

ConneXion-AI: Akıllı Müşteri Hizmetleri ve Yönetim Platformu

BİTİRME PROJESİ RAPORU

İÇİNDEKİLER

- GİRİŞ 1.1. Projenin Amacı 1.2. Proje Kısıtları
- MATERYAL VE YÖNTEM 2.1. React ve Vite Kavramı 2.2. Redux Toolkit ve RTK Query Yapısı 2.3. Tailwind CSS ve Framer Motion Tasarım Sistemi 2.4. JSON Server (Mock Backend) Entegrasyonu
- KURULUM 3.1. Ön Gereksinimler (Node.js) 3.2. Proje Bağımlılıklarının Yüklenmesi 3.3. Mock Sunucusunun Başlatılması 3.4. İstemci (Frontend) Sunucusunun Başlatılması
- SONUÇ
- KAYNAKÇA

1. GİRİŞ

Bu bölümde projenin amacı ve projede karşılaşılan kısıtlara yer verilmiştir.

1.1. Projenin Amacı

ConneXion-AI, işletmelerin müşteri iletişimini merkezileştirmesi, yapay zeka destekli yanıtlar sunması ve destek taleplerini (ticket) profesyonel bir şekilde uçtan uca yönetmesi için tasarlanmış modern, ölçeklenebilir ve yapay zeka entegreli bir müşteri hizmetleri yönetim paneli ve B2B SaaS platformudur. Temel hedefler şunlardır:

- Merkezeleştirme:** Çoklu kanal (Omnichannel) desteği ile tüm müşteri mesajlarını tek bir "Inbox" ekranına toplamak.
- Otomasyon & Yapay Zeka:** Sık sorulan sorulara AI tabanlı "Akıllı Öneriler" ile anında yanıt oluşturmak.
- İzlenebilirlik:** Destek taleplerinin yaşam döngüsünü başından sonuna kadar şeffaf ve durum (statü) bazlı bir şekilde yönetmek.
- İzolasyon (Çoklu Kiracı):** Kurumsal verilerin Tenant (kiracı) bazlı güvenli bir şekilde ayrıştırılmasını sağlayarak Rol Bazlı Erişim Kontrolü (RBAC) ile veri güvenliğini oluşturmak.

1.2. Proje Kısıtları

- Kısıt 1:** Bu aşamadaki proje kapsamı (Faz 1), gerçek bir veritabanı (PostgreSQL/MongoDB vb.) bağlantısı kurmak yerine bir **Mock Backend (JSON Server)** üzerinden veri işlemektedir. Tüm veriler yerel diskteki `db.json` üzerinden okunur ve yazılır.
- Kısıt 2:** Uygulamada yetkilendirme (Auth) sistemi, parola doğrulama adımımdan ziyade, sisteme gömülü rolleri simüle etmek için e-posta bazlı kontrol sağlamaktadır. Güvenli bir JWT veya OAuth tabanlı mekanizma henüz entegre edilmemiştir.
- Kısıt 3:** WebSocket tabanlı gerçek zamanlı iletişim (Real-Time Messaging) altyapısı yerine, API çağrılarıyla durum yönetiminin güncellenmesini sağlayan **RTK Query önbellekleme (caching)** sistemi kullanılarak gerçek zamanlı mesajlaşma simüle edilmiştir.

2. MATERYAL VE YÖNTEM

Projenin geliştirilmesinde kullanılan modern teknolojiler ve platformun mimari yapıtaşları bu bölümde açıklanmaktadır.

2.1. React ve Vite Kavramı

ConneXion-AI, Single Page Application (SPA) standartlarına uygun olarak modüler mimaride tasarlanmıştır.

- **React (v18):** Kullanıcı arayüzünü bileşenler (components) şeklinde inşa etmek için kullanılmıştır. Dinamik render işlemleri ve durum yönetimini kolaylaştırır.
- **Vite:** Geliştirme sürecinde Hot Module Replacement (HMR) desteği sunan, webpack'e göre çok daha hızlı bir derleyici aracıdır. Projenin modül yönetimini optimize eder.

