



**softITo 4. DÖNEM 1. GRUP**

**FRONTEND GELİŞTİRME BİTİRME PROJESİ RAPORU**

**FlowHub CRM — Rol Bazlı Çok Kanallı Müşteri İlişkileri Yönetimi Uygulaması**

Deniz Çaylı

Ünal Öztürk

Sude Sangür

Eren Akarsu

Dilek İnce

Temmuz, 2026

softITo Yazılım Bilişim Akademisi, İstanbul

# İÇİNDEKİLER

|                                                  |    |
|--------------------------------------------------|----|
| 1. GİRİŞ .....                                   | 3  |
| 1.1. Projenin Amacı .....                        | 3  |
| 1.2. Proje Kısıtları .....                       | 3  |
| 2. MATERYAL VE YÖNTEM .....                      | 3  |
| 2.1. React 18 ve Vite .....                      | 4  |
| 2.2. React Router ve Rota Koruması .....         | 4  |
| 2.3. Redux Toolkit ile Durum Yönetimi .....      | 4  |
| 2.4. ApexCharts ile Görselleştirme .....         | 4  |
| 2.5. Veri Katmanı: db.json ve LocalStorage ..... | 4  |
| 3. UYGULAMA .....                                | 4  |
| 3.1. Giriş Ekranı ve Rol Seçimi .....            | 5  |
| 3.2. Genel Çalışma Mantığı .....                 | 5  |
| 3.3. Panel Yapısı .....                          | 5  |
| 3.3.1. Workspace Owner Paneli .....              | 6  |
| 3.3.2. Admin Paneli .....                        | 6  |
| 3.3.3. Sales Paneli .....                        | 6  |
| 3.3.4. Marketing Paneli .....                    | 6  |
| 3.3.5. Support Paneli .....                      | 6  |
| 3.4. Müşteri Yönetimi .....                      | 7  |
| 3.5. Meta Entegrasyonu .....                     | 8  |
| 3.6. Şikayetvar Entegrasyonu .....               | 8  |
| 3.7. Yapay Zekâ Entegrasyonu .....               | 9  |
| 3.8. Otomasyonlar .....                          | 9  |
| 3.9. Raporlama ve Görselleştirme .....           | 10 |
| 3.10. Ekipler Arası İş Akışı .....               | 10 |
| 3.11. Güvenlik ve Geliştirme Yaklaşımı .....     | 10 |
| 4. SONUÇ .....                                   | 11 |
| 5. KAYNAKÇA .....                                | 11 |

# 1. GİRİŞ

Müşteri ilişkileri yönetimi (CRM) sistemleri, kurumların satış, pazarlama ve destek süreçlerini tek bir veri modeli üzerinde birleştiren temel iş uygulamalarındandır. Günümüzde müşteri iletişimi e-posta, sosyal medya, şikâyet platformları ve web formları gibi birden fazla kanal üzerinden yürütülmekte; bu durum farklı departmanların aynı müşteri verisine kendi rolleri çerçevesinde erişebildiği, çok kanallı ve rol bazlı sistemlere olan ihtiyacı artırmaktadır. Bu çalışmada, söz konusu ihtiyaca yönelik olarak FlowHub CRM adlı rol bazlı, çok kanallı bir müşteri ilişkileri yönetimi uygulaması geliştirilmiştir.

FlowHub CRM; Workspace Owner, Admin, Sales, Marketing ve Support olmak üzere beş farklı kullanıcı rolüyle çalışmakta ve her rolün aynı uygulama içerisinde kendi yetki alanına uygun menülere ve verilere erişmesini sağlamaktadır. Uygulama React 18 ve Vite üzerinde geliştirilmiş; sayfa yönlendirmeleri React Router, oturum ve çalışma alanı yönetimi Redux Toolkit, raporlama grafikleri ise ApexCharts ile gerçekleştirilmiştir. Bu rapor, projenin amacını, kullanılan materyal ve yöntemleri, uygulamanın panel yapısını, dış entegrasyonlarını, yapay zekâ ve otomasyon özelliklerini, veri yönetimi yaklaşımını ve güvenlik uygulamalarını bütünsel biçimde sunmaktadır.

## 1.1. Projenin Amacı

Projenin temel amacı, farklı departmanların müşteri ve operasyon süreçlerini tek merkezden yönetebildiği, gerçek bir CRM sisteminin temel iş mantıklarını yansıtan bir web uygulaması geliştirmektir. Bu kapsamda rol bazlı erişim kontrolü, müşteri ve şirket yönetimi, satış pipeline takibi, çok kanallı gelen kutusu, dış kaynak entegrasyonları, otomasyon kuralları, görev yönetimi, raporlama ve yapay zekâ destekli içerik üretimi gibi modüllerin aynı proje içerisinde bütünsel olarak çalışması hedeflenmiştir.

Projenin bir diğer amacı, ekip üyelerinin modern frontend teknolojilerini gerçek bir ürün senaryosu üzerinde uygulamalı olarak deneyimlemesidir. Bileşen tabanlı mimari, merkezi durum yönetimi, korumalı rotalar, jenerik CRUD servisleri ve harici API tüketimi gibi kavramlar, akademik bir örnek yerine uçtan uca çalışan bir demo ürün üzerinde ele alınmıştır.

