



softlTo 4. Dönem 1. Grup

PATİYUVA

AKILLI HAYVAN BARINAĞI VE SAHİPLENDİRME YÖNETİM SİSTEMİ

BİTİRME PROJESİ RAPORU

Hazırlayan

Merve Daşçi

React • Redux Toolkit • Tailwind CSS • JSON Server • Axios • Vite

Temmuz 2026

İstanbul

İÇİNDEKİLER

1. GİRİŞ	3
1.1 Projenin Amacı	3
1.2 Projenin Kapsamı	4
1.3 Projenin Sağladığı Çözümler	4
2. PROJE HAKKINDA	5
2.1 Pati Yuva Nedir?	5
2.2 Kullanıcı Roller	6
2.3 Çözüm Sunulan Problemler	6
2.4 Genel Özellikler	7
3. KULLANILAN TEKNOLOJİLER	8
4. SİSTEM MİMARİSİ	10
5. UYGULAMA MODÜLLERİ	12
6. KURULUM VE ÇALIŞTIRMA	19
7. TEST VE DOĞRULAMA	21
8. SONUÇ	22
9. KAYNAKÇA	23

1. GİRİŞ

1.1 Projenin Amacı

PatiYuva, hayvan barınaklarında gerçekleştirilen bakım, sağlık takibi, sahiplendirme ve operasyonel süreçlerin tek bir sistem üzerinden yönetilebilmesi amacıyla geliştirilmiş web tabanlı bir yönetim platformudur. Proje kapsamında hayvan kayıtlarının tutulması, sağlık bilgilerinin takip edilmesi, gönüllü ve personel görevlerinin yönetilmesi, stok ve bağış işlemlerinin kontrol edilmesi ve sahiplendirme süreçlerinin dijital ortama taşınması hedeflenmiştir.

Barınak süreçlerinin manuel yöntemlerle yürütülmesi; veri kaybı, takip zorluğu, işlem tekrarı ve zaman kaybı gibi problemlere neden olabilmektedir. PatiYuva, farklı operasyonları tek bir platformda birleştirerek süreçlerin daha düzenli, güvenilir ve ölçülebilir biçimde yürütülmesini amaçlamaktadır.

Projede ayrıca ziyaretçiler için üyelik gerektirmeyen bir sahiplendirme kataloğu ve başvuru sistemi bulunmaktadır. Kullanıcılar sahiplendirilmeyi bekleyen hayvanları inceleyebilmekte, detay bilgilerine erişebilmekte ve çevrim içi başvuru oluşturabilmektedir. Rol bazlı yetkilendirme sayesinde yönetici, personel ve gönüllü kullanıcılar yalnızca kendi sorumluluklarına uygun alanlara erişmektedir.

1.2 Projenin Kapsamı

Proje kapsamında React tabanlı, responsive ve modüler bir web uygulaması geliştirilmiştir. Uygulama; public ziyaretçi tarafı ile giriş gerektiren yönetim panelinden oluşmaktadır. Yönetim panelinde hayvan yönetimi, sağlık ve aşı takibi, sahiplendirme başvuruları, gönüllü ve görev yönetimi, bağış ve stok takibi, kapasite yönetimi, bakım planlama, bildirimler, aktivite geçmişi, profil, ayarlar ve raporlama modülleri yer almaktadır.

Sistem yönetici, personel ve gönüllü olmak üzere üç yetkili kullanıcı rolünü desteklemektedir. Giriş yapmayan ziyaretçiler public ana sayfa ve sahiplendirme kataloğuna erişebilmektedir. Her rolün görebildiği sayfalar ve gerçekleştirebildiği işlemler ProtectedRoute yapısı ile sınırlandırılmıştır.

1.3 Projenin Sağladığı Çözümler

PatiYuva, hayvan barınağı operasyonlarını tek merkezde birleştirerek veri bütünlüğü ve süreç takibi sağlamaktadır. Sağlık kayıtları, aşı tarihleri ve bakım planları hayvan bazında izlenebilmekte; kritik stoklar ve yaklaşan kontroller otomatik uyarılarla görünür hale getirilmektedir.