2.2. Redux Toolkit (RTK) ve RTK Query Yapısı

Global veri yönetimi ve HTTP istekleri için Redux mimarisi kullanılmıştır.

- **Redux Toolkit:** Uygulamanın anlık filtreler, aktif konuşma pencereleri ve kullanıcı oturumu gibi genel durumlarını (state) minimum boilerplate (tekrarlı kod) ile yönetir.
- **RTK Query:** Sistemin "Backend" ile konuşmasını sağlar. Manuel fetch/axios kullanımlarını ortadan kaldırarak verilerin önbelleklenmesini (caching), invalidation (güncelleme) işlemlerini ve performanslı API çağrılarını sağlar.

2.3. Tailwind CSS ve Framer Motion Tasarım Sistemi

Kullanıcı deneyimini en üst seviyeye taşımak için görsel altyapı güçlendirilmiştir.

- **Tailwind CSS:** "Utility-first" yaklaşımı kullanılarak hızlı, tamamen duyarlı (responsive) ve özel (custom) CSS yazmaya gerek bırakmayan arayüz tasarımı sağlanmıştır.
- **Framer Motion:** Menü geçişleri, ekran değişimleri, açılır pencereler (modal) ve akıcı arayüz tepkimeleri için fizik tabanlı animasyonlar sisteme dahil edilmiştir.

2.4. JSON Server (Mock Backend) Entegrasyonu

Proje, frontend ile backend arasındaki etkileşimi test etmek adına RESTful API simülasyonunu JSON Server ile gerçekleştirmektedir. İstemci tarafından gönderilen POST, GET, PUT ve DELETE istekleri `db.json` üzerinden diske kaydedilir, bu sayede uygulamanın tam bağımsız olarak test edilmesi mümkün kılınır.

3. KURULUM

Uygulamanın çalıştırılabilmesi ve konfigürasyon aşamaları bu bölümde belirtilmiştir.

3.1. Ön Gereksinimler (Node.js)

Proje geliştirme ortamının çalışabilmesi için işletim sisteminizde **Node.js** paket yöneticisinin kurulu olması gerekmektedir. İlgili kurulumlar resmi Node.js kaynaklarından (node.js.org) sağlanabilir.

3.2. Proje Bağımlılıklarının Yüklenmesi

Terminal/Komut İstemcisi açılarak projenin kök dizinine (ConneXion-AI) gidilir ve `package.json` dosyasında yer alan gerekli kütüphaneler aşağıdaki komutla projeye dahil edilir:

```
npm install
```

3.3. Mock Sunucusunun Başlatılması

Frontend arayüzünün veri çekebilmesi ve işlemleri kaydedebilmesi için önce JSON Server simülasyonu başlatılmalıdır. Bu komut terminal üzerinde ayrı bir sekmede çalıştırılmalıdır:

```
npm run server
```

Bu komutla sunucu `http://localhost:3000` portundan `db.json` veritabanını yayına alır.

3.4. İstemci (Frontend) Sunucusunun Başlatılması

Mock sunucusu çalışırken yeni bir komut istemcisi/terminal sekmesi açılarak uygulamanın Vite geliştirme sunucusu başlatılır:

```
npm run dev
```