## 1.2. Proje Kısıtları

Uygulama bu sürümünde gerçek bir arka uç (backend) sunucusuna bağlanmamakta; başlangıç verileri db.json dosyasından okunarak tarayıcının LocalStorage alanına aktarılmaktadır. Bu nedenle veriler kullanıcıya ve tarayıcıya özeldir; farklı cihazlar arasında senkronizasyon sağlanmamaktadır. Meta entegrasyonu, doğrudan Meta Graph API çağrısı yerine CRM içerisindeki Meta kaynaklı kayıtların yönetimi biçiminde ele alınmıştır. Şikayetvar entegrasyonunda ise Apify actor endpointi yapılandırılmadığı durumlarda proxy üzerinden HTML ayrıştırma yaklaşımına başvurulmaktadır. Yapay zekâ özellikleri Hugging Face Inference API kotalarına bağlıdır ve üretilen içerikler yalnızca taslak niteliğindedir. Uygulama demo verileriyle çalışmakta olup canlı bir üretim ortamına dağıtılmamıştır.

## 2. MATERYAL VE YÖNTEM

Bu bölümde projenin geliştirilmesinde kullanılan temel teknolojiler, kütüphaneler ve veri yönetimi yaklaşımı açıklanmaktadır. Teknoloji seçimlerinde modern frontend ekosisteminde yaygın kabul görmüş, belgeleri güçlü ve ekip üyelerinin eğitim sürecinde deneyim kazandığı araçlar tercih edilmiştir.

### 2.1. React 18 ve Vite

Uygulamanın arayüzü, bileşen tabanlı mimarisiyle React 18 kütüphanesi kullanılarak geliştirilmiştir. React, arayüzün küçük ve yeniden kullanılabilir bileşenlere bölünmesine imkân tanımakta; durum değişikliklerinde yalnızca ilgili bileşenlerin yeniden çizilmesini sağlayarak performans avantajı sunmaktadır. Geliştirme ve derleme aracı olarak Vite tercih edilmiştir. Vite, modül tabanlı geliştirme sunucusu sayesinde anlık yenileme (hot module replacement) sağlamakta ve üretim derlemelerini hızlı biçimde üretmektedir. Özetle React uygulamanın görsel katmanını, Vite ise geliştirme ve derleme altyapısını oluşturmaktadır.

### 2.2. React Router ve Rota Koruması

Sayfa yönlendirmeleri React Router kütüphanesi ile gerçekleştirilmiştir. Uygulamadaki her panel ve alt sayfa ayrı bir rota olarak tanımlanmış; giriş yapan kullanıcının rolüne göre erişebileceği sayfalar ProtectedRoute yapısıyla kontrol altına alınmıştır. Bu yapı sayesinde, örneğin Sales rolüne sahip bir kullanıcının Admin paneline ait bir adrese doğrudan erişmeye çalışması durumunda yönlendirme engellenmekte ve kullanıcı yetkili olduğu ekranlara geri döndürülmektedir. Böylece rol bazlı erişim yalnızca menü görünürlüğüyle değil, rota seviyesinde de güvence altına alınmıştır.

### 2.3. Redux Toolkit ile Durum Yönetimi

Uygulamanın genel durumu Redux Toolkit ile yönetilmektedir. Oturum bilgisi authSlice, aktif çalışma alanı workspaceSlice, mobil kenar çubuğu (sidebar) durumu ise uiSlice içerisinde tutulmaktadır. Kod tabanında admin, sales, marketing veya support rollerine özel ayrı slice yapıları bulunmamakta; roller auth bilgisi, allowedViews tanımları ve sayfa içi rol kontrolleri aracılığıyla ayrıştırılmaktadır. Auth ve aktif çalışma alanı bilgileri redux-persist kütüphanesi sayesinde tarayıcı yenilense dahi korunmakta; bu sayede kullanıcı deneyimi oturum sürekliliği açısından kesintiye uğramamaktadır.

### 2.4. ApexCharts ile Görselleştirme

Dashboard ve rapor sayfalarındaki grafikler ApexCharts kütüphanesi ile oluşturulmuştur. Bar, area, donut ve radial bar grafik tipleri kullanılarak satış performansı, kanal yoğunluğu, abonelik durumu, destek süreci ve gelir trendleri görselleştirilmiştir. Grafiklerin bir bölümü fare ile üzerine gelindiğinde (hover) ayrıntı gösterecek şekilde yapılandırılmış; böylece kullanıcıların özet görünümünden detay bilgiye hızlıca ulaşması sağlanmıştır.

### 2.5. Veri Katmanı: db.json ve LocalStorage

Uygulamadaki başlangıç verileri db.json dosyasından gelmekte ve ilk açılışta tarayıcının LocalStorage alanına kopyalanmaktadır. Bu yaklaşım sayesinde kullanıcı bir şirket eklediğinde, görev tamamladığında, mesaj gönderdiğinde veya fırsat kartını taşıdığına yapılan değişiklikler demo oturumu boyunca korunmaktadır. Veri erişimi jenerik CRUD servisleri üzerinden yürütülmekte;