Akıllı Operasyon Merkezi, gün içinde öncelikli olarak ele alınması gereken veteriner kontrollerini, kritik stokları, bekleyen başvuruları, geciken görevleri ve bakım planlarını tek ekranda

özetlemektedir. Bu yapı, günlük operasyonların planlanmasını kolaylaştırmakta ve kritik konuların gözden kaçmasını azaltmaktadır.

2. PROJE HAKKINDA

2.1 PatiYuva Nedir?

PatiYuva, hayvan barınaklarının idari ve operasyonel ihtiyaçlarına cevap veren web tabanlı bir yönetim sistemidir. Uygulama, barınak içindeki yönetici, personel ve gönüllülerin görevlerini koordine ederken; dışarıdaki ziyaretçilere sahiplendirme kataloğu ve başvuru formu sunmaktadır.

Projenin temel yaklaşımı, barınakta tutulan bilgileri modüllere ayırmak ve tüm modülleri ortak bir veri yapısı üzerinden ilişkilendirmektir. Bir hayvanın temel bilgileri, sağlık geçmişi, bakım planı ve sahiplendirme durumu aynı sistem içinde takip edilebilmektedir.

2.2 Kullanıcı Roller

Rol	Erişim Alanı	Yetki Seviyesi	Temel İşlemler
Ziyaretçi	Public ana sayfa ve katalog	Hayır	Sahiplendirme başvurusu
Gönüllü	Hayvanlar ve kendi görevleri	Sınırlı	Görev durumunu güncelleme
Personel	Operasyon modülleri	Orta	Kayıt ekleme ve güncelleme
Yönetici	Tüm modüller	Tam	Tüm CRUD ve sistem ayarları

Tablo 2.1. Kullanıcı rolleri ve temel yetkiler

Rol bazlı erişim kontrolü, kullanıcının giriş sonrası Redux state ve localStorage içinde tutulan rol bilgisine göre çalışmaktadır. Yetkisiz erişimler Unauthorized sayfasına yönlendirilmekte, kullanıcının yetkili olmadığı menü bağlantıları ise Sidebar içinde gösterilmemektedir.

2.3 Projenin Çözüm Sunduğu Problemler

- Hayvan kayıtlarının dağınık veya manuel tutulması
- Veteriner kontrolleri ve aşı tarihlerinin gecikmesi
- Gönüllü görevlerinin kişilere göre takip edilememesi
- Kritik mama, ilaç ve temizlik stoğunun geç fark edilmesi
- Sahiplendirme başvurularının aşamalarının izlenememesi
- Barınak kapasitesinin bölümlere göre kontrol edilememesi
- Günlük bakım planlarının ve sorumluların merkezi olarak görülememesi

2.4 Projenin Genel Özellikleri

Özellik	Açıklama
Modül yapısı	16+ yönetim ve public modül
Kullanıcı rolleri	Yönetici, personel, gönüllü ve ziyaretçi
Tema	Public açık tema ve yönetim paneli koyu tema
State yönetimi	Redux Toolkit ve createAsyncThunk
API	Axios ve JSON Server REST uçları
Yetkilendirme	Role Based Access Control
Responsive yapı	Masaüstü, tablet ve mobil uyumluluk

Özellik	Açıklama
Bildirim	Kritik stok, sağlık ve görev uyarıları

Tablo 2.2. Projenin genel özellikleri

3. KULLANILAN TEKNOLOJİLER

3.1 Teknoloji Tablosu

Teknoloji	Kullanım Amacı	Katman
React 19	Bileşen tabanlı kullanıcı arayüzü	Frontend
Vite 8	Geliştirme sunucusu ve production build	Araç
Redux Toolkit	Merkezi state ve asenkron işlemler	State Yönetimi
React Router DOM	Public ve protected route yönetimi	Yönlendirme
Tailwind CSS v4	Responsive tasarım ve tema sistemi	UI
Axios	JSON Server ile HTTP iletişimi	Veri İletişimi
JSON Server	REST API simülasyonu ve veri kalıcılığı	Mock Backend
Recharts	Dashboard ve rapor grafiklerinin çizimi	Görselleştirme
Lucide React	Tutarlı ikon sistemi	UI

Tablo 3.1. Projede kullanılan teknolojiler

3.2 React ve Bileşen Mimarisi

Uygulamanın kullanıcı arayüzü React ile geliştirilmiştir. Ortak bileşenler, layout bileşenleri, sayfalar ve özellik bazlı modüller birbirinden ayrılmıştır. PageHeader, StatCard, ProtectedRoute, Sidebar, Topbar ve farklı form bileşenleri yeniden kullanılabilir şekilde tasarlanmıştır.

