



softITo 4. DÖNEM 1. GRUP
FRONTEND GELİŞTİRME BİTİRME PROJESİ RAPORU

Denvea — Dinamik Form Oluşturma ve Yönetim Platformu

Deniz Çaylı
(Bireysel Proje)

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 19 ve Vite 8	4
2.2. Redux Toolkit ile Durum Yönetimi	4
2.3. Tailwind CSS v4 ile Arayüz Tasarımı.....	4
2.4. React Hook Form ve Zod ile Form Doğrulama.....	4
2.5. Recharts ve @dnd-kit	4
2.6. Axios ve JSON Server ile Veri Katmanı	4
2.7. Proje Mimarisi ve Dosya Yapısı.....	4
2.8. Geliştirme Ortamı ve Çalıştırma	5
3. UYGULAMA	5
3.1. Giriş Ekranı ve Kimlik Doğrulama	5
3.2. Yönetici (Admin) Paneli.....	6
3.2.1. Dashboard (Genel Bakış).....	6
3.2.2. Dinamik Form Oluşturucu (Form Builder).....	6
3.2.3. Katılım ve Yanıt Yönetimi.....	7
3.2.4. Kullanıcı ve Yetki Yönetimi.....	7
3.2.5. Detaylı Raporlama ve Analiz.....	7
3.3. Kullanıcı Portalı	7
3.3.1. Marka Seçimi ve Etkinlik Portalı.....	8
3.3.2. Dinamik Form Arayüzü	8
3.3.3. Akıllı Doğrulama ve Teşekkür Sayfası.....	9
3.4. Güvenlik ve Rol Yetkilendirmesi.....	9
4. SONUÇ	9
5. KAYNAKÇA.....	10

1. GİRİŞ

Kurumların müşteri verisi toplama süreçleri; etkinlik kayıtları, kampanya başvuruları ve anket çalışmaları gibi farklı senaryolarda sürekli olarak yeni form ihtiyaçları doğurmaktadır. Her yeni form için ayrı bir geliştirme süreci yürütmek hem zaman hem de maliyet açısından verimsizdir. Bu ihtiyaca yönelik olarak geliştirilen Denvea, çoklu marka (multi-tenant) yapısını destekleyen, dinamik form oluşturma, veri toplama, kullanıcı yönetimi ve gelişmiş analitik raporlama özelliklerine sahip modern bir CRM ve form yönetim platformudur.

Denvea; kullanıcıların kendi formlarını kod yazmadan tasarlayıp yayınlayabilmelerini, bu formlardan gelen yanıtları detaylı grafiklerle analiz edebilmelerini ve markalara özel portal arayüzleri sunabilmelerini sağlamaktadır. Uygulama React 19 ve Vite 8 üzerinde geliştirilmiş; durum yönetimi Redux Toolkit, arayüz tasarımı Tailwind CSS v4, form doğrulama React Hook Form ve Zod, grafik görselleştirme ise Recharts ile gerçekleştirilmiştir. Bu rapor; projenin amacını, kullanılan materyal ve yöntemleri, yönetici paneli ile kullanıcı portalının işleyişini, güvenlik ve rol yetkilendirme yaklaşımını ekran görüntüleri eşliğinde bütünsel biçimde sunmaktadır.

1.1. Projenin Amacı

Projenin temel amacı, teknik bilgiye sahip olmayan kullanıcıların dahi sürükle-bırak arayüzü ile kendi formlarını tasarlayabildiği, yayınlayabildiği ve gelen yanıtları analiz edebildiği uçtan uca bir form yönetim platformu geliştirmektir. Bu kapsamda dinamik form oluşturma, marka bazlı temalandırma, istemci tarafı şema doğrulama, rol bazlı yetkilendirme, katılım verilerinin filtrelenmesi ve dışa aktarımı ile grafik tabanlı raporlama modüllerinin aynı uygulama içerisinde bütünleşik olarak çalışması hedeflenmiştir.

Projenin bir diğer amacı, tek bir form şemasının farklı markalar için otomatik olarak yeniden temalandırılabilirdiği çoklu marka mimarisinin uygulamalı biçimde gösterilmesidir. Böylece aynı altyapı üzerinden Bioderma, The Purest Solutions, La Roche-Posay ve CeraVe gibi farklı markalara özel görünümde formlar sunulabilmektedir. Proje, bireysel bitirme projesi olarak modern frontend teknolojilerinin gerçek bir ürün senaryosu üzerinde derinlemesine deneyimlenmesini de amaçlamaktadır.

