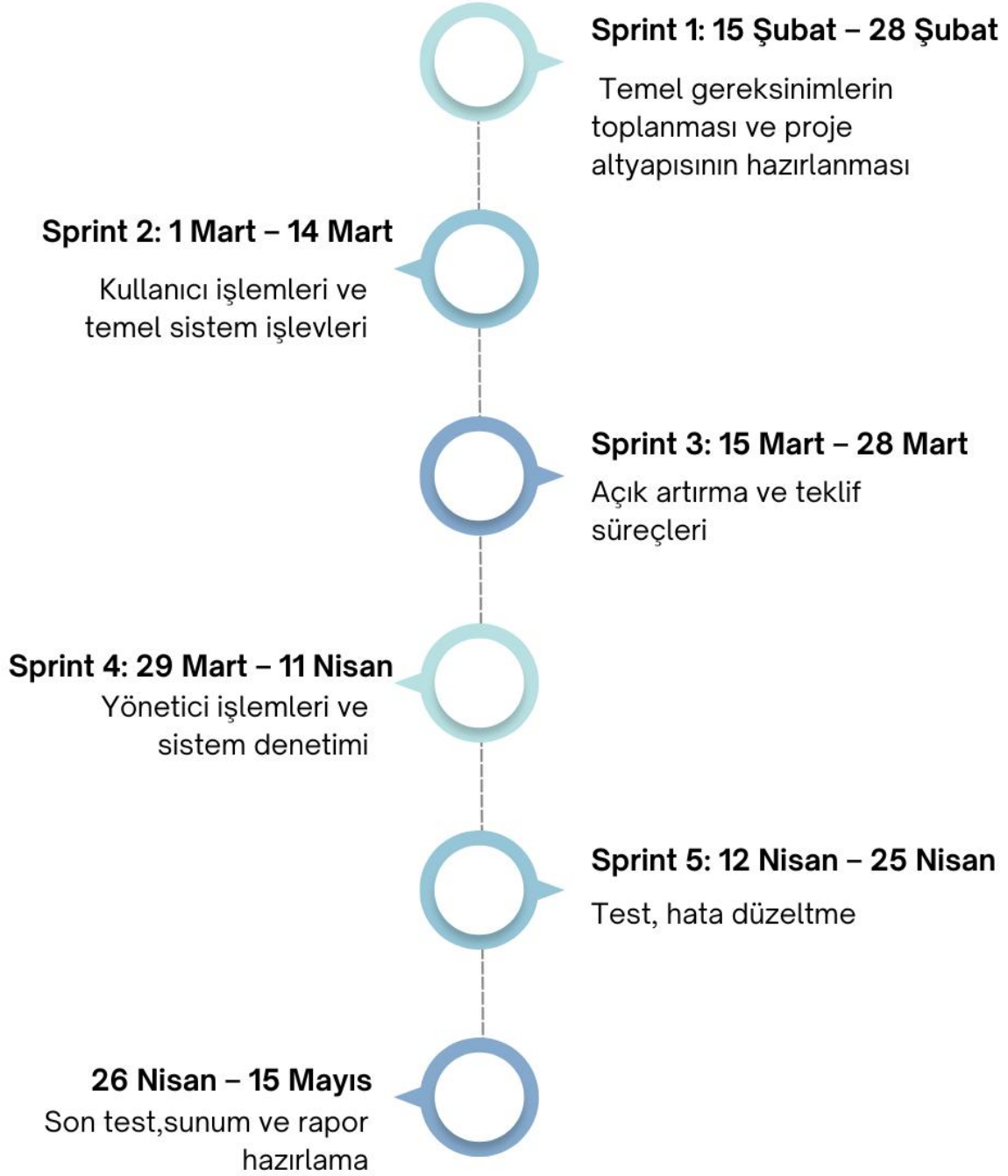
#### **Gerçekleştirilen Görevler:**

- Fonksiyonel testlerin gerçekleştirilmesi
- Backend birim testlerinin yazılması
- Arayüzde kullanıcı deneyimi ve mobil uyumluluk kontrolleri
- Erişilebilirlik (accessibility) denetimlerinin yapılması

### **Son Değerlendirme : 26 Nisan – 15 Mayıs 2025**

#### **Gerçekleştirilen Görevler:**

- Son testlerin yapılması
- Rapor ve sunum hazırlıkları



Görsel-1: Sprint planlaması

### 6.3 Günlük Scrum Toplantıları (Daily Stand-ups)

Proje süreci boyunca her sprint döneminde düzenli olarak günlük Scrum toplantıları gerçekleştirilmiştir. Bu toplantılar, genellikle akşam saatlerinde çevrim içi olarak gerçekleştirilmiş (Microsoft Teams) ve her bir takım üyesinin bir önceki gün ne yaptığını, o gün ne yapmayı planladığını ve karşılaştığı olası engelleri kısa ve öz şekilde paylaşmasına imkân tanımıştır. Toplantılar, maksimum 15 dakika olacak şekilde yapılandırılmış ve sürecin verimli ilerlemesi için zaman yönetimine dikkat edilmiştir.

Daily Stand-up toplantılarının temel amacı, ekip üyeleri arasındaki iletişimi sürekli kılmak, şeffaflığı sağlamak ve projenin ilerleyişini günlük düzeyde takip edebilmek olmuştur. Bu toplantılar sayesinde gelişim sürecinde oluşabilecek olası riskler önceden fark edilmiş, zamanında müdahale edilerek sprint hedeflerinin başarıyla tamamlanması sağlanmıştır. Ayrıca takımın moral ve motivasyonu açısından da bu toplantılar önemli bir rol oynamış; ekip içi koordinasyonu ve sorumluluk bilincini güçlendirmiştir.

Toplantılarda izlenen format gereği her birey şu üç soruya yanıt vermiştir:

- Dün ne yaptım?
- Bugün ne yapacağım?
- Karşılaştığım bir engel var mı?

Bu sistematik yaklaşım sayesinde, proje yönetiminde çeviklik korunmuş, sprintlerin şeffaflığı ve takibi kolaylaştırılmıştır. Tüm toplantılar, proje yöneticisi veya takım lideri tarafından kolaylaştırılmış, gerekli durumlarda notlar alınarak Trello üzerinden kayıt altına alınmıştır.

### 6.3 Backloglar

Açık artırma web sitesi projemizin işleyişini sağlıklı ve kullanıcı odaklı yürütmek amacıyla belirlenen backlog maddeleri, temel fonksiyonlar ve kullanıcı deneyimini artırıcı özellikler çerçevesinde şekillendirilmiştir.

- **Backlog 1: Kullanıcı Kayıt ve Giriş Sistemi**

Kullanıcıların siteye kayıt olabilmesi, kimlik doğrulama süreçlerinin güvenli şekilde yönetilmesi ve oturum açma işlemlerinin etkin hale getirilmesi.

Ayrıca, farklı kullanıcı rolleri (alıcı, satıcı, admin) için ayrı erişim yetkilerinin tanımlanması.

- **Backlog 2: Ürün Ekleme ve Yönetimi**

Satıcıların açık artırmaya sunmak üzere ürünlerini kolayca ekleyebilmeleri, ürün detaylarını güncelleyebilmeleri ve yönetebilmeleri için kullanıcı dostu bir arayüz geliştirilmesi.

- **Backlog 3: Açık Artırma Süreci ve Teklif Verme**

Kullanıcıların aktif açık artırmalara teklif verebilmesi, tekliflerin geçerlilik kontrollerinin sağlanması, canlı teklif takip sistemi ve teklif geçmişinin görüntülenebilmesi.

- **Backlog 4: Ödeme ve Sipariş Yönetimi**

Açık artırma öncesi ön ödeme işlemleri ve açık artırmayı kazanan kullanıcıların güvenli ödeme yapabilmesi, siparişlerin takibi ve yönetimi için kapsamlı modüllerin oluşturulması.

- **Backlog 5: Ürün Arama ve Filtreleme Özellikleri**

Kullanıcıların ürünleri kategori, fiyat aralığı, açık artırma durumu gibi farklı kriterlere göre filtreleyerek arama yapabilmelerinin sağlanması.

## 6.4 Definition of Done (Tamamlanma Kriterleri)

Açık artırma temelli e-ticaret projemizde her bir ürün, modül veya kullanıcı işlevselliği aşağıda belirtilen kriterler sağlandığında "tamamlanmış" kabul edilecektir. Bu standartlar, sistemin güvenli, kararlı ve kullanıcı dostu şekilde çalışmasını garanti altına almak için belirlenmiştir.

### Genel Teknik ve Fonksiyonel Kriterler:

#### Kod Geliştirme ve Standartlara Uygunluk:

- Geliştirilen modül, önceden belirlenmiş yazılım geliştirme standartlarına uygun şekilde yazılmıştır. Kod temizliği, okunabilirlik ve sürdürülebilirlik sağlanmıştır.

### **Frontend ve Backend Entegrasyonu:**

- Geliştirilen işlev, kullanıcı arayüzü ile sunucu tarafı arasında eksiksiz entegrasyon sağlayarak uçtan uca test edilmiştir.

### **Veritabanı Kontrolleri:**

- İlgili veri yapıları doğru şekilde oluşturulmuş, veri güvenliği ve bütünlüğü sağlanmıştır.

### **Özelleşmiş Açık Artırma Fonksiyonları:**

#### **Ürün Yayınlama ve Yönetimi:**

- Satıcıların ürün ekleyebileceği, açıklama, fotoğraf, başlangıç fiyatı gibi detayları girebileceği ekranlar eksiksiz çalışmaktadır.

#### **Açık Artırma Süreci:**

- Ürün sayfasında açık artırma süreci gerçek zamanlı şekilde işler, teklifler anlık olarak diğer kullanıcılara yansıtılır.

#### **Teklif Verme ve Otomatik Güncelleme:**

- Kullanıcılar güvenli bir şekilde teklif verebilir, sistem en yüksek teklifi anında günceller ve görüntüler.

#### **Açık Artırma Bitiş ve Kazanan Belirleme:**

- Süre sonunda açık artırma otomatik olarak sonlandırılır, kazanan teklif belirlenir ve sistemde kaydedilir.