3.3 Redux Toolkit ve Asenkron Veri Yönetimi

Her iş alanı ilgili feature klasörü içinde bulunan Redux slice dosyası ile yönetilmektedir. Hayvanlar, başvurular, sağlık kayıtları, görevler, bağışlar, stoklar ve bildirimler için createAsyncThunk kullanılarak GET, POST, PATCH ve DELETE işlemleri tanımlanmıştır. Loading ve error durumları her modül içinde ayrı olarak ele alınmıştır.

3.4 JSON Server ve Axios

Gerçek bir backend servisi yerine geliştirme ve değerlendirme amacıyla JSON Server kullanılmıştır. db.json dosyası users, animals, adoptions, healthRecords, volunteers, tasks, donations, inventory, notifications, activityLogs, shelterSections, carePlans ve settings gibi koleksiyonları içermektedir. Axios instance, ortak baseURL üzerinden istekleri yönetmektedir.

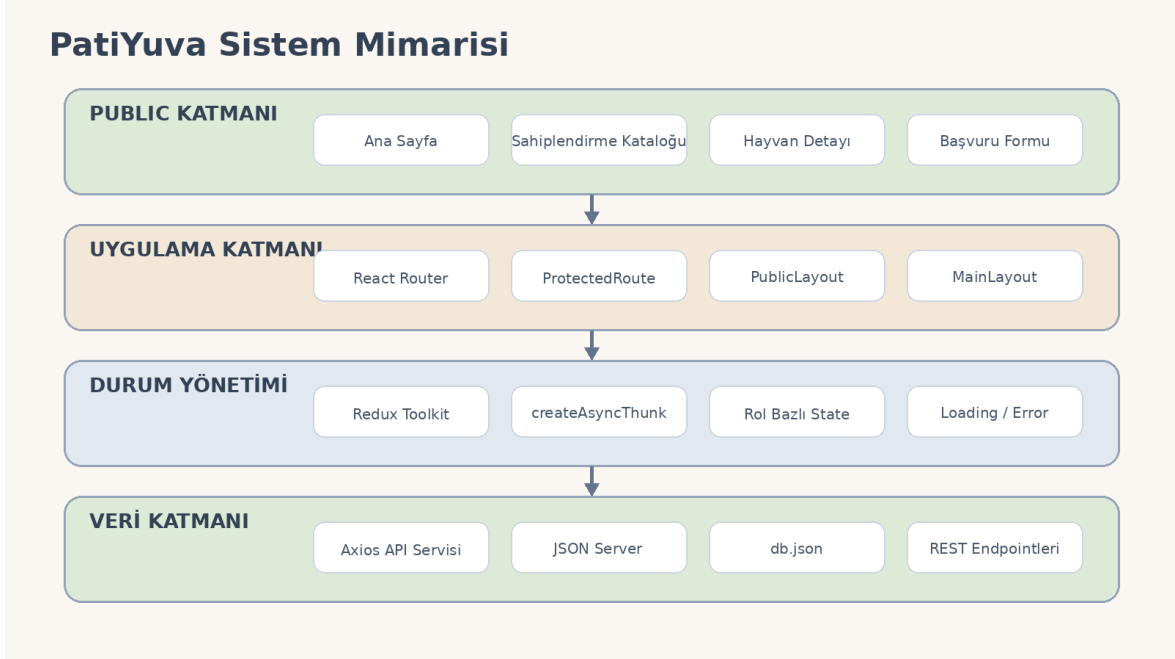
3.5 Tailwind CSS ve Çift Tema

Public sayfalarda krem, açık kahve ve doğal yeşil tonlarından oluşan sıcak bir tema; yönetim panelinde ise koyu slate tonlarından oluşan kurumsal bir tema uygulanmıştır. İki tema ayrı layout yapıları ile yönetildiği için public arayüzde yapılan stil değişiklikleri yönetim panelini etkilememektedir.