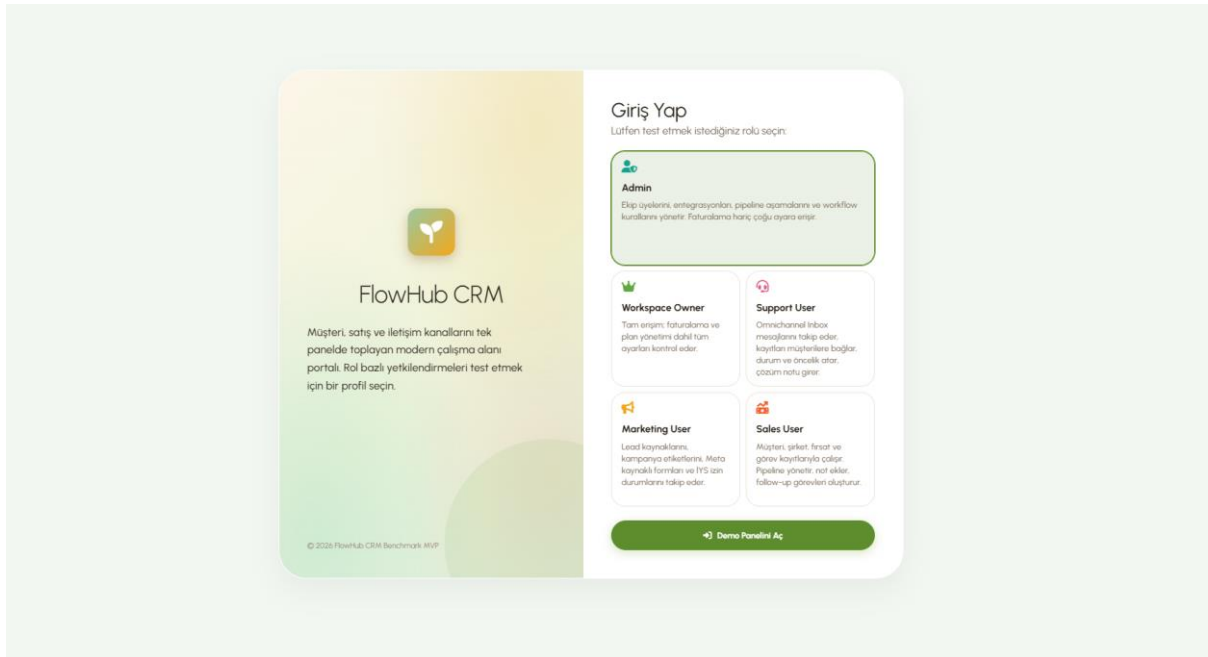
müşteri, şirket, fırsat, konuşma, mesaj, görev, not, kullanıcı, workflow ve workflow run kayıtları benzer create, getAll, update ve remove mantığıyla yönetilmektedir. Veri yapısında workspaceId alanı ile çalışma alanı kapsamı korunmakta; ownerId ve assignedTo alanları ise rol bazlı ekranlarda kullanıcıya ait kayıtların filtrelenmesinde kullanılmaktadır.

### 3. UYGULAMA

Bu bölümde uygulamanın kullanıcı akışı, panel yapısı, modülleri ve entegrasyonları ekran görüntüleri eşliğinde açıklanmaktadır.

#### 3.1. Giriş Ekranı ve Rol Seçimi

Uygulama, kullanıcıyı rol seçim ekranıyla karşılamaktadır. Bu ekranda Admin, Workspace Owner, Support User, Marketing User ve Sales User rolleri, her birinin yetki kapsamını özetleyen kısa açıklamalarla birlikte sunulmaktadır. Kullanıcı test etmek istediği rolü seçtikten sonra Demo Panelini Aç butonu ile ilgili panele yönlendirilmektedir. Bu tasarım, rol bazlı yetkilendirmenin farklı profiller üzerinden kolayca denenebilmesine olanak tanımaktadır. Şekil 3.1.1.'de giriş ve rol seçim ekranı gösterilmektedir.



Şekil 3.1.1. FlowHub CRM Giriş ve Rol Seçim Ekranı

#### 3.2. Genel Çalışma Mantığı

Rol seçiminin ardından kullanıcı rolü authSlice üzerinden belirlenmekte ve tüm rotalar ProtectedRoute yapısıyla korunmaktadır. Aktif çalışma alanı workspaceSlice içerisinde tutulmakta, mobil kenar çubuğu durumu ise uiSlice ile yönetilmektedir. Her rol aynı uygulama içerisinde farklı menülere ve kendi verisine uygun ekranlara erişmektedir. Örneğin Support kullanıcısı destek verdiği müşterileri, Sales kullanıcısı kendi satış fırsatlarını, Admin ise kendisine atanmış müşteri konuşmalarını görmektedir. Bu ayrıştırma, veri modelindeki workspaceId, ownerId ve assignedTo alanlarının rol bazlı ekranlarda filtre olarak kullanılmasıyla sağlanmaktadır.

### 3.3. Panel Yapısı

Uygulamada her rol için ayrı bir panel bulunmakta; paneller ortak CRM verisi üzerinde çalışmakla birlikte içerik ve menü yapısı bakımından birbirinden ayrılmaktadır. Aşağıda her panelin işlevi ayrı başlıklar altında açıklanmaktadır.

#### 3.3.1. Workspace Owner Paneli

Workspace Owner paneli; ekip, müşteri, görev, departman fırsatları, raporlar ve operasyon yönetimi için kullanılmaktadır. Bu panelde yer alan departman bazlı fırsat filtreleri sayesinde Satış, Support ve Marketing süreçleri ayrı ayrı izlenebilmektedir. Workspace Owner, faturalama ve plan yönetimi dahil tüm ayarları kontrol edebilen en geniş yetkili roldür ve diğer departmanları tek panel üzerinden üstten görebilmektedir. Panelde ayrıca Son 7 Gün, Son 30 Gün ve Tüm Zamanlar seçenekleriyle çalışan dönem filtresi bulunmakta; bu filtre satış, açık fırsat, gelir ve kanal yoğunluğu gibi metriklerin görünümünü değiştirmektedir.

#### 3.3.2. Admin Paneli

Admin paneli, CRM'i kullanan şirketlerin SaaS abonelik bilgilerini, ekip büyüklüklerini, aylık yinelenen gelir (MRR) değerlerini ve rapor verilerini göstermektedir. Admin rolü; ekip üyelerini, entegrasyonları, pipeline aşamalarını ve workflow kurallarını yönetebilmekte, faturalama hariç çoğu ayara erişebilmektedir. Admin Inbox ekranı yalnızca admin müşterilerinin konuşmalarını içerecek şekilde filtrelenmiştir. Böylece aynı uygulama içerisinde hem operasyonel ekip görünümü hem de şirket ve abonelik odaklı yönetim görünümü birbirinden ayrılmıştır.