### **Kalite Güvence ve Yayın Kriterleri:**

#### **Testlerin Başarılı Olması:**

- Fonksiyonel testler, kullanıcı testleri ve olası hata senaryoları test edilmiştir.

#### **Hata Takibi ve Giderimi:**

- Tespit edilen tüm hatalar giderilmiş, sistemin hata üretmeden çalıştığı doğrulanmıştır.

### **Mobil ve Tarayıcı Uyumluluğu:**

- Site, yaygın tarayıcılar (Chrome, Firefox, Safari, Edge) ve mobil cihazlarla tam uyumludur.

**Güvenlik Önlemleri:**

- Giriş yetkilendirme, veri şifreleme, teklif güvenliği ve açık artırma manipölasyonlarına karşı koruma sağlanmıştır.

## 7.Kullanıcı Senaryoları

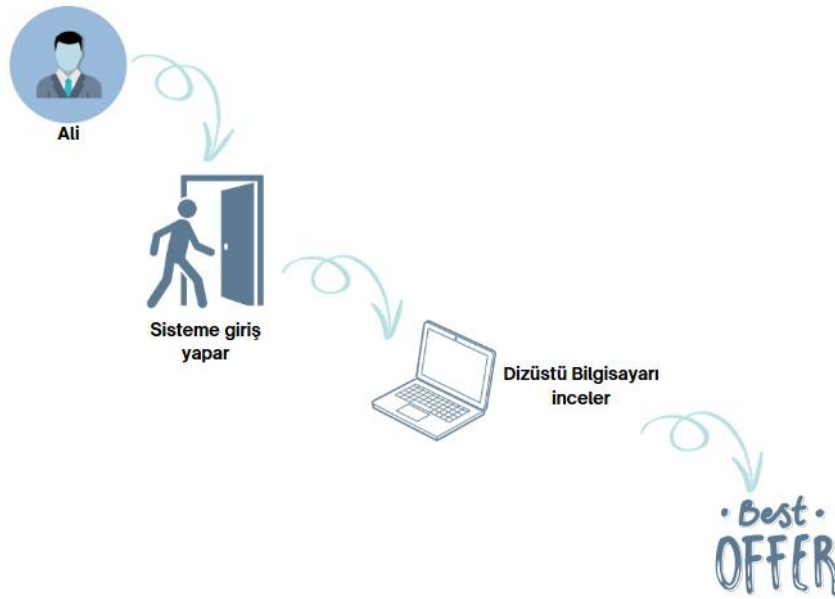
### 7.1 Senaryo 1 :

#### **Kullanıcı Tipi: User**

Ali isimli kullanıcı, giriş yaptıktan sonra "Elektronik" kategorisinde yer alan bir dizüstü bilgisayarı inceler. Ürün açıklamasını ve mevcut en yüksek teklifi görüntüledikten sonra, bu tekliften yüksek olacak şekilde teklif verir. Sistem, teklifin geçerli olduğunu doğrular, yeni teklifin en yüksek olduğunu tüm kullanıcılarla gerçek zamanlı paylaşır ve Ali'ye başarı bildirimini gönderir.

#### **Beklenen Sonuç:**

Teklif başarılı şekilde kaydedilir, ekran anlık olarak güncellenir ve kullanıcı teklifi geri çekemez.



Görsel-2: User Senaryosuna ait görsel

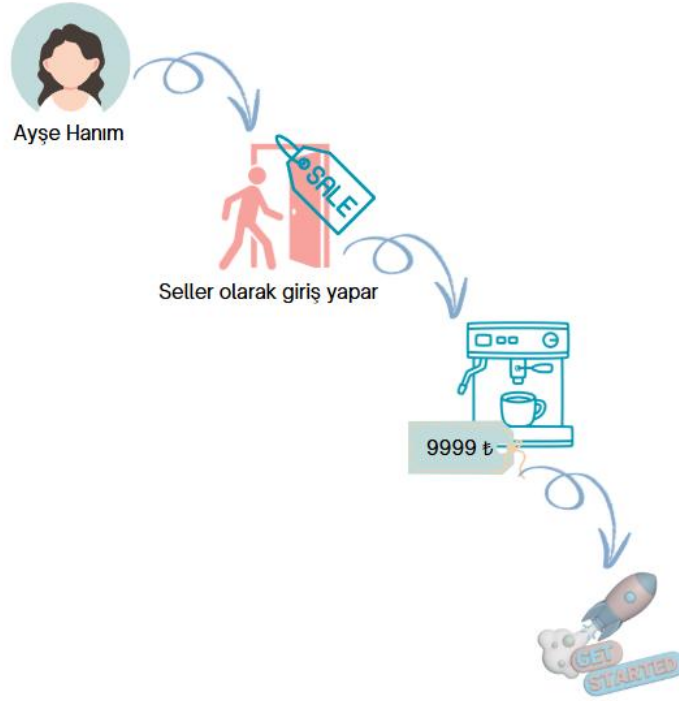
## 7.2 Senaryo 2 :

### Kullanıcı Tipi: Seller

Ayşe Hanım sisteme satıcı hesabıyla giriş yapar. "Yeni Ürün Ekle" sekmesinden bir kahve makinesi ekler. Ürünün başlığını, açıklamasını, başlangıç fiyatını ve açık artırma süresini girer. Ürün görsellerini yükledikten sonra açık artırmayı başlatır.

### Beklenen Sonuç:

Ürün yayına alınır, açık artırma süreci başlar ve diğer kullanıcılar ürünü görüntüleyebilir.



Görsel-3: Seller Senaryosuna ait görsel



### 7.3 Senaryo 3 :

#### **Kullanıcı Tipi: Admin**

Yönetici, sistem panelinde yeni eklenen ürünleri denetlerken, bir satıcının uygunsuz içerik (örneğin sahte marka, yasaklı ürün vb.) yüklediğini fark eder. İlgili ürünü inceleme altına alır, yayından kaldırır ve kullanıcıya sistem içi uyarı gönderir.

#### **Beklenen Sonuç:**

Uygunsuz ürün yayından kaldırılır.



Görsel-4: Admin Senaryosuna ait görsel

## 8.Veritabanı Tasarımı ve Kimlik

Projemiz, **katmanlı mimari** esas alınarak geliştirilmiş olup, bu yapı sayesinde sistemsel işlevlerin ayrıştırılması ve yönetimi kolaylaştırılmıştır. Back-End tarafında **.NET teknolojisi** kullanılmış; veri işlemleri için **Entity Framework Core** tercih edilmiştir. Kod tabanlı **migration yapısı** ile oluşturulan veritabanı şeması, yazılım sürümleri arasında geçişi kolaylaştırmış ve veri tabanı güncellemelerini izlenebilir hale getirmiştir. Bu süreçlerin kayıt altına alındığı `_EFMigrationsHistory` tablosu aracılığıyla yapılan tüm migration işlemleri takip edilebilmektedir.

### 8.1. Ana Varlıklar ve Tablolar

Projemizin işleyişinde kullanılan temel varlıklar ve ilişkili tablolar aşağıda belirtilmiştir:

- **AspNetUsers:**  
Sisteme kayıtlı tüm kullanıcıların temel kimlik ve oturum bilgilerini içermektedir.
- **Dresses, Vehicles, Electronics, MusicalInstruments:**  
Satışa sunulan ürün türleri, kendi özel tabloları altında ayrı ayrı yönetilmiştir. Bu sayede ürün türlerine göre detaylı ve ölçeklenebilir bir yapı sağlanmıştır.
- **Bids:**  
Kullanıcıların ürünlere verdiği teklif bilgilerini barındırır. Her teklif hem kullanıcıya hem de ürüne bağlı olarak oluşturulur.
- **PaymentHistories:**  
Satış işlemi tamamlanan ürünlerin ödeme geçmişi bu tabloda takip edilmektedir. Satıcı ve alıcı tarafındaki mali işlemler burada kayıt altına alınır.
- **Estates:**  
Ürünlerin genel kategorilerini tanımlar. Tablolar arası sınıflandırma ve filtreleme işlemlerine katkı sağlar.

## 8.2. Kimlik Doğrulama ve Yetkilendirme Yapısı

Projemizde kullanıcı kimlik doğrulaması ve yetkilendirme süreçleri, **ASP.NET Identity** çatısı altında güvenli, modüler ve özelleştirilebilir bir biçimde yapılandırılmıştır. Aşağıdaki tablolar bu yapının temel taşlarını oluşturur:

- AspNetUsers: Kullanıcı bilgileri
- AspNetRoles: Roller
- AspNetUserRoles: Kullanıcı–Rol ilişkileri
- AspNetUserClaims: Kimlik talepleri (claims)
- AspNetUserLogins: Harici oturum bağlantıları

Bu yapı sayesinde kullanıcıların sisteme güvenli şekilde giriş yapması, rollerine uygun erişim düzeylerinin belirlenmesi ve yönetilmesi mümkün hale getirilmiştir.

## 8.3. Veri İlişkileri

Veritabanı modelimizdeki temel ilişkiler aşağıdaki gibi kurgulanmıştır:

- Her bir ürün (örneğin bir müzik aleti) açık artırma yoluyla satışa sunulmaktadır.
- Birden fazla kullanıcı bu ürüne teklif verebilir (çoktan-bire ilişki).
- Her teklif, hem belirli bir kullanıcıya hem de belirli bir ürüne bağlıdır.

Bu ilişkilendirme sayesinde:

- **Bir kullanıcı → birçok teklif** verebilir.
- **Bir ürün → birçok teklif** alabilir.
- **Her teklif → bir kullanıcı ve bir ürün** ile eşleşmelidir.