4. SİSTEM MİMARİSİ

4.1 Genel Mimari

PatiYuva katmanlı ve feature-based bir frontend mimarisi kullanmaktadır. Kullanıcı arayüzü ve route katmanı, Redux Toolkit durum yönetimi ve Axios veri erişim katmanı üzerinden JSON Server ile haberleşmektedir. Public sayfalar PublicLayout, yönetim sayfaları ise MainLayout içinde çalışmaktadır.



Şekil 4.1. PatiYuva sistem mimarisi

4.2 Proje Klasör Yapısı

Feature-based klasör yapısı sayesinde aynı modüle ait sayfa, form ve slice dosyaları bir arada tutulmaktadır. Bu yaklaşım modüllerin bağımsız geliştirilmesini ve bakımını kolaylaştırmaktadır.

```
src/  
├── app/  
│   └── store.js  
├── components/  
│   ├── common/  
│   └── layout/  
├── features/  
│   ├── animals/  
│   ├── adoptions/  
│   ├── health/  
│   ├── volunteers/  
│   ├── tasks/  
│   ├── donations/  
│   ├── inventory/  
│   └── notifications/
```

```
| | reports/
| | settings/
| pages/
| routes/
| services/
| styles/
```

4.3 Route ve Yetkilendirme Yapısı

Public route'lar /, /adopt, /adopt/:id ve /login adreslerinden oluşmaktadır. Yönetim paneli route'ları ProtectedRoute ile çevrelenmiştir. allowedRoles özelliği kullanıcının rolünü kontrol ederek erişime izin vermekte veya kullanıcıyı /unauthorized sayfasına yönlendirmektedir.

4.4 Veri İlişkileri

Modüller, JSON Server koleksiyonları arasındaki kimlik alanları ile ilişkilendirilmiştir. Sağlık kayıtları, bakım planları ve sahiplendirme başvuruları animalId alanı ile animals koleksiyonuna; görevler assignedVolunteerId ile volunteers koleksiyonuna; stok hareketleri inventoryId ile inventory koleksiyonuna bağlanmaktadır.