1.2. Proje Kısıtları

Uygulama bu sürümünde gerçek bir üretim veritabanına bağlanmamakta; veri okuma ve yazma işlemleri JSON Server aracılığıyla db.json dosyası üzerinden simüle edilmektedir. Bu nedenle formlar, yanıtlar, kullanıcılar ve marka tanımları yerel geliştirme ortamındaki mock veritabanında tutulmaktadır. Kimlik doğrulama akışı demo hesaplar üzerinden yürütülmekte olup üretim seviyesi bir kimlik sağlayıcısı entegre edilmemiştir. Marka portalı, tanımlı dört harici marka ve Denvea markası ile sınırlıdır. Uygulama yerel geliştirme ortamında çalışacak şekilde yapılandırılmıştır; frontend sunucusu ve mock API sunucusu concurrently kütüphanesi ile aynı anda başlatılmaktadır.

2. MATERYAL VE YÖNTEM

Bu bölümde projenin geliştirilmesinde kullanılan temel teknolojiler, kütüphaneler, proje mimarisi ve geliştirme ortamı açıklanmaktadır. Teknoloji seçimlerinde modern frontend ekosisteminde yaygın kabul görmüş, belgeleri güçlü ve birbiriyle uyumlu çalışan araçlar tercih edilmiştir.

2.1. React 19 ve Vite 8

Uygulamanın arayüzü, bileşen tabanlı mimarisiyle React 19 kütüphanesi kullanılarak geliştirilmiştir. React, form oluşturucu ve dinamik form render katmanı gibi yoğun etkileşimli ekranların küçük ve yeniden kullanılabilir bileşenlere bölünmesine imkân tanımaktadır. Geliştirme ve derleme aracı olarak Vite 8 tercih edilmiştir. Vite, hızlı HMR (hot module replacement) desteği sayesinde geliştirme sürecinde yapılan değişikliklerin anında tarayıcıya yansımaları sağlamakta ve üretim derlemelerini yüksek performansla üretmektedir.

2.2. Redux Toolkit ile Durum Yönetimi

Uygulamanın genel durumu Redux Toolkit ile merkezi olarak yönetilmektedir. Kod tabanında auth, forms, submissions ve users olmak üzere ayrı slice yapıları bulunmakta; sunucu ile veri alışverişi gerektiren işlemler asenkron Thunk aksiyonları üzerinden yürütülmektedir. Bu yapı sayesinde oturum bilgisi, form şemaları, gelen yanıtlar ve kullanıcı listesi uygulamanın her noktasından tutarlı biçimde erişilebilir durumdadır.

2.3. Tailwind CSS v4 ile Arayüz Tasarımı

Arayüz tasarımında Tailwind CSS v4 kullanılmıştır. Modern CSS değişkenleri desteği sayesinde marka bazlı tema renkleri dinamik olarak uygulanabilmekte; yönetici panelinde seçilen ana renk, ikincil renk ve bileşen rengi gibi değerler portal tarafındaki form görünümüne otomatik olarak yansıtılmaktadır. Bu yaklaşım, premium arayüz hedefini tutarlı bir tasarım diliyle desteklemektedir.

2.4. React Hook Form ve Zod ile Form Doğrulama

Form yönetimi React Hook Form, şema doğrulama ise Zod kütüphanesi ile gerçekleştirilmiştir. Bu ikili entegrasyon sayesinde telefon formatı, zorunlu alanlar ve dosya boyutu gibi alan gereksinimleri, form sunucuya gönderilmeden önce istemci tarafında denetlenmektedir. React Hook Form'un kontrolsüz (uncontrolled) yaklaşımı, çok alanlı dinamik formlarda dahi yüksek performans sağlamaktadır.

2.5. Recharts ve @dnd-kit

Raporlama ekranlarındaki grafikler Recharts kütüphanesiyle oluşturulmuş; area, bar ve pie (pasta) grafik tipleri kullanılmıştır. Form tasarımı ekranındaki sürükle-bırak etkileşimi ise erişilebilirlik desteğiyle öne çıkan @dnd-kit kütüphanesi ile sağlanmıştır. İkon setinde modern ve tutarlı çizgi ikonları sunan Lucide React tercih edilmiştir.