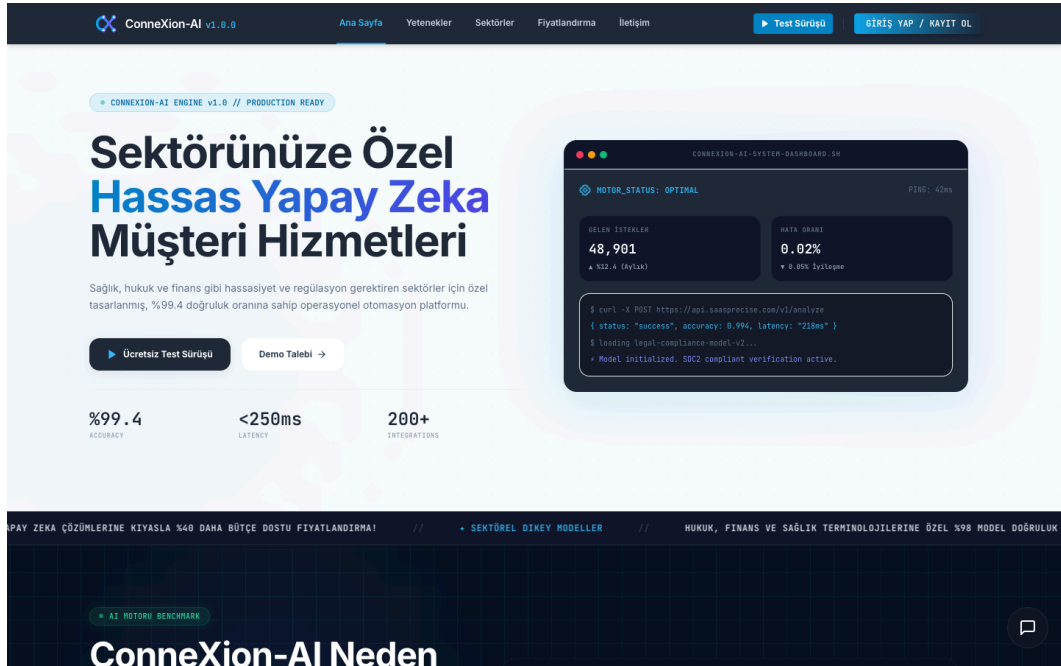
Uygulama başarıyla derlendikten sonra `http://localhost:5173` adresi üzerinden tarayıcıda görüntülenmeye hazır hale gelecektir.

3.5. Uygulama Ekran Görüntüleri

Proje yerel sunucuda ayağa kaldırıldığında erişilebilecek başlıca arayüzler ve işlevleri aşağıda detaylandırılmıştır:

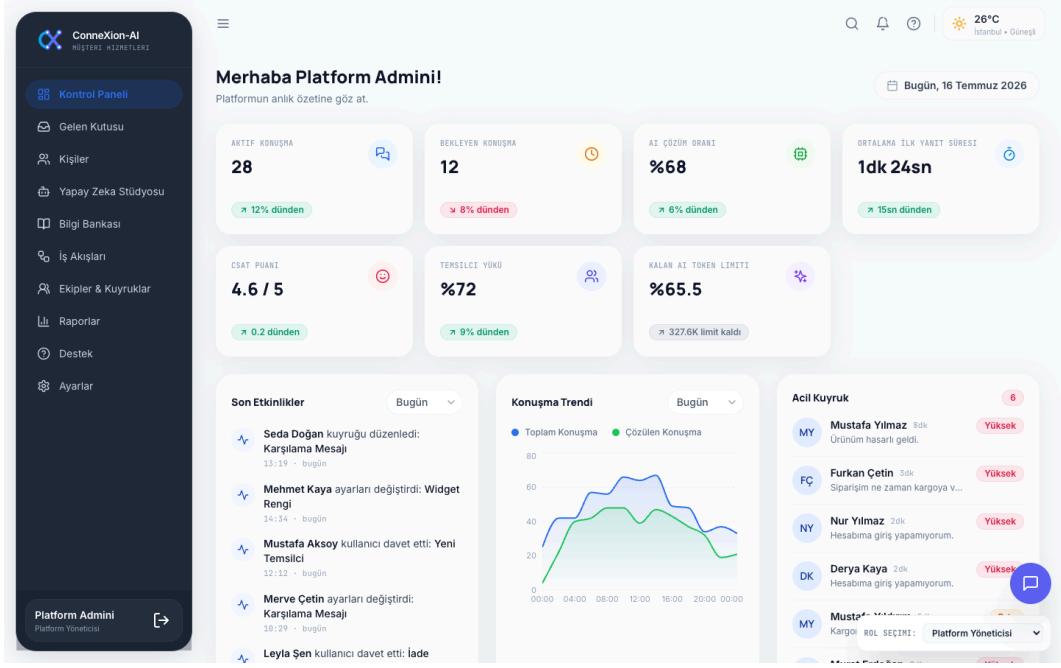
1. Açılış Ekranı (Landing Page)

Uygulamaya ilk giriş yapıldığında kullanıcıyı karşılayan, ürün özelliklerinin tanıtıldığı modern karşılama sayfası.