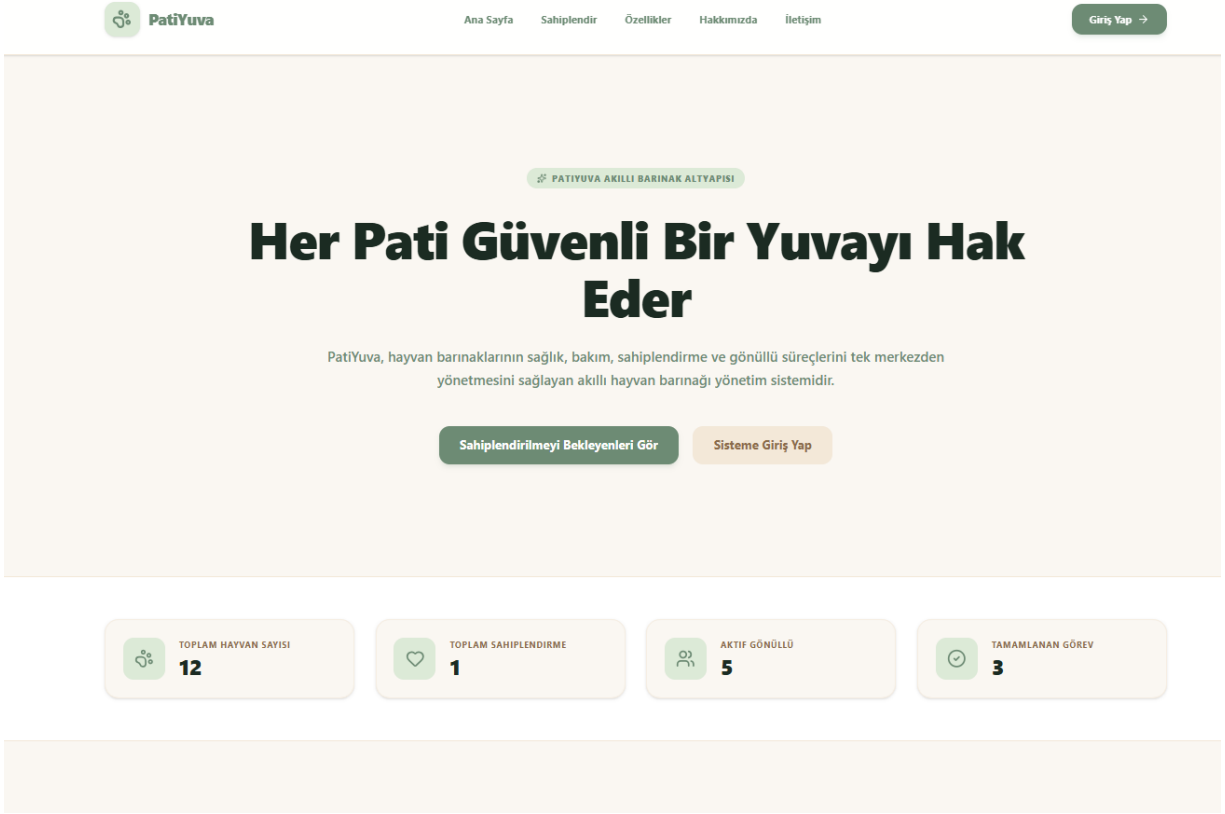
5. UYGULAMA MODÜLLERİ



Şekil 5.1. PatiYuva modül haritası

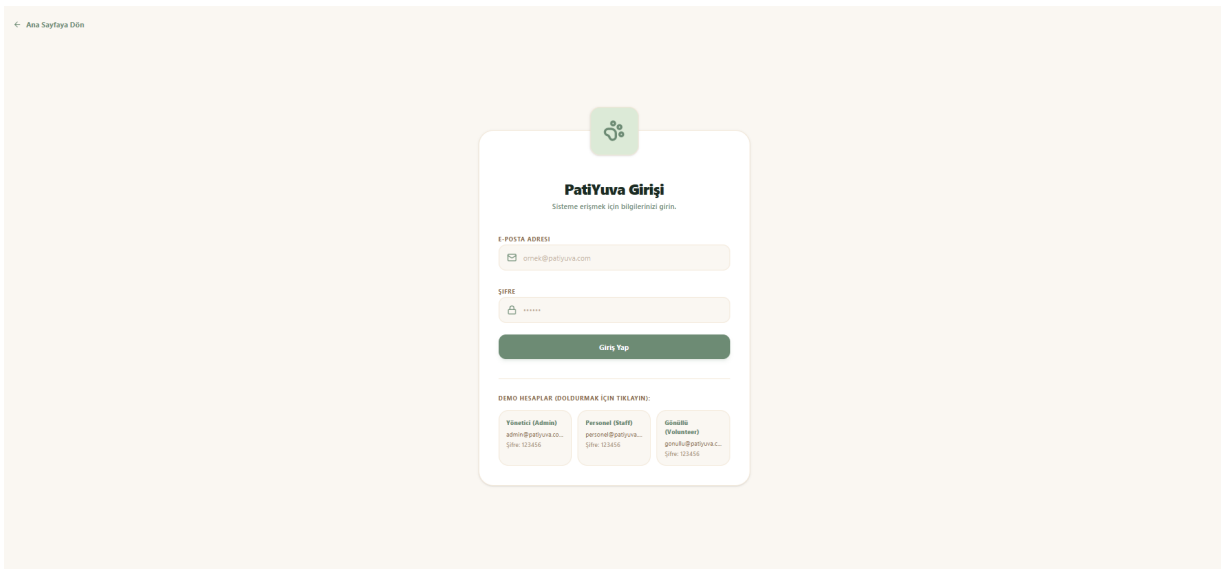
5.1 Public Ana Sayfa ve Sahiplendirme Kataloğu

Public ana sayfa, uygulamayı ilk kez ziyaret eden kullanıcılar için hazırlanmıştır. Açık renkli tema üzerinde hero alanı, barınak istatistikleri, öne çıkan hayvanlar, sistem özellikleri, hakkımızda ve iletişim bölümleri yer almaktadır. Sahiplendirme kataloğu yalnızca sahiplendirilmeyi bekleyen hayvanları göstermekte ve kullanıcıların tür, yaş ve cinsiyete göre filtreleme yapmasına imkân vermektedir.