2.6. Axios ve JSON Server ile Veri Katmanı

API iletişimi, merkezi bir HTTP istemcisi olarak yapılandırılan Axios üzerinden yürütülmektedir. Veri kaynağı olarak JSON Server kullanılmakta; formlar, yanıtlar, kullanıcılar ve marka tanımları db.json dosyası üzerinden canlı okuma ve yazma simülasyonu ile yönetilmektedir. Bu yaklaşım, gerçek bir REST API davranışının backend geliştirmesine ihtiyaç duyulmadan modellenmesini sağlamaktadır.

2.7. Proje Mimarisi ve Dosya Yapısı

Proje, sorumlulukların net biçimde ayrıştırıldığı katmanlı bir klasör yapısıyla organize edilmiştir. src/app klasörü Redux store ve global durum yönetimini, src/features klasörü auth, forms, submissions ve users slice'larını barındırmaktadır. Yerleşim bileşenleri AdminLayout ve PortalLayout olarak src/layouts altında ayrılmış; Axios instance ve doğrulama şemaları gibi ortak yardımcı araçlar src/lib klasöründe toplanmıştır. Bileşen katmanında admin klasörü form oluşturucu parçaları gibi yönetici paneline özel bileşenleri, portal klasörü kullanıcı portalına özel dinamik render bileşenlerini, shared klasörü ise ProtectedRoute gibi ortak yetkilendirme bileşenlerini içermektedir. Sayfa bileşenleri src/pages altında, yönlendirme şeması ise src/routes altındaki AppRoutes yapısında tanımlanmıştır.

2.8. Geliştirme Ortamı ve Çalıştırma

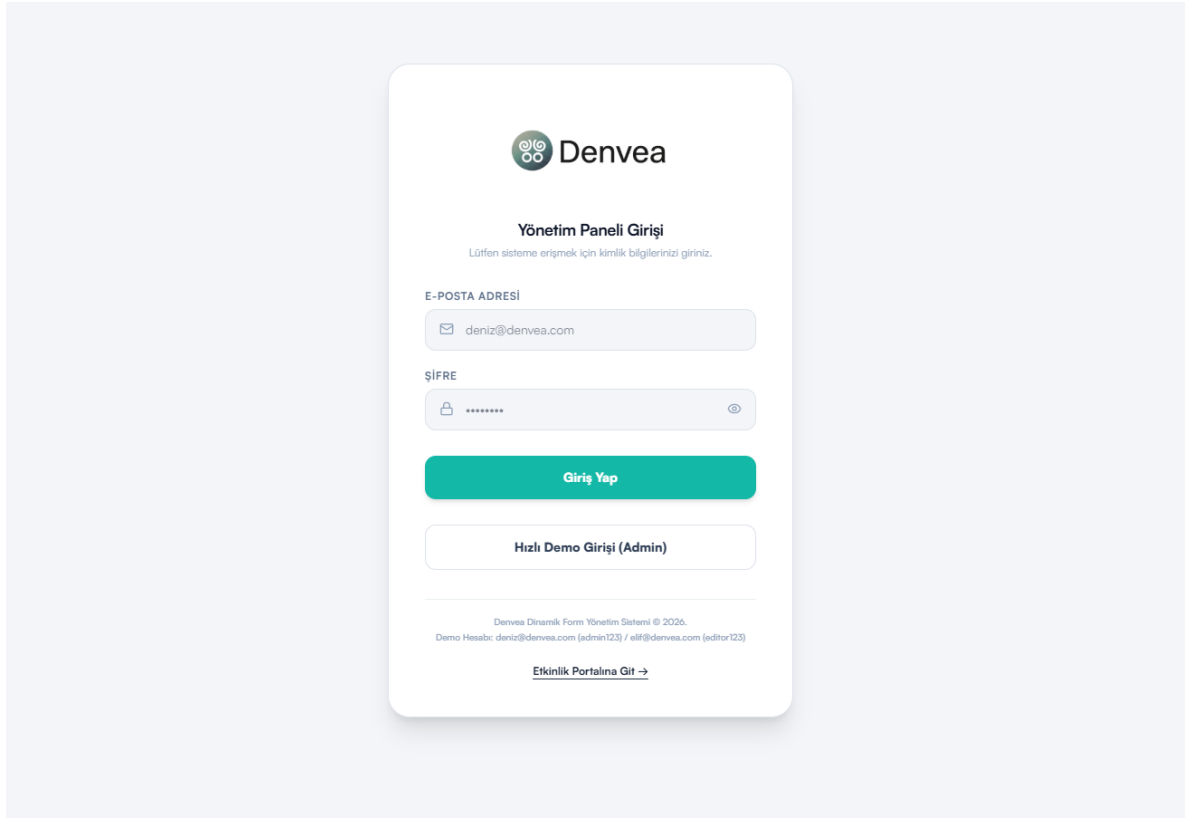
Projenin çalıştırılması için Node.js (v18 veya üzeri) ve npm gereklidir. Bağımlılıklar npm install komutuyla yüklendikten sonra uygulama npm run dev komutuyla başlatılmaktadır. Bu komut, concurrently kütüphanesi aracılığıyla frontend sunucusunu (Vite) ve mock veri sunucusunu (JSON Server) aynı anda ayağa kaldırmaktadır. Frontend portalı <http://localhost:5173>, mock API sunucusu ise <http://localhost:3001> adresi üzerinden hizmet vermektedir.