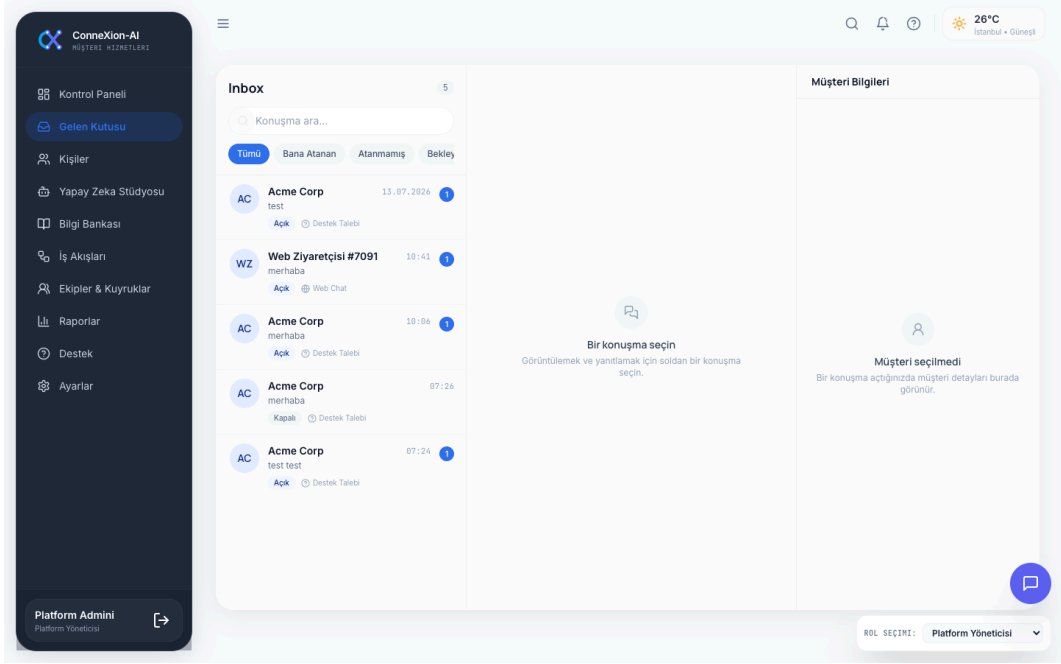
2. Yönetim Paneli (Dashboard)

Platform yöneticileri için genel metriklerin, aktif kullanıcı ve bilet sayılarının özetlendiği analiz paneli.



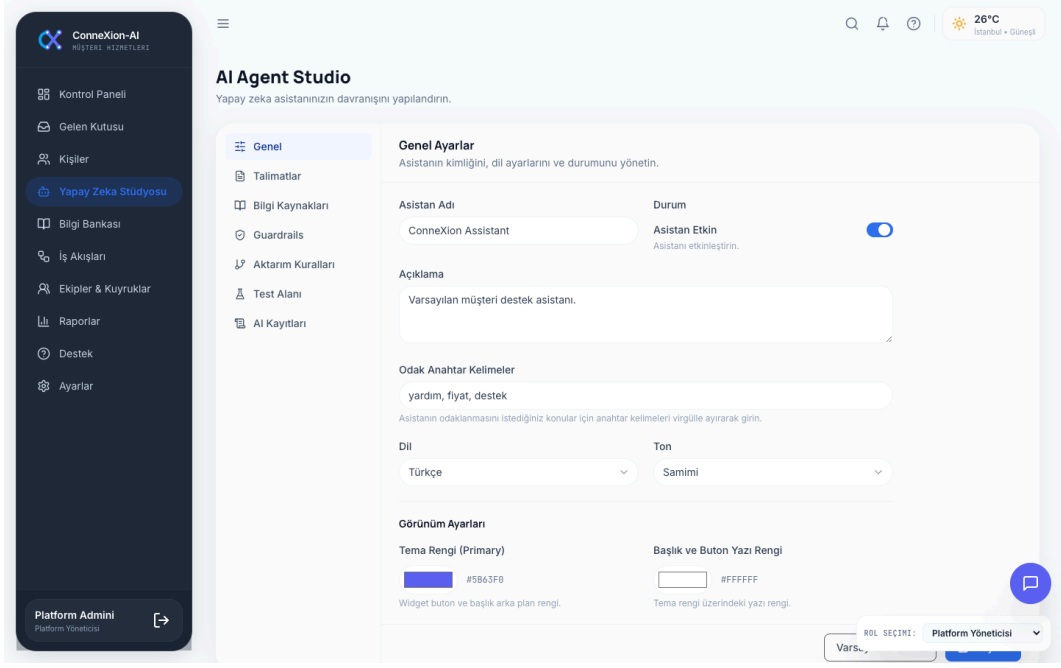
3. Gelen Kutusu (Inbox)