5.2 Kullanıcı Giriş Sistemi

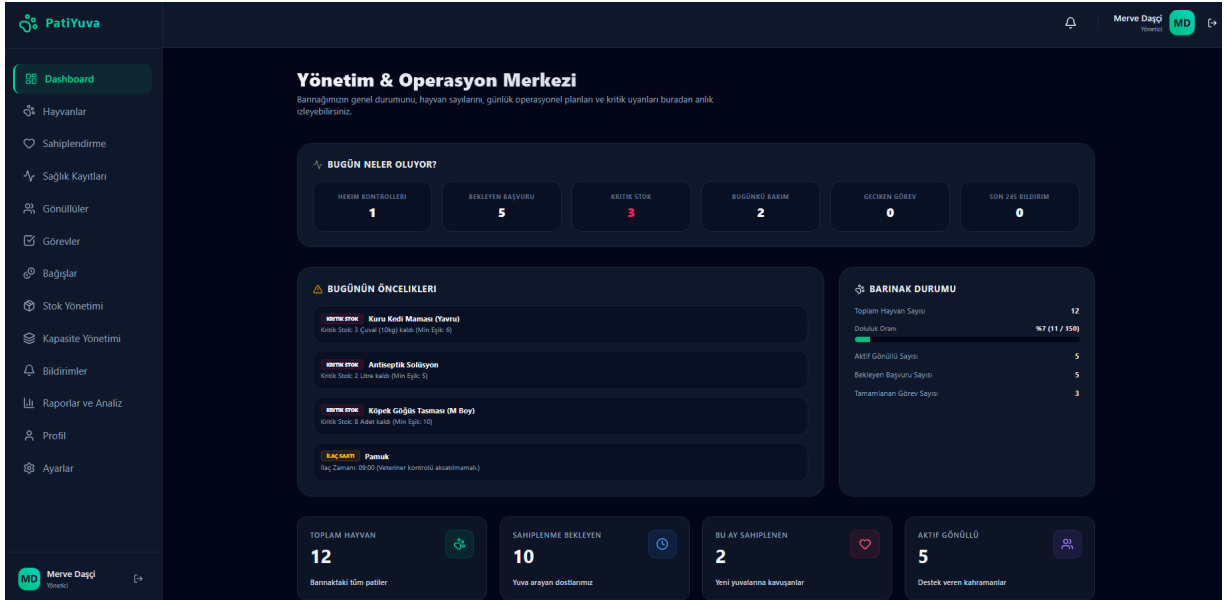
Login sayfası JSON Server users koleksiyonunu kullanır. Demo hesap kartlarına tıklanarak e-posta ve şifre alanları otomatik doldurulabilir. Başarılı girişte kullanıcı bilgileri Redux state ve localStorage içine kaydedilir. Sayfa yenilendiğinde oturum korunur ve kullanıcı rolüne uygun sayfalara erişir.