3. UYGULAMA

Bu bölümde uygulamanın kimlik doğrulama akışı, yönetici paneli modülleri, kullanıcı portalı ve rol yetkilendirme yapısı ekran görüntüleri eşliğinde açıklanmaktadır.

3.1. Giriş Ekranı ve Kimlik Doğrulama

Yönetim paneline erişim, e-posta ve şifre alanlarından oluşan giriş ekranı üzerinden sağlanmaktadır. Ekranda hızlı demo girişi seçeneği bulunmakta; demo hesap bilgileri admin ve editor rolleri için ayrı sunulmaktadır. Giriş ekranından ayrıca Etkinlik Portalına doğrudan geçiş bağlantısı yer almakta; böylece yönetim tarafı ile son kullanıcı tarafı arasındaki ayrım daha ilk ekranda ortaya konmaktadır. Şekil 3.1.1.'de yönetim paneli giriş ekranı gösterilmektedir.



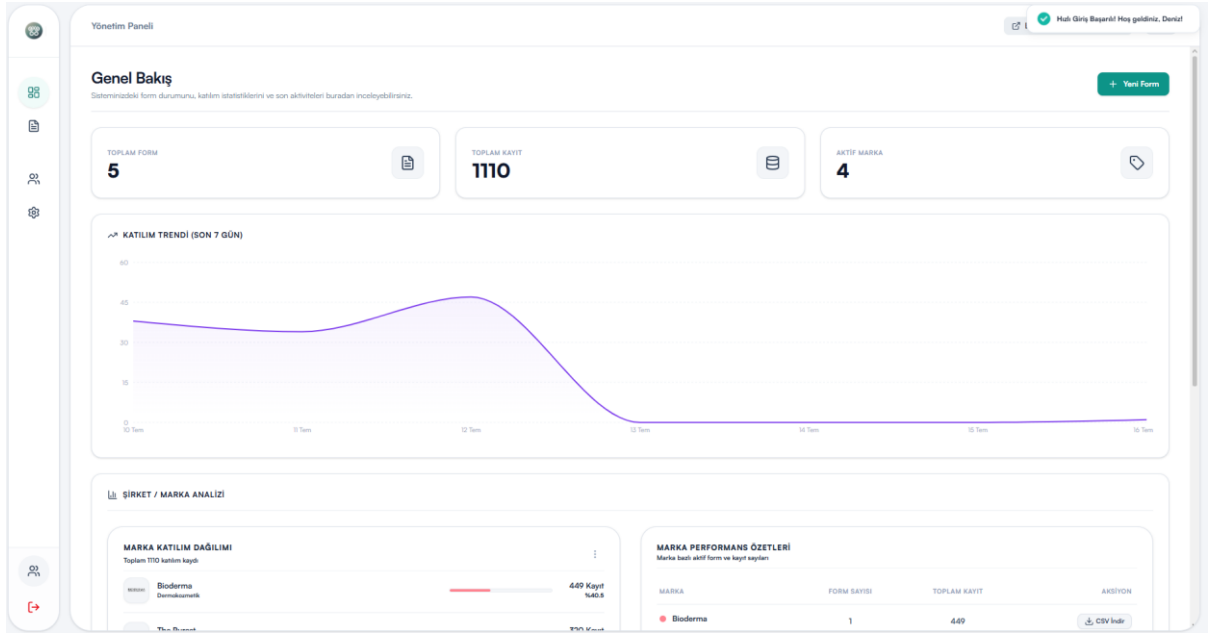
Şekil 3.1.1. Denvea Yönetim Paneli Giriş Ekranı