#### 3.3.3. Sales Paneli

Sales paneli; müşteri, şirket ve satış pipeline yönetimine odaklanmaktadır. Satış kullanıcısı müşteri, şirket, fırsat ve görev kayıtlarıyla çalışmakta; pipeline yönetimi, not ekleme ve takip (follow-up) görevleri oluşturma işlemlerini gerçekleştirmektedir. Fırsat kartları kanban görünümü üzerinde sürükleyip bırak yöntemiyle satış aşamaları arasında taşınabilmekte; kart taşındığında ilgili değişiklik LocalStorage üzerindeki CRM verisine anında yansımaktadır.

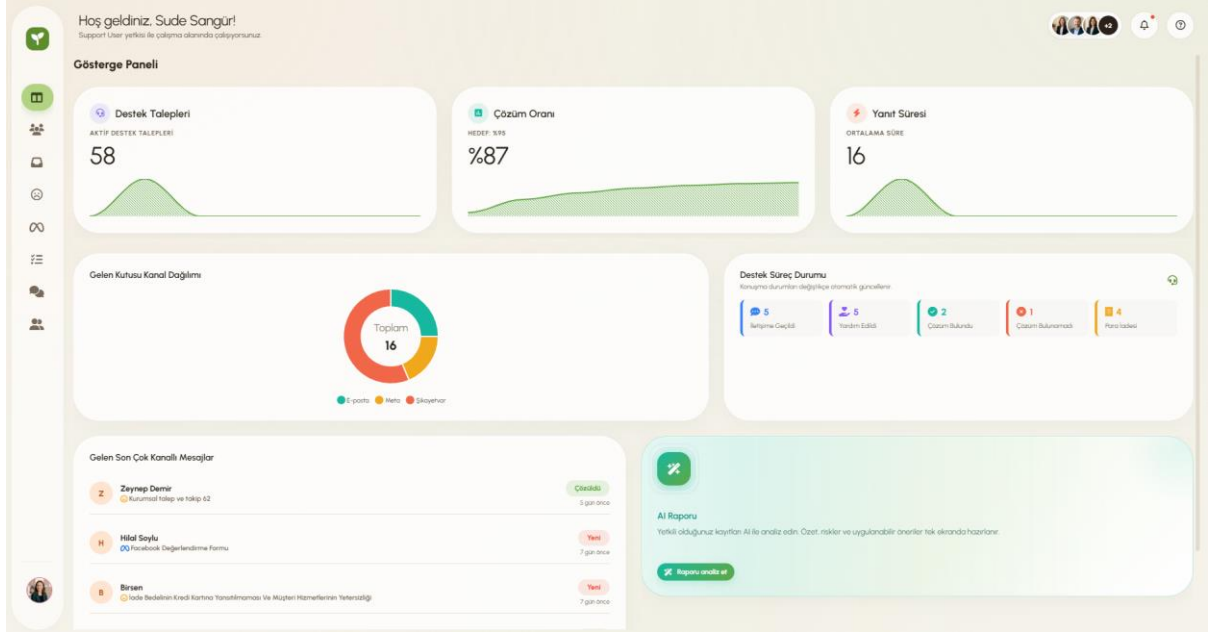
#### 3.3.4. Marketing Paneli

Marketing paneli; Meta kaynakları, etkinlikler, PR paketleri ve pazarlama aktiviteleri üzerinden çalışmaktadır. Pazarlama kullanıcısı lead kaynaklarını, kampanya etiketlerini, Meta kaynaklı form başvurularını ve İYS (İleti Yönetim Sistemi) izin durumlarını takip etmektedir. Marketing fırsatları, satış aşamaları yerine iletişim, etkinlik, PR paketi ve satış gibi pazarlama süreçlerine özgü kolonlarla gösterilmekte; böylece pazarlama ekibinin iş akışı satış ekibinden bağımsız biçimde modellenmektedir.

#### 3.3.5. Support Paneli

Support paneli; çok kanallı gelen kutusu (Omnichannel Inbox), Şikayetvar kayıtları, destek müşterileri ve çözüm durumlarına odaklanmaktadır. Destek kullanıcısı gelen mesajları takip etmekte, kayıtları müşterilere bağlamakta, durum ve öncelik atamakta ve çözüm notu girmektedir. Support süreç kolonları şikayet, iletişim, sorun devam ediyor ve çözüldü biçiminde tanımlanmıştır. Şekil 3.3.5.1.'de Support kullanıcısının gösterge paneli görülmektedir. Panelde aktif destek talepleri, çözüm oranı ve ortalama

yanıt süresi metrikleri, gelen kutusu kanal dağılımı, destek süreç durumu ve son çok kanallı mesajlar ile yapay zekâ destekli rapor analizi alanı bir arada sunulmaktadır.