Tüm müşteri mesajlarının (WhatsApp, E-posta, Canlı Destek vb.) tek bir ekranda toplandığı ve temsilcilerin çoklu kanaldan iletişim kurmasını sağlayan merkezi ekran.



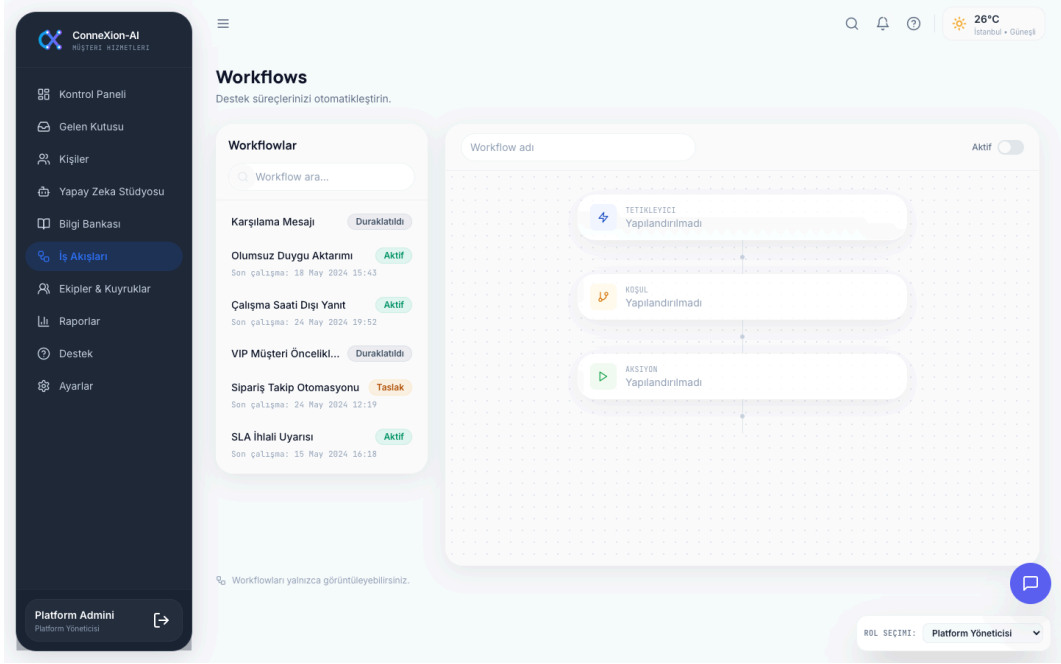
4. Yapay Zeka Ajan Stüdyosu (AI Agent Studio)

Sık sorulan sorular için özel yapay zeka modellerinin eğitilebildiği, test ortamı barındıran stüdyo ekranı.