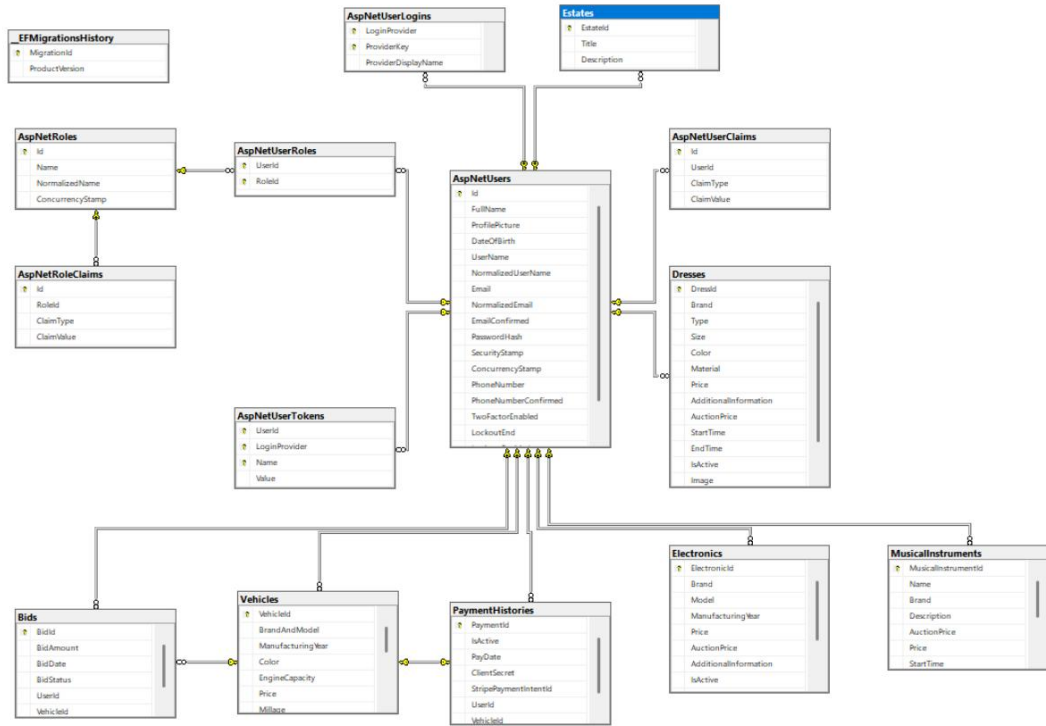
Bu yapı, veri bütünlüğünü sağlarken teklif sürecinin sistematik olarak izlenmesini de mümkün kılar.

## 8.4. Normalizasyon İlkeleri

Veritabanı tasarımı sırasında **3. Normal Form (3NF)** prensiplerine dikkat edilmiştir. Bu kapsamda:

- Veri tekrarının önüne geçilmiştir.
- Veri bütünlüğü ve tutarlılığı korunmuştur.
- Sorgu performansı optimize edilerek gereksiz veri yükü azaltılmıştır.

Bu yapılandırma, sistemin ölçeklenebilirliğini ve bakım kolaylığını desteklemektedir.



Görsel-5: ER Diyagramı

ElectronicId	Brand	Model	Manufactur...	Price	AuctionPrice	Additional...	IsActive	Image	StartTime	EndTime	SellerId
1	Apple	iPhone 13	2021	999,99	0	Az kullanılm...	True	https://cdn1...	2025-04-20 ...	2025-04-30 ...	381b1269-1...
2	Samsung	Galaxy S22	2022	899,99	0	Garantili	True	https://img...	2025-04-20 ...	2025-04-27 ...	381b1269-1...
3	Sony	PlayStation 5	2020	699,99	0	2 kol ile bir...	True	https://img...	2025-04-20 ...	2025-05-04 ...	381b1269-1...
4	Microsoft	Xbox Series X	2020	649,99	0	Kutu dahil	True	https://cdn...	2025-04-20 ...	2025-04-29 ...	381b1269-1...
5	Dell	XPS 13	2021	1200,00	0	Hafif çizik	True	https://cdn...	2025-04-20 ...	2025-04-25 ...	af7b9e9c-a...
6	HP	Spectre x360	2022	1300,00	0	Touch ekran	True	https://www...	2025-04-20 ...	2025-04-28 ...	af7b9e9c-a...
7	Apple	MacBook Ai...	2021	999,00	0	Pil döngüsü ...	True	https://med...	2025-04-20 ...	2025-05-01 ...	af7b9e9c-a...
8	Lenovo	ThinkPad X1...	2021	1100,00	0	İş için ideal	True	https://m.m...	2025-04-20 ...	2025-04-26 ...	af7b9e9c-a...
9	Canon	EOS R	2019	1500,00	0	Fotoğrafçıla...	True	https://tech...	2025-04-20 ...	2025-05-02 ...	af7b9e9c-a...
10	DJI	Mavic Air 2	2020	800,00	0	Hobi için dr...	True	https://ir.o...	2025-04-20 ...	2025-04-30 ...	af7b9e9c-a...
11	Samsung	Galaxy Tab ...	2022	600,00	0	Tablet, kale...	True	https://www...	2025-04-20 ...	2025-04-25 ...	af7b9e9c-a...
12	Apple	iPad Pro	2021	1000,00	0	11 inch ekran	True	https://i0.w...	2025-04-20 ...	2025-04-24 ...	af7b9e9c-a...
13	Bose	QuietComfo...	2022	299,99	0	Gürültü eng...	True	https://cdn...	2025-04-20 ...	2025-04-23 ...	af7b9e9c-a...
14	Sony	WH-1000X...	2021	279,99	0	Kulak üstü k...	True	https://sony...	2025-04-20 ...	2025-04-27 ...	af7b9e9c-a...
15	GoPro	Hero 10	2021	399,99	0	Aksiyon ka...	True	https://sale...	2025-04-20 ...	2025-04-26 ...	af7b9e9c-a...
16	Asus	ROG Zephyr...	2022	1400,00	0	Oyun bilgis...	True	https://cdn...	2025-04-20 ...	2025-04-29 ...	af7b9e9c-a...
17	Apple	Apple Watc...	2021	450,00	0	Akıllı saat	True	https://usbs...	2025-04-20 ...	2025-04-28 ...	af7b9e9c-a...
18	Garmin	Fenix 6	2020	500,00	0	Outdoor saat	True	https://cdn...	2025-04-20 ...	2025-04-25 ...	af7b9e9c-a...
19	Razer	Blade 15	2022	1600,00	0	Gamer lapt...	True	https://m.m...	2025-04-20 ...	2025-05-01 ...	af7b9e9c-a...
20	LG	OLED CX	2020	1800,00	0	4K OLED TV	True	https://www...	2025-04-20 ...	2025-05-03 ...	af7b9e9c-a...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