3.2. Yönetici (Admin) Paneli

Yönetici paneli; genel bakış, form yönetimi, katılım yönetimi, kullanıcı yönetimi ve raporlama modüllerini tek bir arayüzde toplamaktadır. Aşağıda her modül ayrı başlıklar altında açıklanmaktadır.

3.2.1. Dashboard (Genel Bakış)

Dashboard ekranı; toplam form, toplam kayıt ve aktif marka istatistiklerini özet kartlar hâlinde sunmaktadır. Ekranın merkezinde son 7 günün katılım trendini gösteren alan (area) grafiği yer almakta; alt bölümde ise marka bazlı katılım dağılımları ve marka performans özetleri listelenmektedir. Marka performans tablosunda her markanın form sayısı ve toplam kayıt adedi görüntülenmekte, ilgili verinin tek tıkla CSV olarak indirilmesine imkân tanınmaktadır. Şekil 3.2.1.1.'de yönetim panelinin genel bakış ekranı görülmektedir.

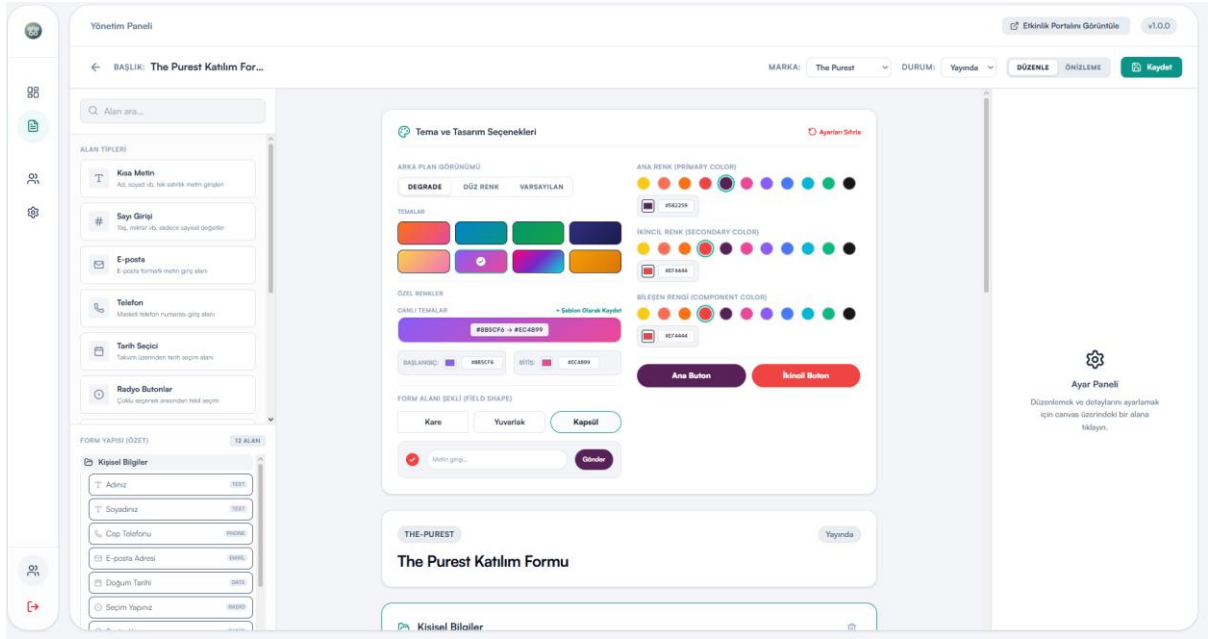


Şekil 3.2.1.1. Yönetim Paneli Genel Bakış (Dashboard) Ekranı

3.2.2. Dinamik Form Oluşturucu (Form Builder)

Form oluşturucu, uygulamanın çekirdek modülüdür. @dnd-kit destekli sürükle-bırak arayüzü ile form başlıkları, bölümleri (section) ve alanları tasarlanabilmektedir. Sol panelde kısa metin, sayı girişi, e-posta, telefon, tarih seçici, radyo butonlar, dosya, tekli/çoklu seçim listeleri ve Türkiye il-ilçe bağlı listeleri gibi gelişmiş alan tipleri sunulmakta; bu alanlar form yapısına eklenerek özelleştirilebilmektedir.