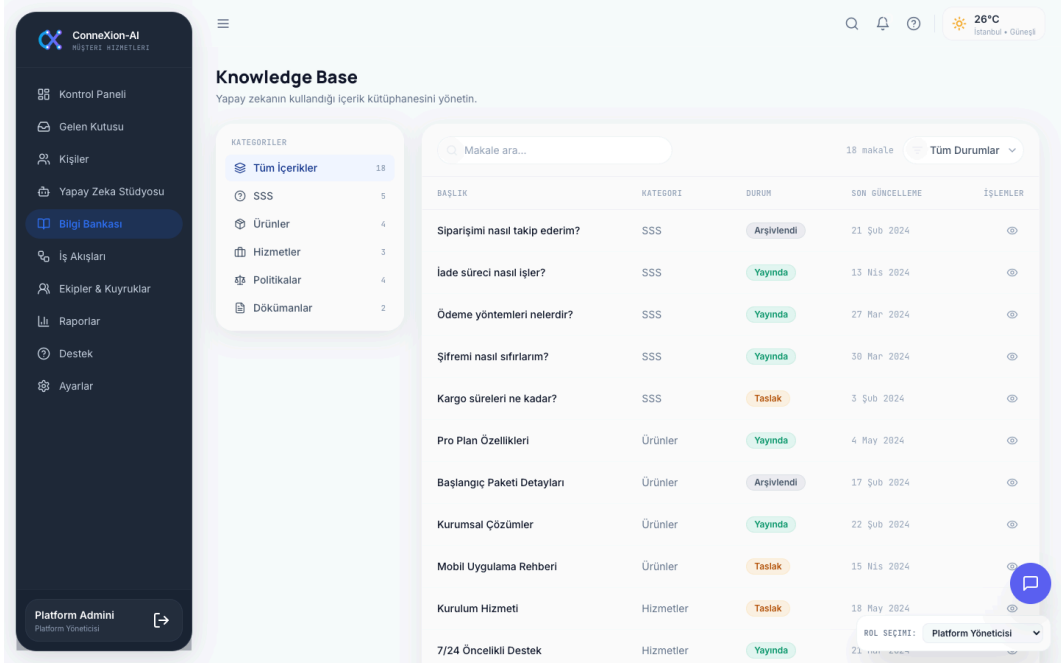
5. İş Akışları (Workflows)

Sürükle bırak mantığıyla müşteri destek süreçlerinin ve otomasyon kurallarının görsel olarak kurgulandığı ekran.



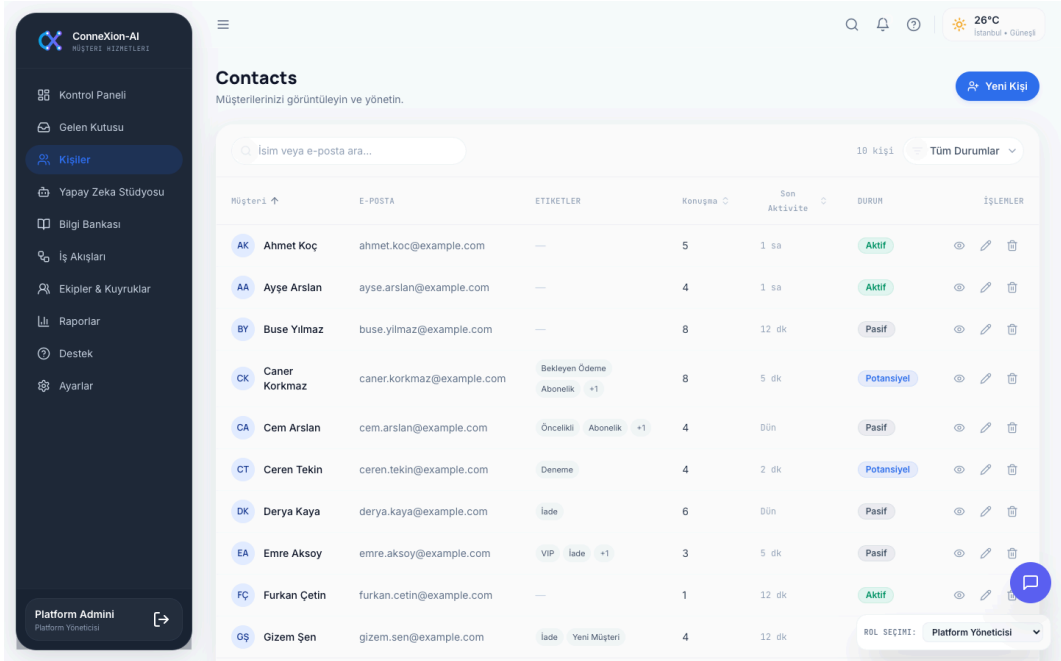
6. Bilgi Bankası (Knowledge Base)