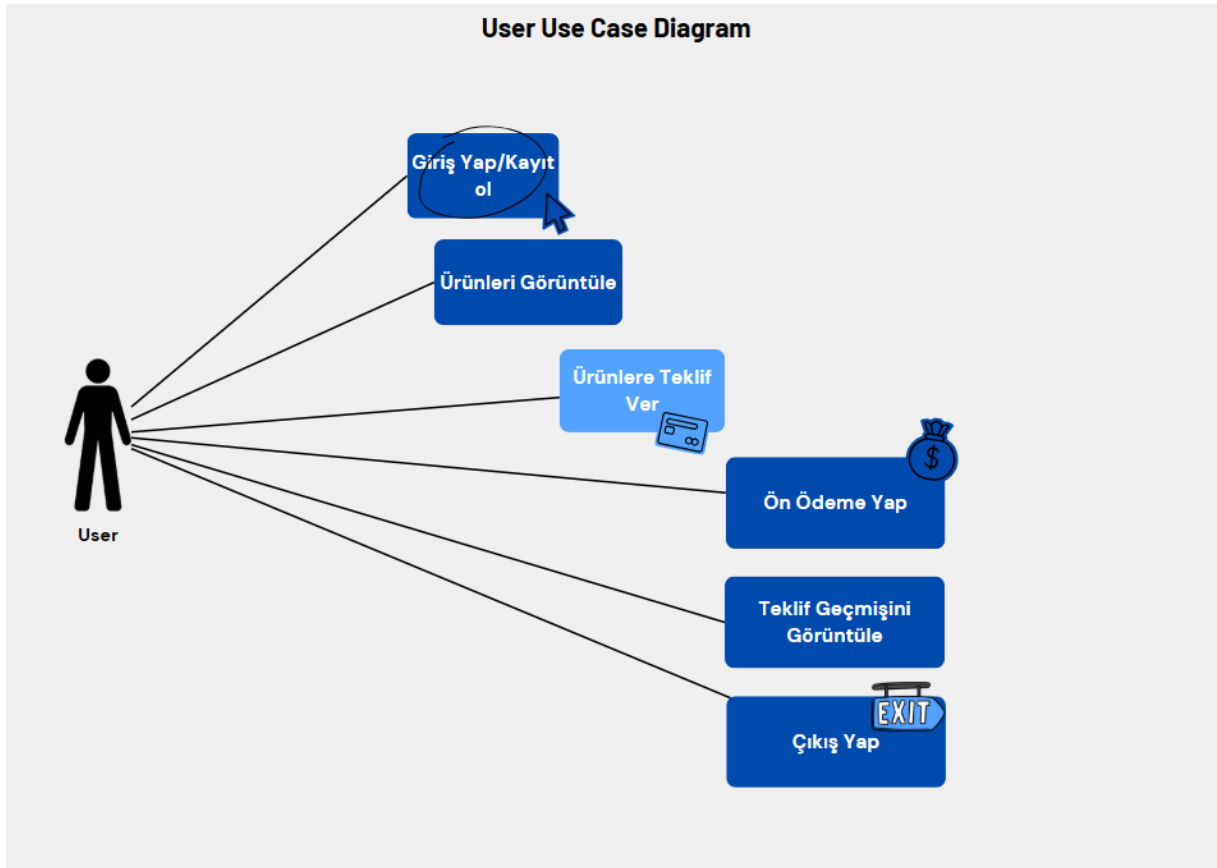
Görsel-6 : Veritabanımıza ait görsel

## 8.5 USE CASE DİYAGRAMLARI

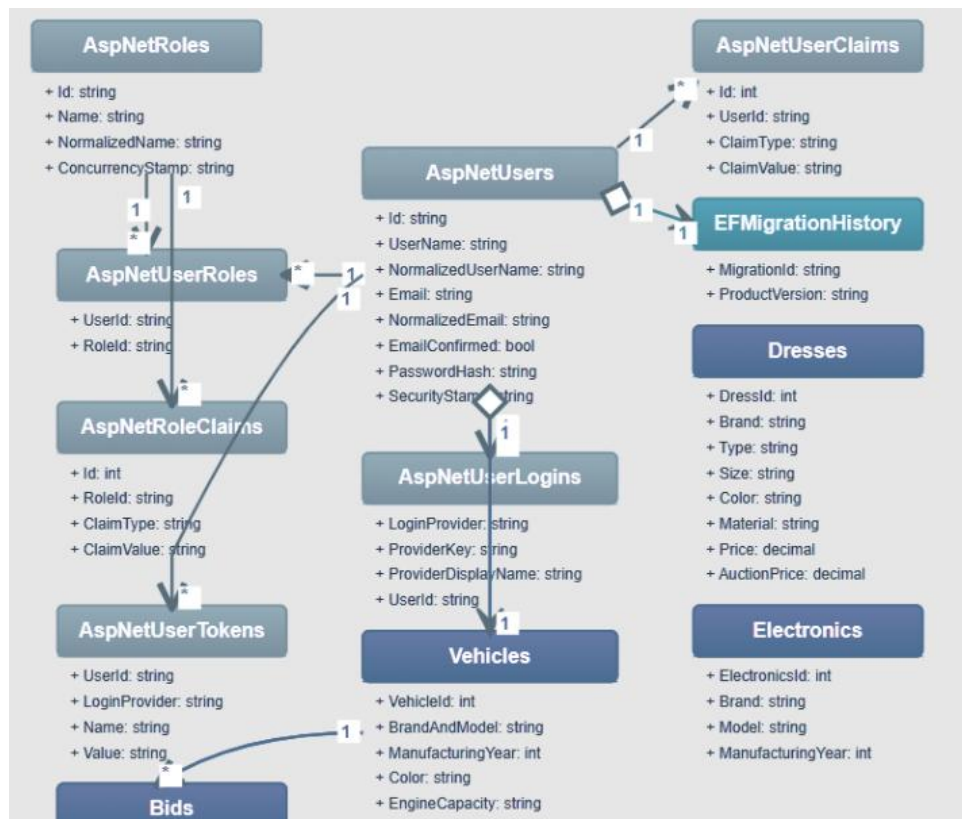
Bu bölümde sunulan use case (kullanım senaryosu) diyagramları, geliştirilen açık artırma tabanlı e-ticaret sisteminin kullanıcılarla olan etkileşimlerini modellemek amacıyla hazırlanmıştır. Kullanım senaryoları, sistemin işlevsel gereksinimlerini grafiksel olarak ifade eden UML tabanlı bir yaklaşımdır ve kullanıcıların sistemle hangi işlemleri gerçekleştirebileceğini açık biçimde ortaya koyar.

Sistem üç ana kullanıcı rolünü desteklemektedir:

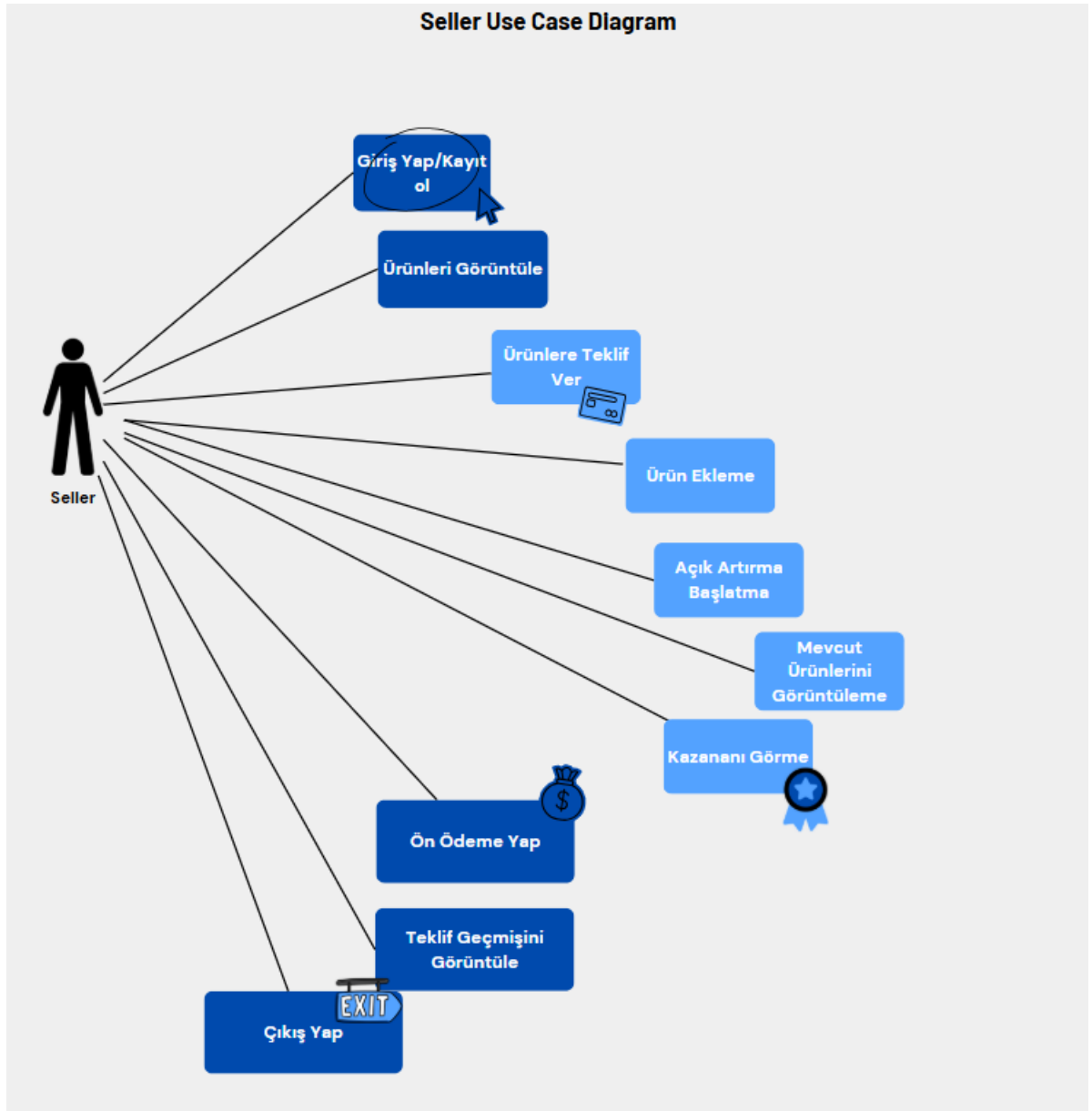
- **Kullanıcı (User):** Sisteme kayıt olan ve ürün arama, teklif verme, satın alma gibi işlemleri gerçekleştiren genel kullanıcı profili.
- **Satıcı (Seller):** Ürün listeleyerek açık artırmaya sunan ve satış sürecini yöneten kullanıcı türü.
- **Yönetici (Admin):** Sistemin genel yönetiminden ve denetiminden sorumlu olan yetkili kullanıcı.



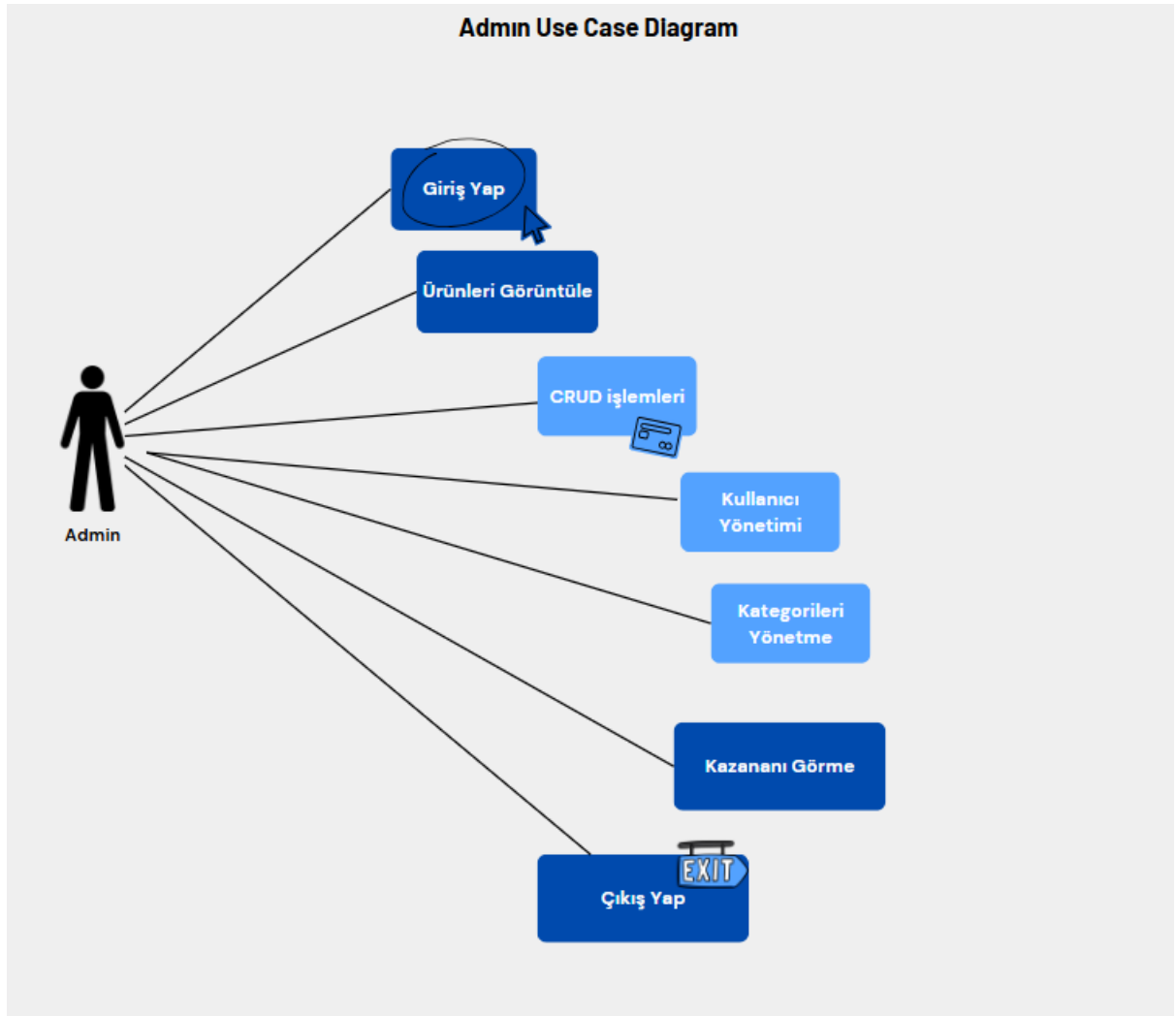
Görsel-7 : Kullanıcıya ait use case diyagramı