Şekil 3.3.5.1. Support Rolü Gösterge Paneli ve AI Raporu Alanı

### 3.4. Müşteri Yönetimi

Müşteriler modülü, CRM'in temel veri varlığı olan müşteri kayıtlarının listelendiği, filtrelendiği ve yönetildiği ekrandır. Müşteri listesi; ad soyad, şirket, unvan, e-posta, telefon, durum, kaynak, İYS izni, kampanya etiketi ve sorumlu temsilci alanlarını içermektedir. Kayıtlar aşama, kaynak ve İYS izin durumuna göre filtrelenebilmekte; arama kutusu üzerinden isim bazlı sorgulama yapılabilen ve görünüm tablo ile kart biçimleri arasında değiştirilebilmektedir. Her müşteri kaydında kaynak bilgisi (Manuel, E-posta, Şikayetvar, Meta Ads) tutulmakta; bu bilgi çok kanallı yapının müşteri seviyesinde izlenmesini sağlamaktadır

Hoş geldiniz, Deniz Çaylı! Workshop Owner yetkisi ile çalışma alanında çalışabilirsiniz.

Müşteriler 153 Müşteri

Müşterilerde arayın...

Filtreler: Aşama (Tümü) Kaynak (Tümü) İYS İzin Durumu (Tümü) Filtreleri Temizle

| AD SOYAD         | ŞİRKET                    | UNVAN                    | E-POSTA                               | TELEFON           | DURUM         | KAYNAK     | İYS İZİNİ  | KAMPANYA ETİKETİ | SORUMLU     |
|------------------|---------------------------|--------------------------|---------------------------------------|-------------------|---------------|------------|------------|------------------|-------------|
| Fatma Çelik      | Özürk Medya Holding       | Genel Müdür              | fatma.celik@ozurkmedya.com            | +90 531 101 11 17 | Bekliyor      | Manuel     | Bekliyor   | Meta Form        | Elif İnce   |
| Mustafa Yıldırım | Anısan Elektronik A.Ş.    | Pazarlama Yöneticisi     | mustafa.yildirim@anisanelektronik.com | +90 532 102 12 24 | Aktif Müşteri | Şikayetvar | Reddedildi | Kara Cuma        | Sude Sangır |
| Elif Özdemir     | Aslan İnşaat Ltd. Şti.    | İK Müdürü                | elif.ozdemir@aslaninsoat.com          | +90 533 103 13 31 | Vazgeçti      | E-posta    | Onaylı     | B2B VIP          | Enen Akarsu |
| Hüseyin Kılıç    | Koç Sağlık A.Ş.           | Proje Yöneticisi         | huseyin.kilic@kocsaaglik.com          | +90 534 104 14 38 | Takip         | Meta Ads   | Bekliyor   | Referans         | Deniz Çaylı |
| Selin Kara       | Şişmek Sanayi Ltd. Şti.   | Lojistik Şefi            | selin.kara@sismeksanayi.com           | +90 535 105 15 45 | Bekliyor      | Manuel     | Reddedildi | Kurumsal         | Elif İnce   |
| İbrahim Özkan    | Aydemir Teknoloji Holding | İdari İşler Müdürü       | ibrahim.ozkan@aydemirteknoloji.com    | +90 536 106 16 52 | Aktif Müşteri | Şikayetvar | Onaylı     | Yeni Pazarlama   | Sude Sangır |
| Gamze Korkmaz    | Aksay Tekstil A.Ş.        | COO                      | gamze.korkmaz@aksaytektstil.com       | +90 537 107 17 59 | Vazgeçti      | E-posta    | Bekliyor   | Kurumsal Anlayış | Enen Akarsu |
| Yusuf Aktaş      | San Danışmanlık Ltd. Şti. | Dijital Pazarlama Uzmanı | yusuf.aktas@санданışманlık.com        | +90 538 108 18 66 | Takip         | Meta Ads   | Reddedildi | Webinar          | Deniz Çaylı |
| Aslı Güneş       | Acar Yapı A.Ş.            | BT Direktörü             | asli.gunes@acar-yapi.com              | +90 530 109 19 73 | Bekliyor      | Manuel     | Onaylı     | Sadakat Programı | Elif İnce   |