Kurum içi belgelerin ve sıkça sorulan soruların yönetildiği, müşteri temsilcilerine destek amaçlı hazırlanan içerik havuzu.



7. Kişiler (Contacts)

Müşterilerin detaylı profil bilgilerinin, iletişim geçmişinin ve atanmış biletlerin listelendiği CRM ekranı.



8. Ekip Yönetimi (Team)

Destek ekiplerinin (Tier 1, Tier 2 vb.) organize edildiği ve yönlendirme kuyruklarının (queues) belirlendiği alan.

ConneXion-AI

MÜŞTERİ HİZMETLERİ

Kontrol Paneli

Gelen Kutusu

Kişiler

Yapay Zeka Stüdyosu

Bilgi Bankası

İş Akışları

Ekipler & Kuyruklar

Raporlar

Destek

Ayarlar

Platform Admini

Platform Yöneticisi

Team & Queues

Destek ekibinizi, kuyruklarınızı ve atamaları yönetin.

Ekip ÜyeleriKuyruklarPerformans

Üye ara...

Üye Davet Et

ÜYE	RÖL	DURUM	KUYRUK	AKTİF KONUŞMA	İŞLEMLER
AY Ahmet Yılmaz ahmet.yilmaz@acme.com	Platform Yöneticisi	Çevrimiçi	—	0	
ZK Zeynep Kaya zeynep.kaya@acme.com	Workspace Yöneticisi	Çevrimiçi	Genel Destek	2	
MO Mehmet Öztürk mehmet.ozturk@acme.com	Yönetici	Mesgul	Teknik Destek	5	
MA Mustafa Aksoy mustafa.aksoy@acme.com	Destek Temsilcisi	Mesgul	Faturalama	2	
PS Pınar Şen pinar.sen@acme.com	Destek Temsilcisi	Çevrimdışı	Teknik Destek	2	
NA Nur Aslan nur.aslan@acme.com	Destek Temsilcisi	Mesgul	Satış Öncesi	5	
YA Yasemin Aksoy yasemin.aksoy@acme.com	Destek Temsilcisi	Çevrimiçi	Faturalama	5	
LŞ Leyla Şen leyla.sen@acme.com	Destek Temsilcisi	Uzakta	Faturalama	4	
MK Mehmet Kaya mehmet.kaya@acme.com	Destek Temsilcisi	Uzakta	VIP Müşteriler	5	

RÖL SEÇİMİ: Platform Yöneticisi

9. Raporlar (Reports)

Çözüm süreleri, müşteri memnuniyeti ve performans gibi verilerin detaylı grafiklerle raporlandığı ekran.