Görsel-8 : UML Diyagramı



Görsel-9 : Seller'a ait use case diyagramı



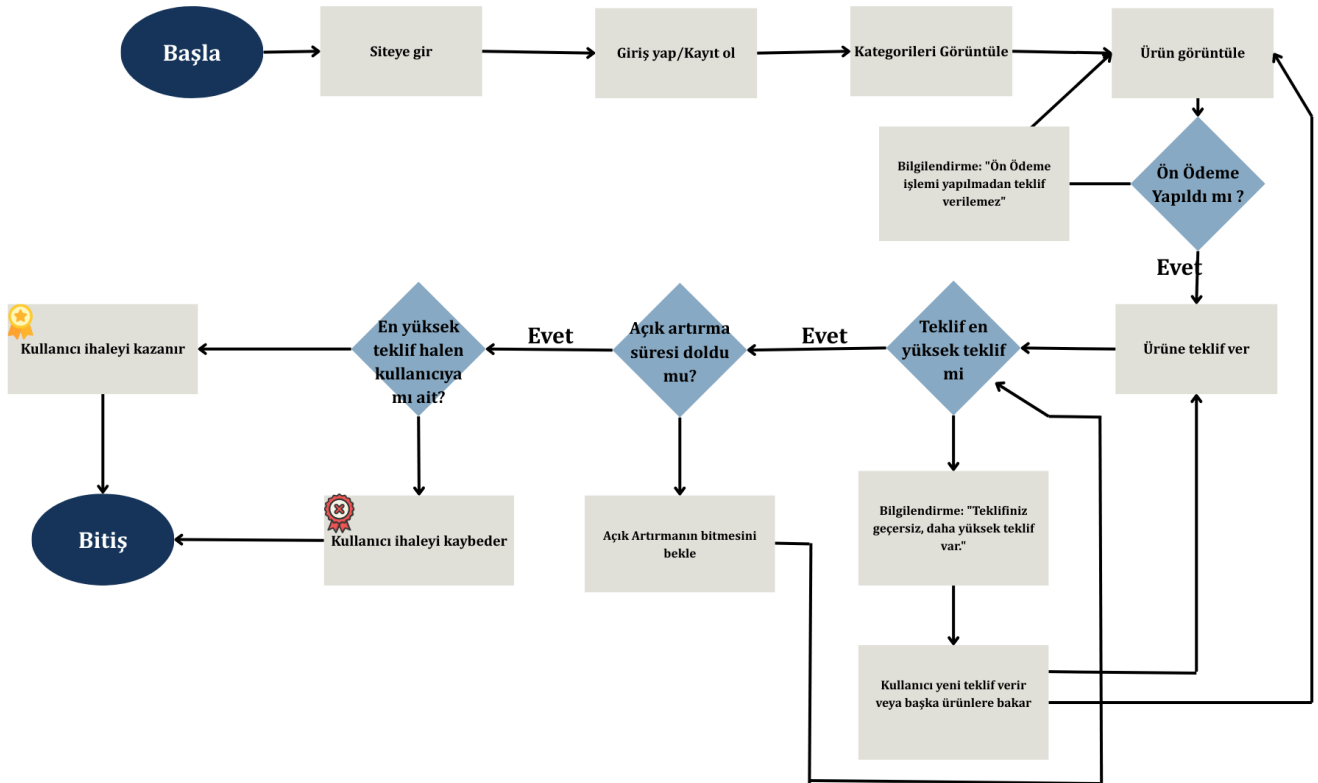
Görsel-10 : Admine ait use case diagramı



## 8.6 AKIŞ ŞEMALARI (FLOWCHARTS)

### Kullanıcı İşlem Akış Şeması (User Interaction Flowchart)

Aşağıda sunulan akış şeması, açık artırma sistemine giriş yapan bir kullanıcının teklif verme süreci içerisindeki olası eylem ve durumları ayrıntılı olarak göstermektedir. Kullanıcı sisteme giriş yaptıktan sonra, kategorileri görüntüleyip ürünleri incelemekte, ardından seçtiği ürüne teklif verebilmektedir. Sistem, teklifin geçerli olup olmadığını kontrol ederek kullanıcıyı bilgilendirmekte; geçersiz teklif durumunda yönlendirme yapmaktadır. Ayrıca açık artırma süresi, teklifin güncelliği, ön ödeme zorunluluğu gibi çeşitli sistem kontrolleri şemada yer almaktadır. Bu yapı, kullanıcı davranışlarının mantıksal bir düzen içerisinde nasıl ilerlediğini görsel olarak ifade ederek hem tasarım sürecine hem de kullanıcı deneyimi analizlerine katkı sağlamaktadır.



Görsel-11: Kullanıcı İşlem Akış Şeması

## 9.Gereksinim Analizi

Bu bölümde, sistemin hangi işlevleri yerine getirmesi gerektiği ve bu işlevleri ne şekilde gerçekleştireceği ayrıntılı olarak açıklanmıştır. Gereksinimler, iki ana başlık altında sınıflandırılmıştır: **Fonksiyonel Gereksinimler** ve **Fonksiyonel Olmayan Gereksinimler**.

### 9.1 Fonksiyonel Gereksinimler

Fonksiyonel gereksinimler, sistemin kullanıcılarla nasıl etkileşime gireceğini ve hangi işlemleri yerine getirmesi gerektiğini tanımlar. Aşağıda, geliştirilen açık artırma sisteminin temel işlevlerine ilişkin gereksinimler tablo hâlinde sunulmuştur:

Kodu	Tanım
FR-01	Kullanıcılar sisteme kayıt olabilmeli ve giriş yapabilmelidir.
FR-02	Satıcılar ürün ekleyebilmeli ve ürün detaylarını düzenleyebilmelidir.
FR-03	Tüm kullanıcılar ürünleri listeleyebilmeli ve filtreleyebilmelidir.
FR-04	Kayıtlı kullanıcılar aktif ürünlere teklif verebilmelidir.
FR-05	Sistem, ürün süresi sona erdiğinde en yüksek teklif sahibini belirlemelidir.
FR-06	Admin kullanıcılar, içerik ve kullanıcı yönetimini gerçekleştirebilmelidir.

Görsel-12: Fonksiyonel Gereksinimlere ait tablo

## 9.2 Fonksiyonel Olmayan Gereksinimler

Fonksiyonel olmayan gereksinimler, sistemin performans, güvenlik, erişilebilirlik ve genel kullanıcı deneyimi açısından nasıl çalışması gerektiğini tanımlar. Yazılımın sürdürülebilirliği ve güvenilirliği açısından bu gereksinimlerin karşılanması kritik öneme sahiptir.

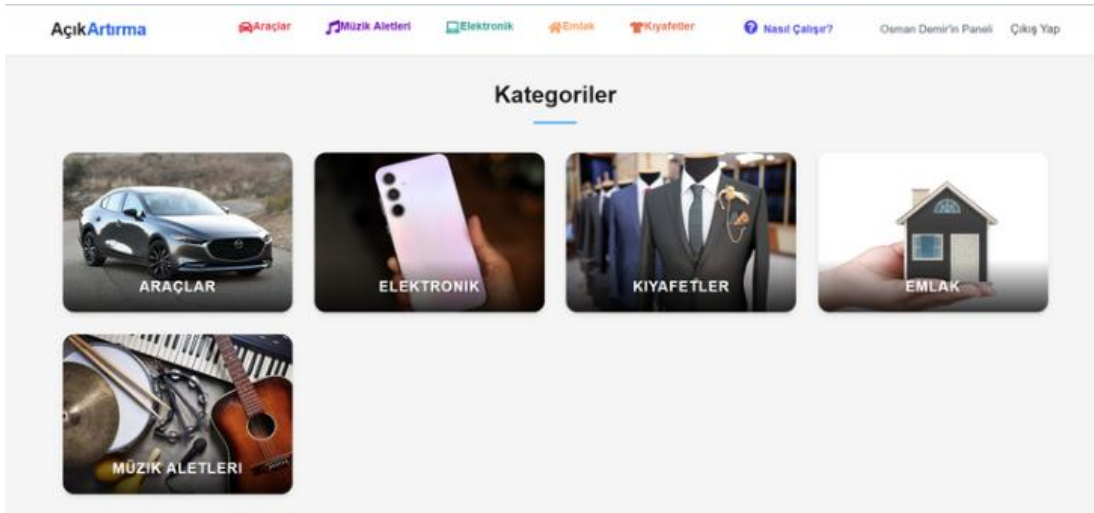
Kodu	Tanım
NFR-01	Sistem 2 saniyeden kısa sürede tepki vermelidir.
NFR-02	Sistem, mobil ve masaüstü tarayıcılarla uyumlu çalışmalıdır.
NFR-03	JWT temelli güvenli oturum yönetimi kullanılmalıdır.
NFR-04	Şifreler hash'lenmeli
NFR-05	Tüm kritik işlemler loglanmalı ve gerektiğinde denetlenebilir olmalıdır.
NFR-06	Yetkisiz kullanıcılar admin sayfalarına erişememeli.