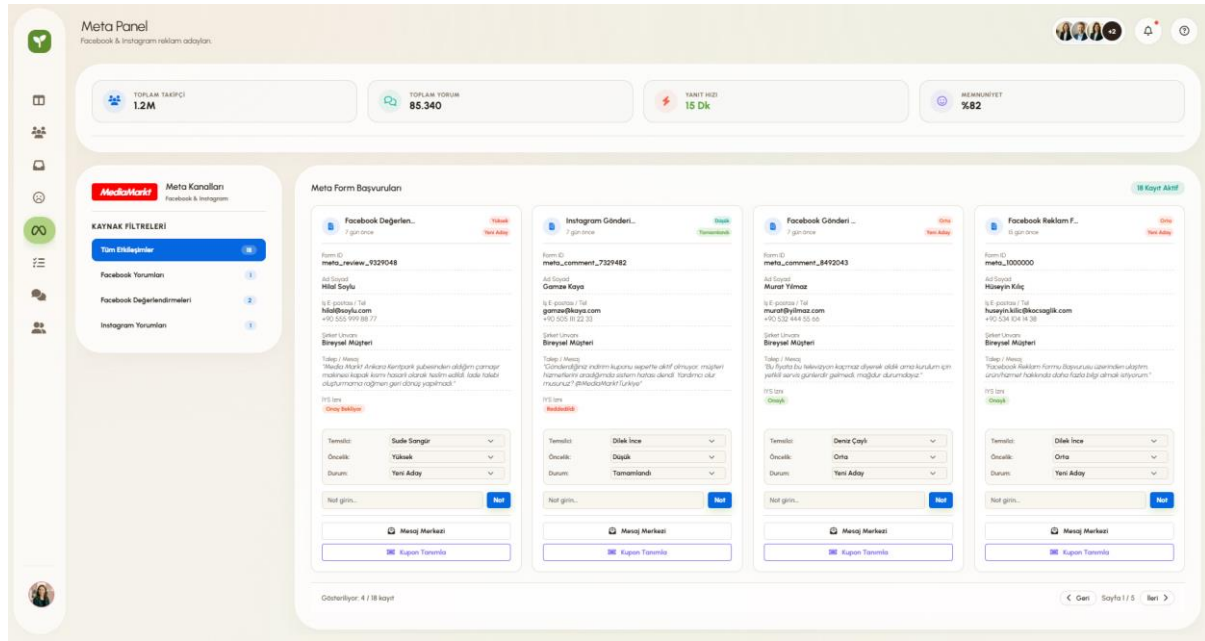
Çözümleniyor: 40 / 153 kayıt

< Geri Sayfa 1 / 4 İleri >

Şekil 3.4.1. Müşteriler Ekranı ve Filtreleme Yapısı

### 3.5. Meta Entegrasyonu

Meta paneli, uygulamadaki Meta kanallı CRM konuşmalarını ve form başvurularını listelemektedir. Panelde Facebook yorumları, Facebook değerlendirmeleri ve Instagram yorumları kaynak filtreleri ile ayrıştırılabilmekte; toplam takipçi, toplam yorum, yanıt hızı ve memnuniyet gibi özet metrikler üst bölümde sunulmaktadır. Her form başvurusu kartında form kimliği, ad soyad, iletişim bilgileri, talep mesajı ve İYS izin durumu görüntülenmekte; temsilci atama, öncelik ve durum değiştirme, not ekleme, kupon tanımlama ve ilgili Inbox konuşmasına gitme işlemleri yapılabilmektedir. Kod tarafında doğrudan Meta Graph API çağrısı yerine, CRM içerisindeki Meta kaynaklı kayıtların yönetimi gerçekleştirilmektedir. Şekil 3.5.1.'de Meta paneli gösterilmektedir.

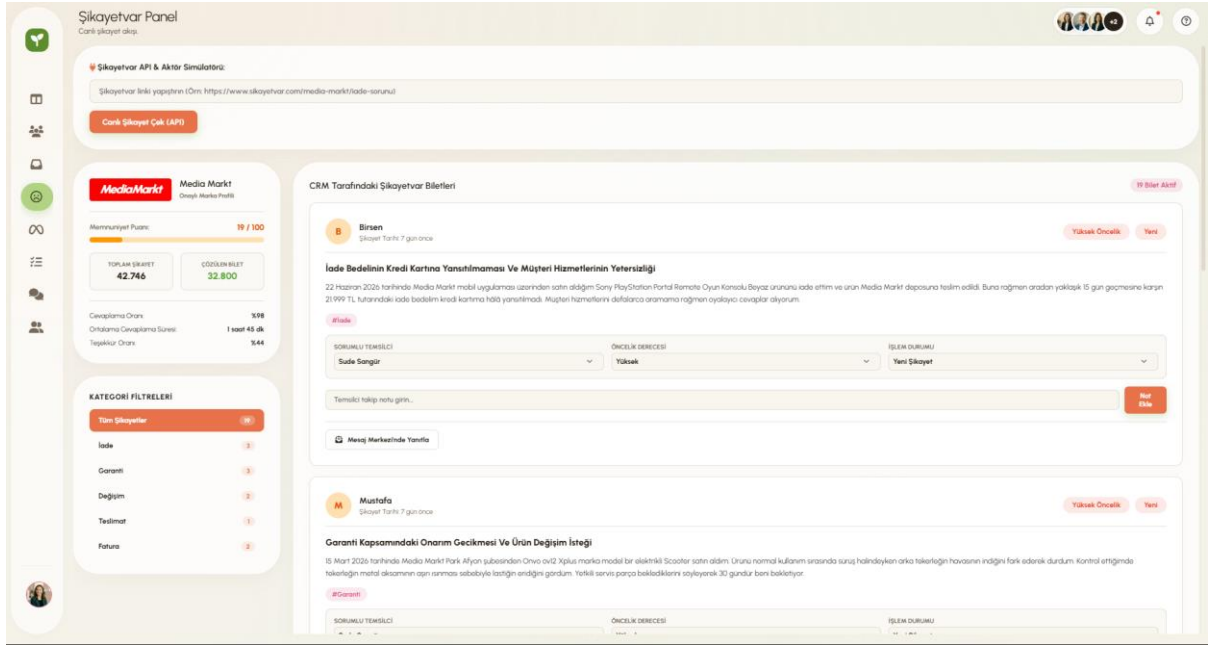


Şekil 3.5.1. Meta Paneli ve Form Başvuruları Ekranı

### 3.6. Şikayetvar Entegrasyonu

Şikayetvar paneli, URL tabanlı bir veri alma akışına sahiptir. Kullanıcı bir Şikayetvar linki girdiğinde uygulama öncelikle Apify actor endpointini kullanmayı denemektedir. Bu endpoint yapılandırılmamışsa ya da hata vermesi durumunda, proxy üzerinden ilgili sayfanın HTML içeriği alınmakta ve temel şikâyet alanları ayrıştırılmaktadır. Bu nedenle Şikayetvar tarafında hem API kullanımı hem de yedek (fallback) web kazıma yaklaşımı bir arada bulunmaktadır.

Alınan şikâyet kaydı CRM konuşmasına dönüştürülmekte; aktif bir workflow bulunması hâlinde çalışma kaydı ve takip görevi otomatik olarak üretilmektedir. Panelde marka profili, memnuniyet puanı, toplam şikâyet ve çözölen bilet sayıları ile cevaplama oranı gibi metrikler sol bölümde sunulmakta; şikâyetler iade, garanti, değişim, teslimat ve fatura gibi kategorilere göre filtrelenebilmektedir. Her bilet üzerinde sorumlu temsilci, öncelik derecesi ve işlem durumu atanabilmekte, takip notu girilebilmekte ve Mesaj Merkezi üzerinden yanıt verilebilmektedir. Şekil 3.6.1.'de Şikayetvar paneli görölmektedir.