Modülün bir diğer önemli bileşeni tema ve tasarım seçenekleridir. Arka plan görünümü degrade, düz renk ve varsayılan seçenekleriyle belirlenebilmekte; hazır temaların yanı sıra başlangıç ve bitiş renkleri seçilerek özel canlı temalar oluşturulabilmekte ve şablon olarak kaydedilebilmektedir. Ana renk, ikincil renk ve bileşen rengi ayrı ayrı tanımlanabilmekte; form alanı şekli kare, yuvarlak ve kapsül seçenekleri arasından belirlenebilmektedir. Yapılan tüm tema tercihleri, ekranın alt bölümündeki canlı önizleme üzerinde anında izlenebilmektedir. Form, marka ataması ve yayın durumu (Yayında/Taslak) seçimiyle birlikte kaydedilmektedir. Şekil 3.2.2.1.'de form yönetimi ekranı ve tema seçenekleri gösterilmektedir.



Şekil 3.2.2.1. Dinamik Form Oluşturucu ve Tema Tasarım Ekranı

3.2.3. Katılım ve Yanıt Yönetimi

Katılım yönetimi (Submissions) modülü, formlardan gelen yanıtların merkezi olarak izlendiği ekrandır. Gelen kayıtlar marka, form türü ve tarih aralığına göre filtrelenebilmekte; her yanıt, alan bazlı değerlerin görüntülediği detaylı yanıt kartları hâlinde incelenebilmektedir. Yanıt verileri, Türkçe karakter desteğine uygun (UTF-8 BOM) biçimde tek tıkla Excel/CSV olarak dışa aktarılabilir. Bu özellik, toplanan verilerin harici araçlarda sorunsuz biçimde işlenebilmesini sağlamaktadır.

3.2.4. Kullanıcı ve Yetki Yönetimi

Kullanıcı yönetimi modülü üzerinden sisteme yönetici (Admin) veya içerik üretici (Editor) rolünde kullanıcılar eklenebilmektedir. Editör rolündeki kullanıcılar için marka bazlı kısıtlama (allowedBrands) atanabilmekte; bu sayede editörlerin yalnızca kendilerine izin verilen markaların formlarını ve yanıtlarını görmeleri sağlanmaktadır. Bu yapı, çoklu marka mimarisinin veri erişim güvenliği tarafındaki karşılığını oluşturmaktadır.