Görsel-13: Fonksiyonel Olmayan Gereksinimlere ait tablo

# 10.Web Arayüzü



Görsel-14 : Ana Sayfa görselimiz



Görsel-15 : Kategoriler Arayüzü

**AçıkArtırma**[Araçlar](#)[Müzik Aletleri](#)[Elektronik](#)[Emlak](#)[Kıyafetler](#)[Nasıl Çalışır?](#)[Giriş Yap](#)[Üye Ol](#)

### Giriş Yap

Email:

Şifre:

[Giriş Yap](#)

Hesabınız yok mu? [Kayıt Ol](#)

Görsel-16 : Giriş Yap/Kayıt Ol sayfamız

AçıkArtırma

Araçlar

Müzik Aletleri

Elektronik

Emlak

Kıyafetler

Nasıl Çalışır?

Giriş Yap

Üye Ol

Tümü


McLaren 720S

Koenigsegg Jesko

Ferrari SF90 Stradale

Pagani Huayra

Daha Fazla Göster



Chevrolet Malibu

Model Yılı: 2017

Renk: Silver

Motor Hacmi: 2.4 L

Kilometre: 28.000 km

Plaka: 34AA21


Well maintained, single owner

Satış Fiyatı: 15.800 TL

Müzayede Başlangıç: 9 TL

Kalan Süre: 59 gün

Teklif Ver



Nissan Altima

Model Yılı: 2017

Renk: Black

Motor Hacmi: 2.5 L

Kilometre: 20.000 km

Plaka: 34AA21


Great condition, low mileage

Satış Fiyatı: 16.000 TL

Müzayede Başlangıç: 9 TL

Kalan Süre: 45 gün

Teklif Ver



Honda Civic

Model Yılı: 2019

Renk: Blue

Motor Hacmi: 1.8 L

Kilometre: 20.000 km

Plaka: 34AA21


Good condition, one previous owner

Satış Fiyatı: 18.000 TL

Müzayede Başlangıç: 9 TL

Kalan Süre: 21 gün

Teklif Ver



Toyota Camry

Model Yılı: 2020

Renk: Silver

Motor Hacmi: 2.5 L

Kilometre: 15.000 km

Plaka: 34AA21

Excellent condition, single owner

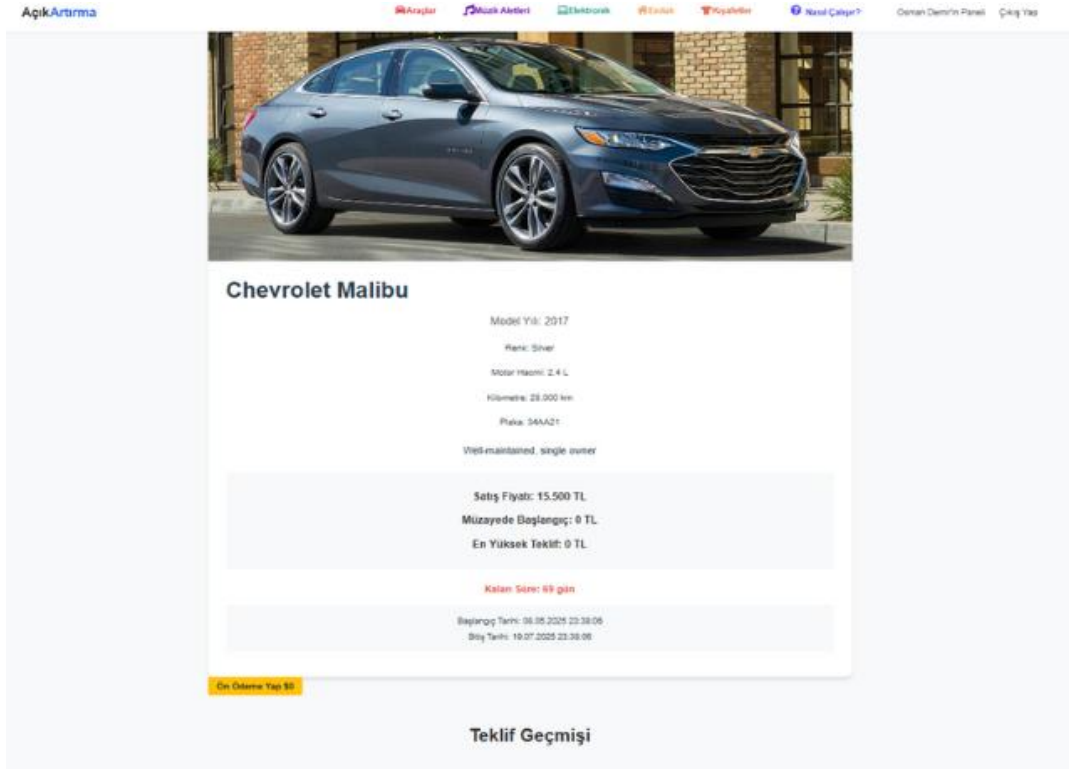
Satış Fiyatı: 25.000 TL

Müzayede Başlangıç: 9 TL

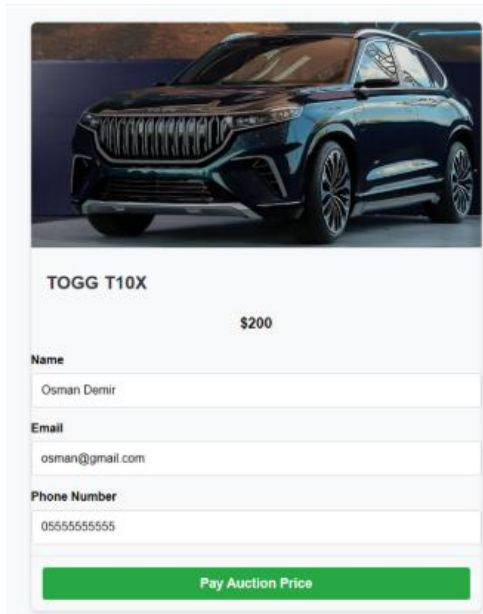
Kalan Süre: 45 gün

Teklif Ver

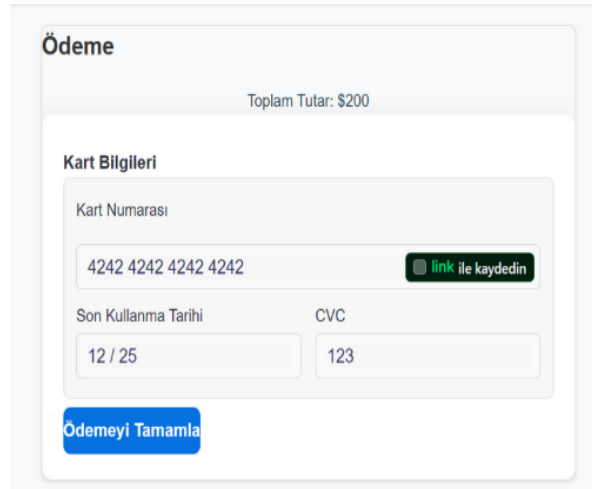
Görsel-17 : Ürünlerin Listelendiği Sayfamız



Görsel-18 : Ürün Detay Sayfamız




Görsel-19 : Ön ödemeye geçiş ekranı



Görsel-20 : Ödeme Ekranı

[AçıkArtırma](#)[Araçlar](#)[Mizik Aletleri](#)[Elektronik](#)[Emlak](#)[Kıyafetler](#)[Nasıl Çalışır?](#)[Osman Demir'in Paneli](#)[Çıkış Yap](#)



## TOGG T10X

Model Yılı: 2025  
Renk: Black  
Motor Hacmi: 2 L  
Kilometre: 0 km  
Plaka: 25AA25

Hatasız Boyasız Tramensiz. Alıcısına Hayırlı Olsun..

Satış Fiyatı: 1.300.000 TL  
En Yüksek Teklif: 2.100.000 TL

Kalan Süre: 31 gün

Başlangıç Tarihi: 14.05.2025 10:02:00  
Bitiş Tarihi: 15.06.2025 10:02:00

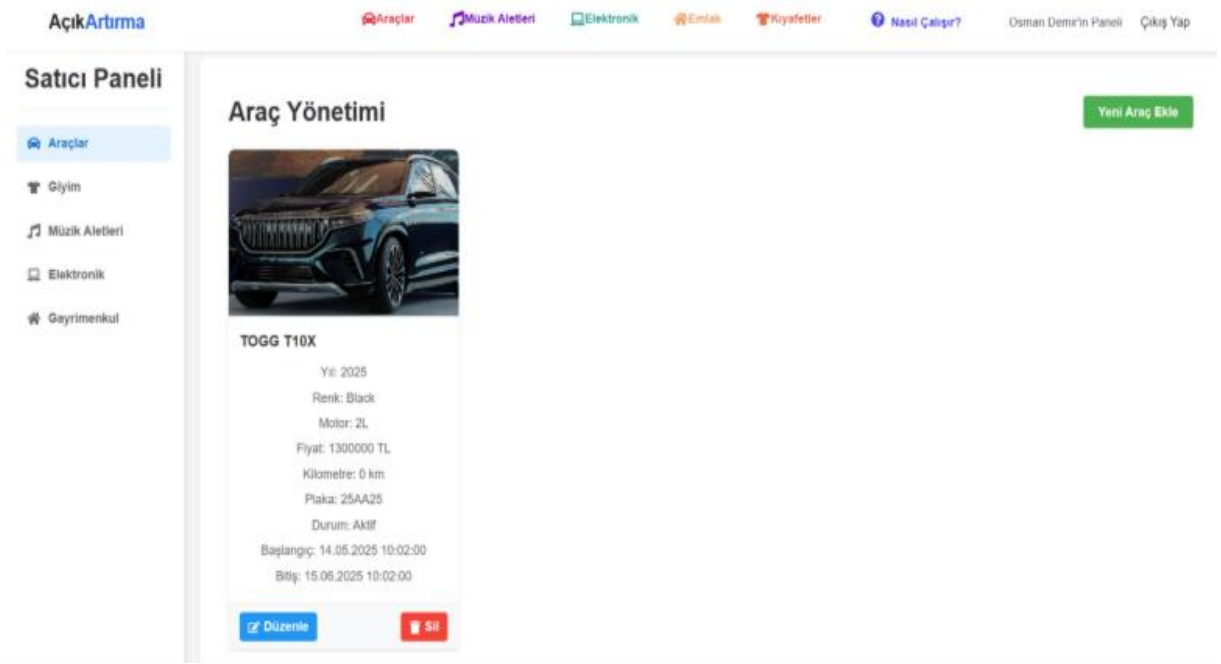
[Teklif Ver](#)