Şekil 3.6.1. Şikayetvar Paneli ve Canlı Şikâyet Çekme Akışı

### 3.7. Yapay Zekâ Entegrasyonu

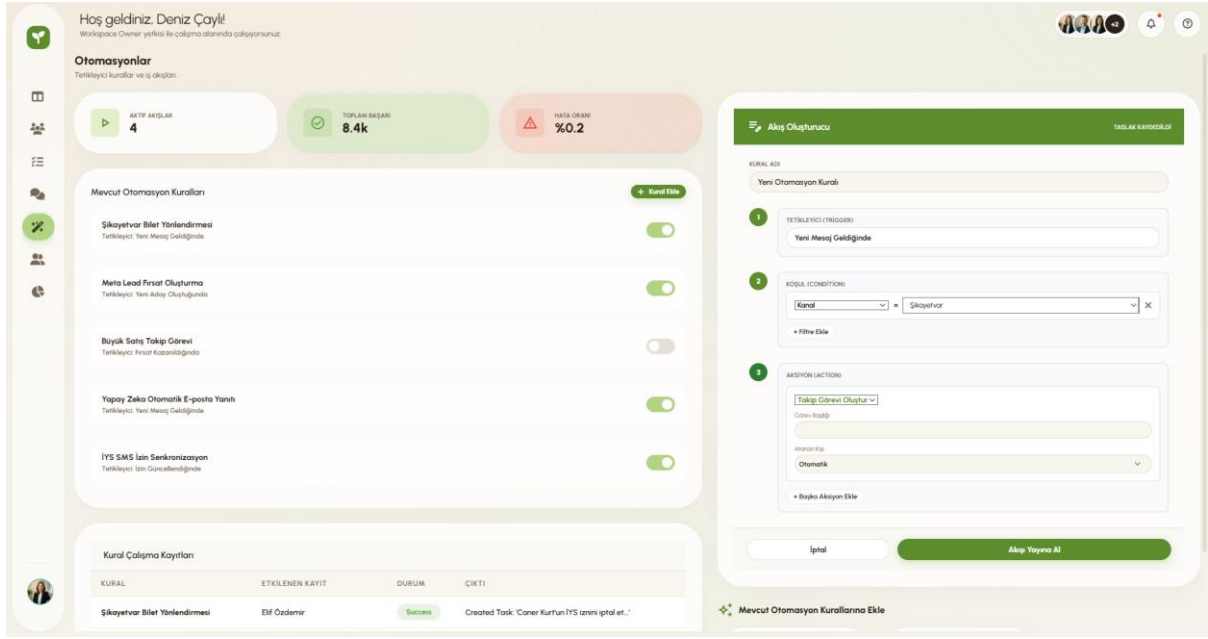
Uygulamanın yapay zekâ özelliği Hugging Face Inference API ile çalışmaktadır. Güvenlik gerekçesiyle Hugging Face erişim tokeni frontend koduna yazılmamış; token Vercel ortam değişkeni olarak tutulmuş ve yapay zekâ istekleri Vercel serverless endpointi üzerinden gerçekleştirilmiştir. Bu yaklaşım, tokenin tarayıcı tarafından görüntülenmesini engellemektedir.

Yapay zekâ; seçilen rapor verisini analiz ederek kısa özet, riskler ve uygulanabilir öneriler üretmektedir. Ayrıca Support ve Admin Inbox ekranlarında, müşteri konuşmasının içeriğine göre mesaj taslağı hazırlayabilmektedir. Üretilen taslak hiçbir koşulda otomatik olarak gönderilmemekte; kullanıcı taslağı önce görmekte, gerekli düzenlemeleri yapmakta ve isterse manuel olarak göndermektedir. Bu tasarım kararı, yapay zekâ çıktısının her zaman insan onayından geçmesini güvence altına almaktadır.

### 3.8. Otomasyonlar

Otomasyon modülü, tekrarlayan CRM işlemlerinin kural tabanlı olarak yürütülmesini sağlamaktadır. Kullanıcı, akış oluşturucu üzerinde tetikleyici (trigger), koşul (condition) ve aksiyon (action) adımlarını tanımlayarak yeni otomasyon kuralları oluşturabilmektedir. Koşul tarafında kanal, konu içeriği, tutar eşiği ve izin durumu gibi kriterler tanımlanabilmekte; aksiyon tarafında ise takip görevi oluşturma, fırsat oluşturma ve SMS gönderme seçenekleri sunulmaktadır.

Mevcut otomasyon kuralları liste hâlinde görüntülenmekte ve her kural anahtar (toggle) ile aktif ya da pasif duruma getirilebilmektedir. Kural çalışma kayıtları (workflow run) hangi kuralın hangi kayıt üzerinde çalıştığını log olarak göstermekte; böylece otomasyonların izlenebilirliği sağlanmaktadır. Şikayetvar akışında aktif bir workflow bulunması hâlinde bu mekanizma doğrudan devreye girmekte ve gelen şikâyet için otomatik takip görevi üretilmektedir. Şekil 3.8.1.'de otomasyon ekranı ve akış oluşturucu görülmektedir.