ConneXion-AI

MÜŞTERİ HİZMETLERİ

Kontrol Paneli

Gelen Kutusu

Kişiler

Yapay Zeka Stüdyosu

Bilgi Bankası

İş Akışları

Ekipler & Kuyruklar

Raporlar

Destek

Ayarlar

Platform Admini

Platform Yöneticisi

Reports

Global Analitik - performans ve trend analizleri

Son 30 Gün

Diş A Aktar

Toplam Konuşma

3.482

AI Çözüm Oranı

%68

Ort. Yanıt Süresi

1dk 24sn

CSAT Puanı

4.6

Kapanan Konuşma

3.104

Konuşma Hacmi

Zaman içinde toplam konuşma sayısı

AI Performansı

AI tarafından çözölen ve aktarılan konuşmalar

Yanıt Süreleri

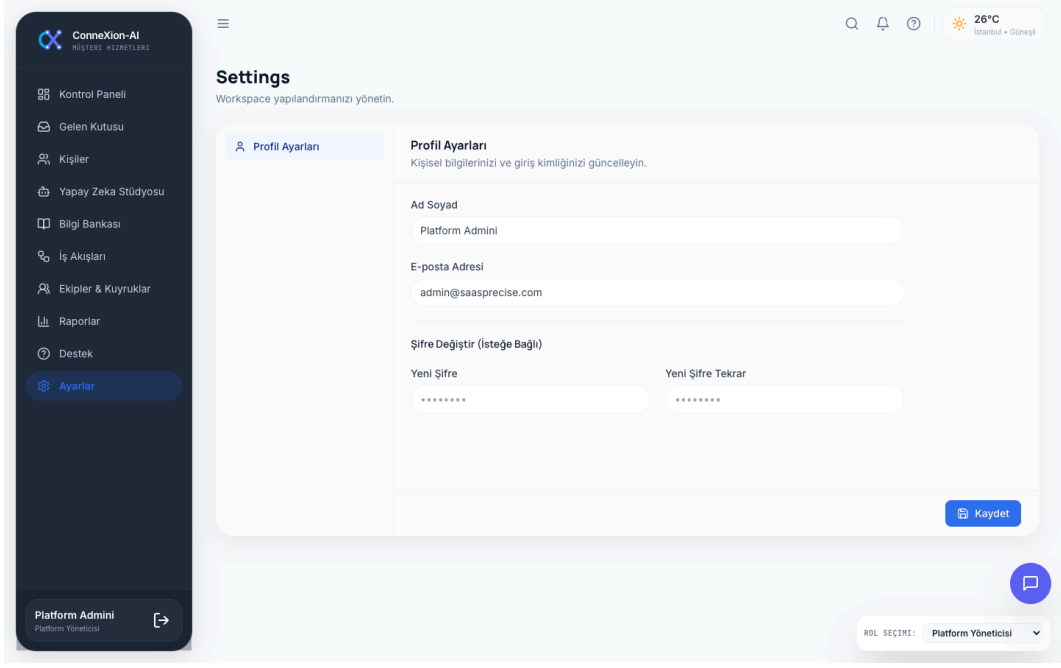
Ortalama ilk yanıt ve çözüm süresi

Müşteri Memnuniyeti

CSAT puanı trendi

10. Ayarlar (Settings)

Sistem tercihlerinin, kullanıcı yetkilerinin (RBAC) ve entegrasyonların yapılandırıldığı yönetim alanı.



4. SONUÇ

Bu projede, modern web teknolojileri (React, Redux, Tailwind) kullanılarak ölçeklenebilir ve yapay zeka odaklı bir B2B Müşteri Hizmetleri Platformu geliştirilmiştir. Rol Bazlı Erişim Kontrolü (RBAC) mekanizması ile Tenant (müşteri) verilerinin izole edilmesi başarmış ve karmaşık durum yönetimleri RTK Query ile sadeleştirilmiştir. Sonuç olarak sadece çalışan bir arayüz değil, aynı zamanda kurumsal sistemlerde karşılaşılabilecek müşteri verisi, performans izleme ve dışa aktarım (PDF/Excel) gibi süreçlerin verimli bir prototipi ortaya koyulmuştur. Kurulum adımları ve mimari, sistemin bir sonraki aşamasında (Gerçek Backend, Veritabanı ve OpenAI entegrasyonu) esnek bir şekilde genişletilebileceğini göstermektedir.

5. KAYNAKÇA

[1] React Documentation: <https://reactjs.org/> [2] Vite Next Generation Frontend Tooling: <https://vitejs.dev/> [3] Redux Toolkit & RTK Query: <https://redux-toolkit.js.org/> [4] Tailwind CSS Framework: <https://tailwindcss.com/> [5] Framer Motion Animation Library: <https://motion.dev/> [6] JSON Server Repository: <https://github.com/typicode/json-server> [7] Recharts, Composed Chart Library: <https://recharts.org/>