### Teklif Geçmişi

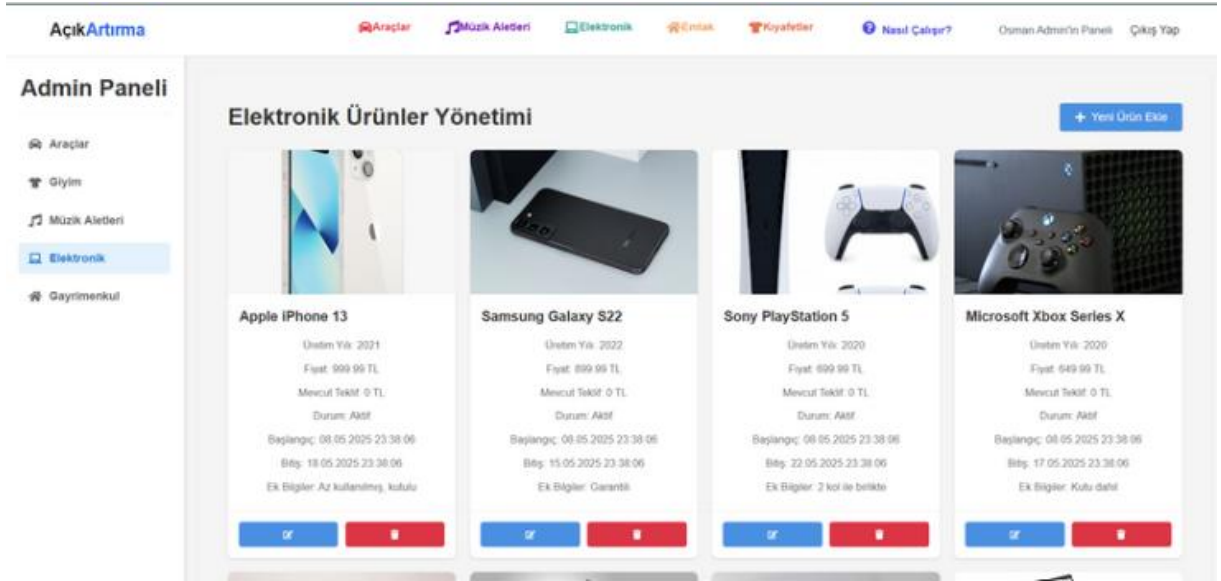
1	14.05.2025 19:12:51 <a href="#">Osman Demir (osman.demir@gmail.com)</a>	\$1350000
2	14.05.2025 19:14:17 <a href="#">Emre Meriç (emre.meric@gmail.com)</a>	\$1400000
3	14.05.2025 19:15:17 <a href="#">Mehmet Ali Nacık (mehmetalinacak@gmail.com)</a>	\$1500000
4	14.05.2025 19:17:07 <a href="#">Deniz (deniz@gmail.com)</a>	\$2100000

Görsel-21 : Teklif Geçmişinin Görüntülendiği Ekran





Görsel-22: Seller Paneli



Görsel-23 : Admin Paneli



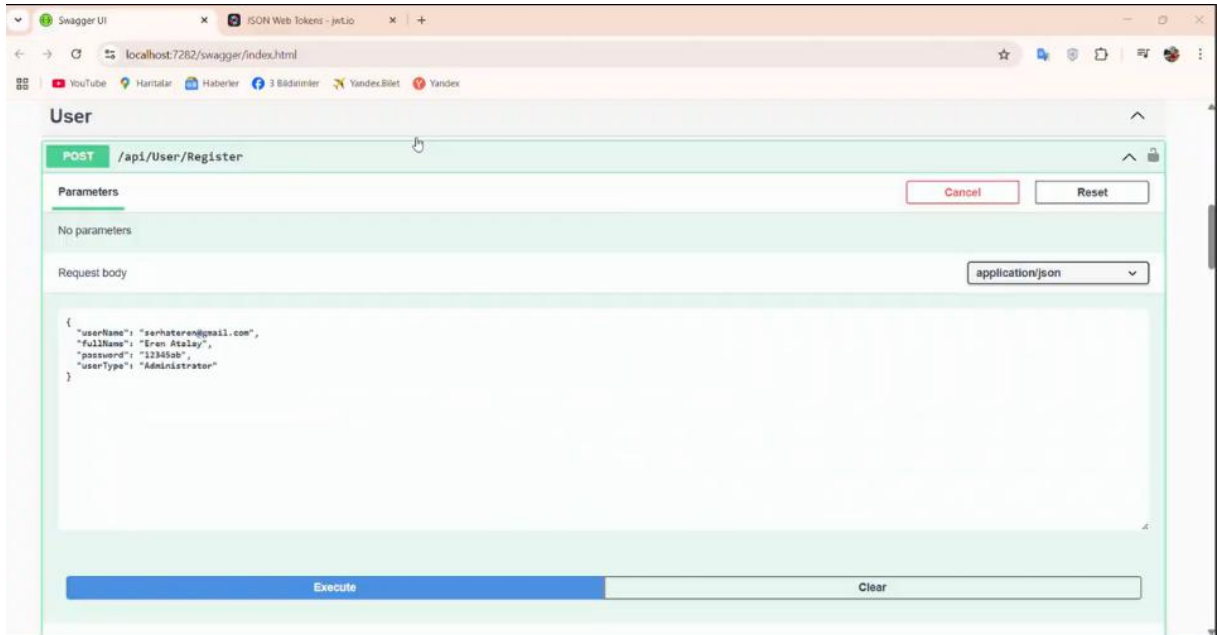


Görsel-24 : Uyarı Bildirimlerine ait görsel

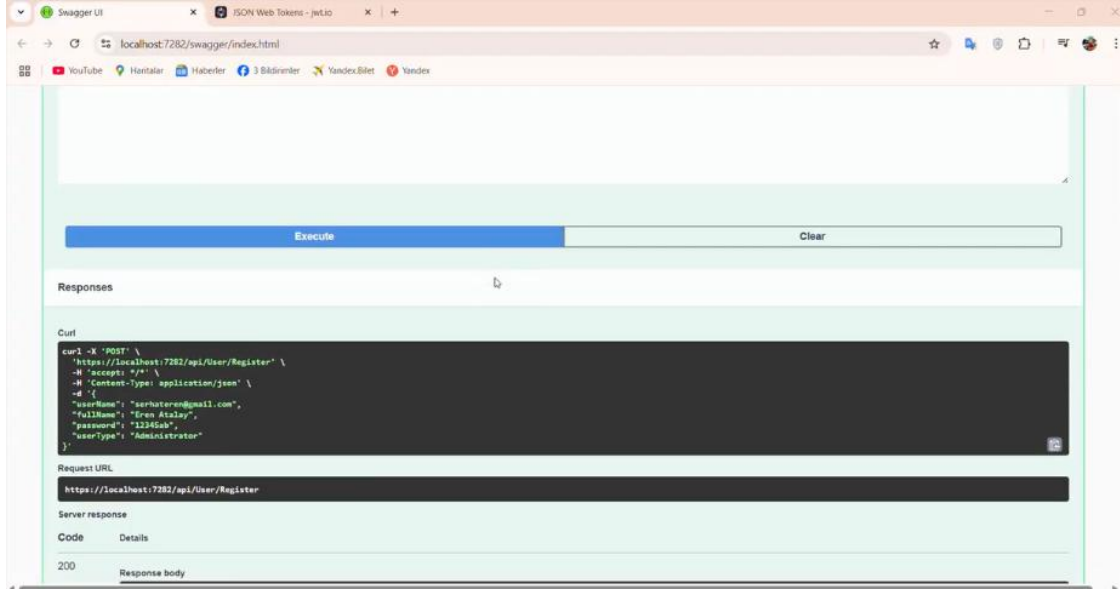
# 11.Test

## 11.1 Back-end Kısımının Testi

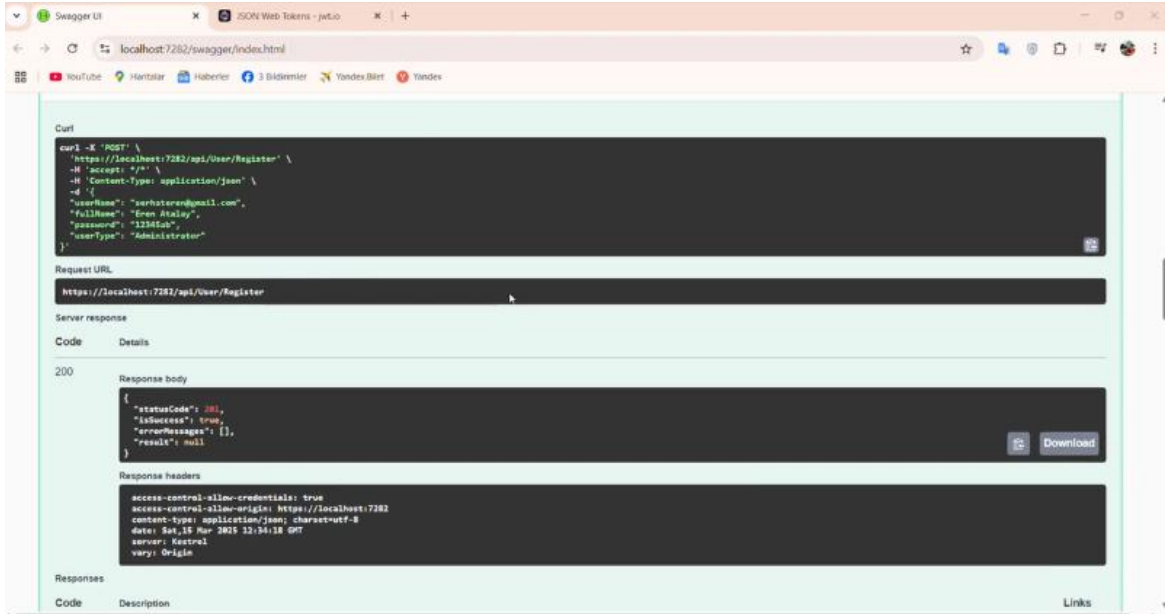
*Backend testleri kapsamında, örnek test görselinde de görülebileceği üzere, manuel olarak Swagger arayüzü üzerinden çeşitli endpoint'leri test ettik. Bu testler sayesinde API'nin doğru çalışıp çalışmadığını gözlemleme fırsatı bulduk. Her bir endpoint'e istek göndererek, beklenen yanıtların alınıp alınmadığını ve sistemin hata durumlarındaki davranışlarını manuel olarak kontrol ettik. Swagger'ın sunduğu kullanıcı dostu arayüz sayesinde, API dokümantasyonu ile test süreci entegre bir şekilde ilerleyerek geliştirme aşamasında önemli geri bildirimler sağladı.*