3.2.5. Detaylı Raporlama ve Analiz

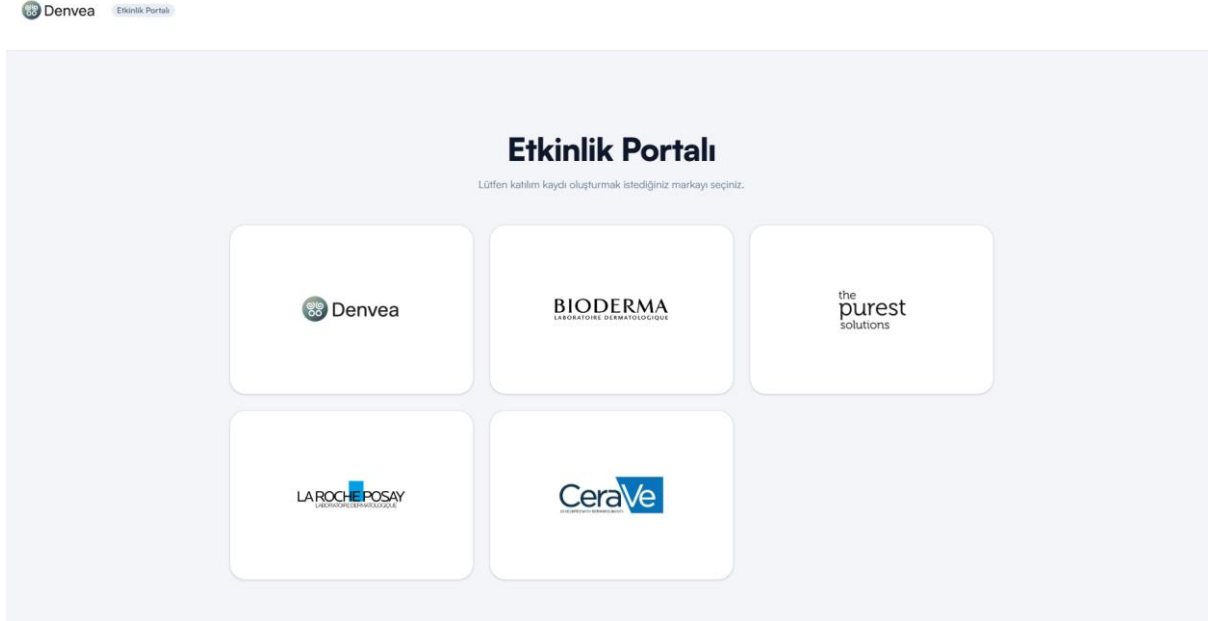
Raporlama modülü, form bazında derinlemesine analiz imkânı sunmaktadır. Seçilen formun son 30 günlük katılım grafiği, katılımcıların il dağılımı ve test/seçim sorularının cevap oranlarını gösteren pasta grafikleri bu ekranda görüntülenmektedir. Ayrıca forma gelen son yanıtların listesi de aynı ekranda sunulmakta; böylece yöneticinin özet metriklerden bireysel kayıtlara kadar tüm katmanları tek noktadan izlemesi mümkün olmaktadır.

3.3. Kullanıcı Portalı

Kullanıcı portalı, yönetici panelinde tasarlanan formların son kullanıcıya sunulduğu katmandır. Portal; marka seçimi, dinamik form arayüzü ve teşekkür sayfası olmak üzere üç temel adımdan oluşmaktadır.

3.3.1. Marka Seçimi ve Etkinlik Portalı

Etkinlik Portalı, katılım kaydı oluşturmak isteyen kullanıcıyı marka seçim ekranıyla karşılamaktadır. Bu ekranda Denvea markasının yanı sıra Bioderma, The Purest Solutions, La Roche-Posay ve CeraVe markaları için optimize edilmiş giriş kartları sunulmaktadır. Kullanıcı, katılmak istediği markanın kartına tıklayarak ilgili markanın yayında olan formuna yönlendirilmektedir. Şekil 3.3.1.1.'de etkinlik portalının marka seçim ekranı görülmektedir.



Şekil 3.3.1.1. Etkinlik Portalı Marka Seçim Ekranı

3.3.2. Dinamik Form Arayüzü

Dinamik form arayüzü, yönetici panelinde tasarlanan form şemalarını gerçek zamanlı olarak render etmektedir. Form; ilgili markanın renkleri ve biçim tercihlerine (örneğin yuvarlatılmış köşeler ve özel marka butonları) göre otomatik tema oluşturularak sunulmaktadır. Böylece aynı form altyapısı, her marka için o markanın kurumsal kimliğine uygun bir görünümle çalışmaktadır. Şekil 3.3.2.1.'de CeraVe markası için render edilen örnek form yapısı gösterilmektedir; formun üst bölümünde marka logosu, altında ise Kişisel Bilgiler bölümü altında ad, soyad, cep telefonu, e-posta ve doğum tarihi alanları yer almaktadır.

Şekil 3.3.2.1. Marka Temalı Dinamik Form Örneği (CeraVe)

3.3.3. Akıllı Doğrulama ve Teşekkür Sayfası

Form gönderimi öncesinde tüm alanlar, React Hook Form ve Zod entegrasyonu ile istemci tarafında denetlenmektedir. Telefon formatı, zorunlu alanların doldurulması ve dosya boyutu/varlığı gibi gereksinimler karşılanmadan form sunucuya gönderilmemekte; kullanıcı hatalı alanlar konusunda anında bilgilendirilmektedir. Form başarıyla tamamlandığında ise kullanıcı, markanın renk şeması ve logosuyla uyumlu, yumuşak SVG animasyonlarıyla zenginleştirilmiş bir teşekkür sayfasına yönlendirilmektedir. Bu sayfa, katılım deneyimini marka kimliğiyle tutarlı biçimde tamamlamaktadır.