Şekil 3.8.1. Otomasyonlar Ekranı ve Akış Oluşturucu

### 3.9. Raporlama ve Görselleştirme

Dashboard ve rapor sayfalarında ApexCharts kütüphanesiyle oluşturulan bar, area, donut ve radial bar grafik tipleri kullanılmıştır. Bu grafikler aracılığıyla satış performansı, kanal yoğunluğu, abonelik durumu, destek süreci ve gelir trendleri görselleştirilmektedir. Grafiklerin bir bölümü hover etkileşimiyle ayrıntı gösterecek şekilde yapılandırılmıştır.

Workspace Owner panelindeki dönem filtresi Son 7 Gün, Son 30 Gün ve Tüm Zamanlar seçenekleriyle çalışmakta; satış, açık fırsat, gelir ve kanal yoğunluğu gibi metriklerin görünümünü dinamik olarak değiştirmektedir. Admin raporları ise SaaS şirketleri ve abonelik durumları üzerinden oluşturulmaktadır. Böylece aynı uygulama içerisinde hem operasyonel ekip raporu hem de şirket ve abonelik raporu birbirinden ayrılmış durumdadır.

### 3.10. Ekipler Arası İş Akışı

FlowHub CRM'de ekipler birbirinden bağımsız ekranlara sahip olmakla birlikte ortak CRM verisi üzerinde çalışmaktadır. Sales tarafında bir fırsat satış aşamalarında ilerlerken, Marketing tarafında etkinlik ve PR süreçleri, Support tarafında ise şikâyet ve çözüm süreçleri izlenmektedir. Workspace Owner bu departmanları tek panel üzerinden üstten kontrol etmektedir.

Görevler bu iş akışının önemli bir parçasını oluşturmaktadır. Workspace Owner bir kullanıcıya görev atadığında, görev hem atayan kullanıcının ekranında hem de atanmış kullanıcının görev listesinde görünmektedir. Kullanıcı görevi tamamladığında durum güncellenmekte ve bu değişiklik ortak veri üzerinde yansımaktadır. Müşteri notları ise müşteri ilişkisi geçmişinin CRM içerisinde kalıcı olarak tutulmasını sağlamaktadır.

### 3.11. Güvenlik ve Geliştirme Yaklaşımı

Güvenlik tarafında en kritik uygulama, yapay zekâ entegrasyonunda kullanılan Hugging Face tokenının frontend kaynak koduna eklenmemesidir. Token, Vercel ortam değişkeni olarak saklanmakta ve yapay

zekâ istekleri serverless endpoint aracılığıyla yürütülmektedir. Ayrıca yapay zekâ çıktıları yalnızca taslak olarak sunulmakta; kullanıcı onayı olmadan hiçbir otomatik mesaj gönderimi yapılmamaktadır. Rota seviyesinde ise ProtectedRoute yapısı, yetkisiz rol geçişlerini engellemektedir.

Geliştirme sürecinde ekip üyeleri kendi sorumluluk alanlarına göre branch mantığıyla çalışmıştır. Admin, Sales, Marketing, Support ve Workspace Owner ekranları ayrı geliştirme sorumluluklarıyla ele alınmış, ardından aynı proje yapısı içerisinde birleştirilmiştir. Bu yaklaşım hem görev paylaşımını netleştirmiş hem de kod değişikliklerinin yönetimini ve olası çakışmaların çözümünü kolaylaştırmıştır.

## 4. SONUÇ

Bu çalışmada, rol bazlı ve çok kanallı bir müşteri ilişkileri yönetimi uygulaması olan FlowHub CRM geliştirilmiş; uygulamanın mimarisi, panel yapısı, dış entegrasyonları, yapay zekâ ve otomasyon özellikleri ile veri yönetimi ve güvenlik yaklaşımı incelenmiştir. Proje kapsamında React 18 ve Vite üzerinde bileşen tabanlı bir arayüz kurulmuş; React Router ve ProtectedRoute ile rol bazlı erişim, Redux Toolkit ile merkezi durum yönetimi, ApexCharts ile raporlama ve görselleştirme gerçekleştirilmiştir.

Geliştirme süreci değerlendirildiğinde; öncelikle rol yapısı ve veri modeli tasarlanmış, ardından her departman için ayrı paneller geliştirilmiş, Meta ve Şikayetvar entegrasyonları ile çok kanallı yapı kurulmuş, son olarak yapay zekâ ve otomasyon modülleri sisteme eklenmiştir. Jenerik CRUD servisleri ve LocalStorage tabanlı veri katmanı sayesinde, kullanıcının uygulama üzerinde yaptığı tüm değişiklikler demo oturumu boyunca tutarlı biçimde korunmuştur.

Sonuç olarak FlowHub CRM; demo verileriyle çalışan bir uygulama olmakla birlikte, gerçek bir CRM sisteminin temel iş mantıklarını, veri ilişkilerini ve kullanıcı iş akışlarını başarılı biçimde yansıtmaktadır. Rol bazlı erişim, müşteri ve şirket yönetimi, satış pipeline'ı, destek süreçleri, dış kaynak entegrasyonları, otomasyon, görev yönetimi, raporlama ve yapay zekâ destekli taslak oluşturma gibi modüllerin aynı arayüzde bütünleşik olarak çalışması; projenin hem teknik hem de pratik boyutlarıyla ekip üyelerine kapsamlı bir geliştirme deneyimi kazandırmıştır.

## 5. KAYNAKÇA

1. React Documentation, <https://react.dev>
2. Vite Documentation, <https://vitejs.dev>
3. React Router Documentation, <https://reactrouter.com>
4. Redux Toolkit Documentation, <https://redux-toolkit.js.org>
5. ApexCharts Documentation, <https://apexcharts.com>
6. Hugging Face Inference API Documentation, <https://huggingface.co/docs/api-inference>
7. Vercel Serverless Functions Documentation, <https://vercel.com/docs/functions>