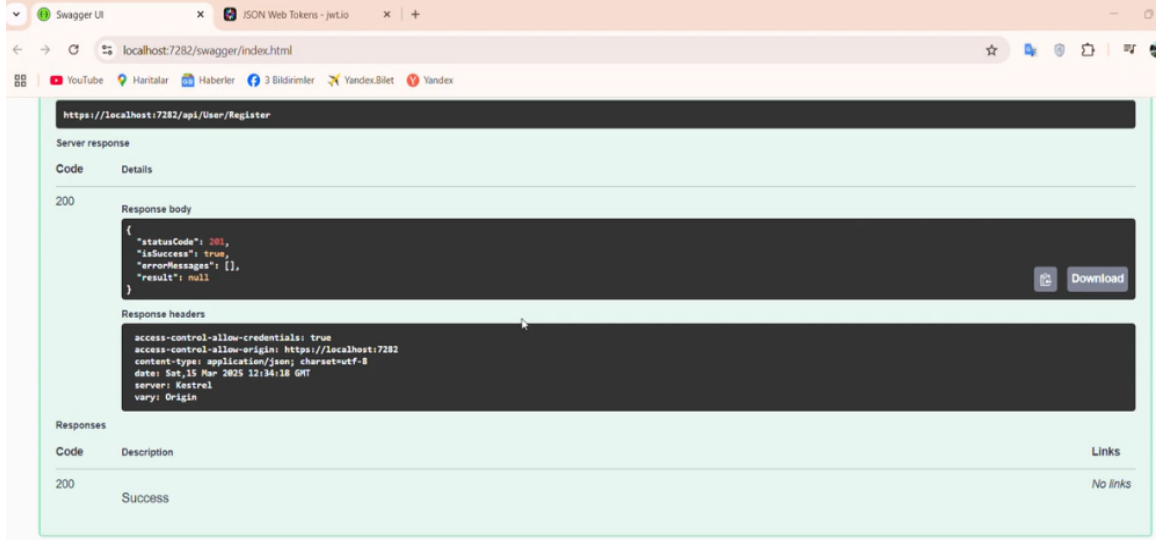
Görsel-25: Örnek Backend Testi-1



Görsel-26 : Örnek Backend Testi-2



Görsel-27: Örnek Backend Testi-3

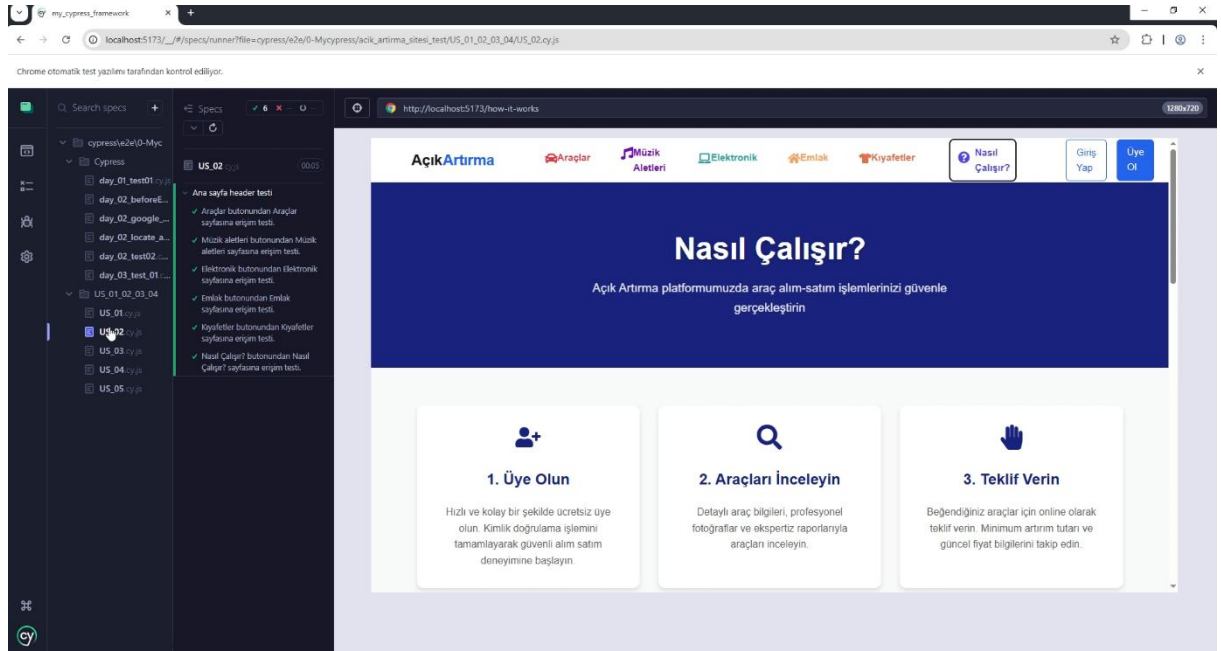


Görsel-28 : Örnek Backend Testi-4

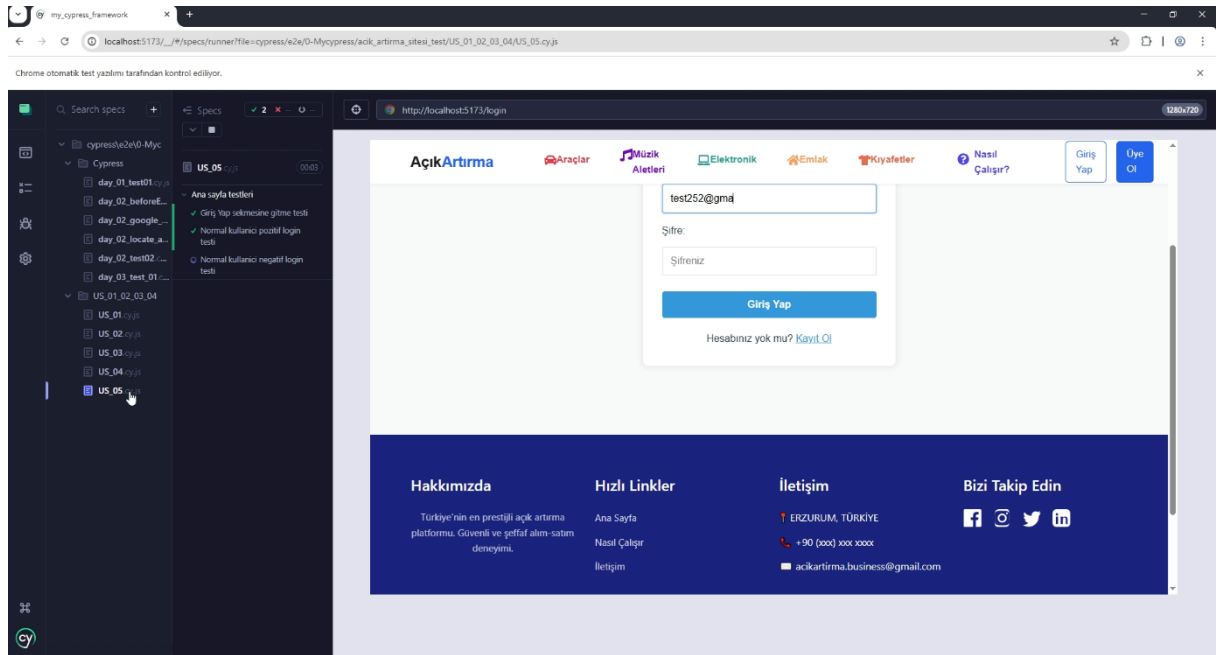
## 11.2 Front-end Kısımının Testi

Frontend testleri kapsamında web uygulamasının kullanıcı arayüzünde gerçekleştirilen işlemlerin doğru şekilde çalışıp çalışmadığını doğrulamak amacıyla Cypress test otomasyon aracı kullanılmıştır. Cypress, modern JavaScript tabanlı uygulamalar için geliştirilmiş güçlü bir test aracıdır ve özellikle etkileşimli web arayüzlerinin güvenilirliğini test etmekte etkilidir. Yapılan testlerde, kullanıcıların sisteme giriş yapması, sayfalar arasında gezinmesi ve sistemin beklenmedik durumlara verdiği tepkiler gibi birçok farklı senaryo dikkate alınmıştır. Bu testler sayesinde kullanıcı deneyimini doğrudan etkileyebilecek olası hatalar önceden tespit edilmiş ve düzeltilmiştir.

Testler sırasında Cypress'in sunduğu gerçek tarayıcı üzerinde çalışma, anlık hata tespiti, ekran kaydı alma ve test senaryolarını adım adım takip etme gibi gelişmiş özelliklerinden faydalanılmıştır. Ayrıca, test sürecinin hızlı ve kolay bir şekilde tekrarlanabilir olması sayesinde sürekli entegrasyon süreçlerine de entegre edilerek yazılım geliştirme sürecine önemli katkılar sağlanmıştır. Genel olarak yapılan bu frontend testleri, web uygulamasının hem işlevselliğini hem de kullanıcı dostu yapısını garanti altına almış, olası kullanıcı hatalarının önüne geçilmesine yardımcı olmuştur.



Görsel-29 : Örnek Front-end Testi-1



Görsel-30 : Örnek Front-end Testi-2