3.4. Güvenlik ve Rol Yetkilendirmesi

Sistem genelinde Admin ve Editor olmak üzere iki temel rol bulunmaktadır. Admin rolü; tüm markaların formlarına ve yanıtlarına erişebilmekte, yeni formlar oluşturabilmekte, kullanıcı ekleyip silebilmekte ve sistem ayarlarını düzenleyebilmektedir. Editor rolü ise yalnızca kendisine tanımlanmış markaların formlarını ve yanıtlarını inceleyebilmekte; yeni kullanıcı ekleme veya genel sistem ayarlarını değiştirme yetkisi bulunmamaktadır.

Rota seviyesinde erişim kontrolü, shared bileşenler altında yer alan ProtectedRoute yapısıyla sağlanmaktadır. Oturum açmamış bir kullanıcının yönetim paneli rotalarına doğrudan erişmeye çalışması durumunda yönlendirme engellenmekte ve kullanıcı giriş ekranına döndürülmektedir. Editör kullanıcılar için tanımlanan allowedBrands kısıtı ise veri katmanında filtre olarak uygulanmakta; böylece yetki ayrımı hem arayüz hem de veri erişimi seviyesinde güvence altına alınmaktadır.

4. SONUÇ

Bu çalışmada, çoklu marka yapısını destekleyen dinamik bir form oluşturma ve yönetim platformu olan Denvea geliştirilmiş; uygulamanın mimarisi, yönetici paneli modülleri, kullanıcı portalı, tema sistemi ve rol yetkilendirme yaklaşımı incelenmiştir. Proje kapsamında React 19 ve Vite 8 üzerinde bileşen tabanlı bir arayüz kurulmuş; Redux Toolkit ile merkezi durum yönetimi, React Hook Form ve Zod ile

istemci tarafı şema doğrulama, @dnd-kit ile sürükle-bırak form tasarımı, Recharts ile grafik tabanlı raporlama ve JSON Server ile veri katmanı simülasyonu gerçekleştirilmiştir.

Geliştirme süreci değerlendirildiğinde; öncelikle veri modeli ve rol yapısı tasarlanmış, ardından form şemalarını üreten form oluşturucu modülü geliştirilmiş, bu şemaları marka temalarıyla gerçek zamanlı render eden portal katmanı kurulmuş ve son olarak katılım yönetimi ile raporlama modülleri sisteme eklenmiştir. Tema sisteminin form oluşturucu üzerinden yönetilip portal tarafına otomatik yansması, tek şema ile çok markalı sunum hedefinin başarıyla gerçekleştirildiğini göstermektedir.

Sonuç olarak Denvea; mock veri katmanı ile çalışan bir uygulama olmakla birlikte, gerçek bir form yönetim ve CRM platformunun temel iş mantıklarını, veri ilişkilerini ve kullanıcı akışlarını başarılı biçimde yansıtmaktadır. Dinamik form oluşturma, marka bazlı temalandırma, akıllı doğrulama, rol bazlı yetkilendirme, veri dışı aktarımı ve analitik raporlama gibi modüllerin aynı arayüzde bütünleşik olarak çalışması; projenin bireysel bir bitirme projesi olarak hem teknik hem de pratik boyutlarıyla kapsamlı bir geliştirme deneyimi sunduğunu ortaya koymaktadır.

5. KAYNAKÇA

1. React Documentation, <https://react.dev>
2. Vite Documentation, <https://vite.dev>
3. Redux Toolkit Documentation, <https://redux-toolkit.js.org>
4. Tailwind CSS Documentation, <https://tailwindcss.com>
5. React Hook Form Documentation, <https://react-hook-form.com>
6. Zod Documentation, <https://zod.dev>
7. Recharts Documentation, <https://recharts.org>
8. dnd kit Documentation, <https://dnd-kit.com>
9. Lucide React Documentation, <https://lucide.dev>
10. Axios Documentation, <https://axios-http.com>
11. JSON Server Documentation, <https://github.com/typicode/json-server>