5.3 Dashboard ve Akıllı Operasyon Merkezi

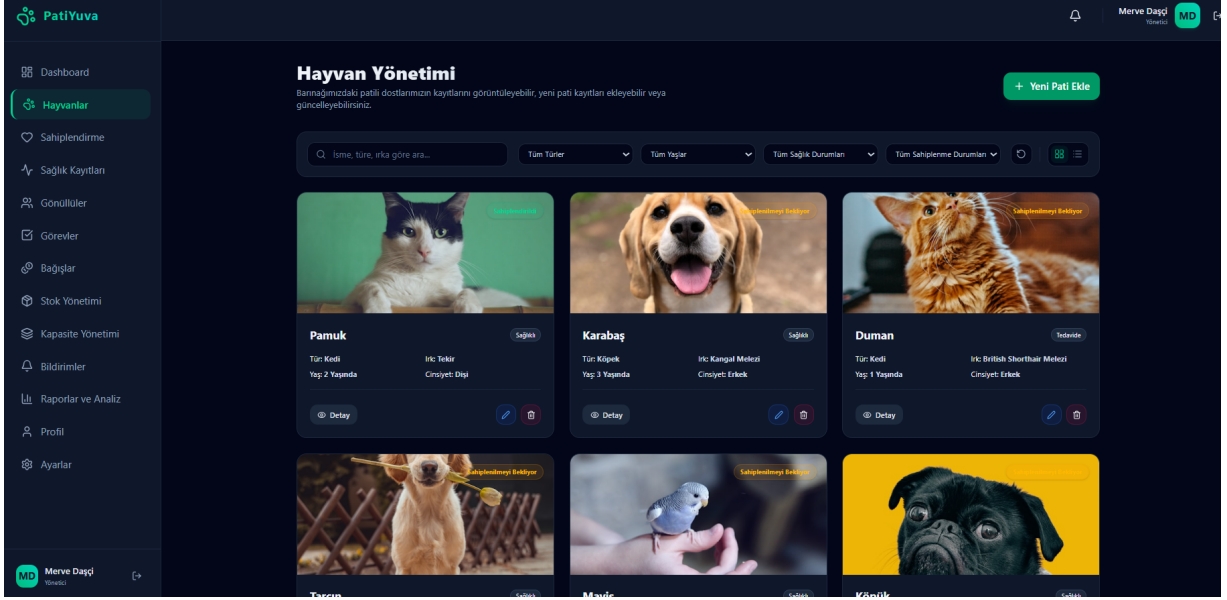
Dashboard, barınağın genel durumunu özetleyen kartlar ve grafikler içerir. Akıllı Operasyon Merkezi günlük veteriner kontrollerini, kritik stokları, bekleyen başvuruları, geciken görevleri ve bakım planlarını öne çıkarır. Veriler farklı koleksiyonlardan dinamik olarak hesaplanır ve hardcoded istatistik kullanılmaz.

- Toplam hayvan ve sahiplendirme istatistikleri
- Aktif gönüllü ve bekleyen görev sayıları
- Kritik stok ve yaklaşan veteriner kontrolleri
- Son aktiviteler ve bildirimler
- Aylık sahiplendirme ve bağış grafikleri



5.4 Hayvan Yönetimi

Animals modülü hayvanların kart ve tablo görünümünde listelenmesini sağlar. İsme, türe, sağlık durumuna ve sahiplendirme durumuna göre arama ve filtreleme yapılabilir. Yönetici tüm CRUD işlemlerini gerçekleştirebilir; personel ekleme ve güncelleme yapabilir; gönüllü ise yalnızca görüntüleyebilir.



5.5 Hayvan Detay Sayfası

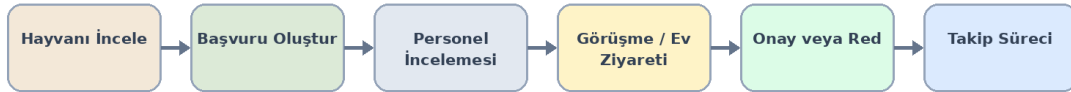
Hayvan detay sayfası genel bilgiler, sağlık geçmişi, aşılar, bakım kayıtları, sahiplendirme süreci ve bakım planı sekmelerinden oluşmaktadır. İlgili veriler animalId kullanılarak farklı koleksiyonlardan getirilir. Böylece her hayvana ait bütün süreçler tek sayfada incelenebilir.

5.6 Sahiplendirme Yönetimi

Sahiplendirme başvuruları yalnızca basit bir CRUD listesi olarak değil, aşamalı bir iş akışı şeklinde yönetilmektedir. Başvurular Bekliyor, İnceleniyor, Görüşme Planlandı, Ev Ziyareti Planlandı, Onaylandı, Reddedildi, Sahiplendirildi ve Takip Süreci gibi durumlara geçirilebilir.

ADAT BİLGİSİ	BASVURU TARİHİ	RANDEVU / GÖRÜŞME	DURUM	İŞLEMLER
Fatma Kara	Bilinmeyen (td. 1)	Planlanmadı	Onaylandı	İncele Reddet
Can Yılmaz	Bilinmeyen (td. 2)	Planlanmadı	Bekleniyor	İncele Reddet
Elif Şen	Bilinmeyen (td. 4)	Planlanmadı	Bekleniyor	İncele Reddet
Murat Can	Bilinmeyen (td. 3)	Planlanmadı	Reddedildi	İncele Reddet
Selin Öztürk	Bilinmeyen (td. 6)	Planlanmadı	Bekleniyor	İncele Reddet
Okan Arıcı	Bilinmeyen (td. 8)	Planlanmadı	Bekleniyor	İncele Reddet

Sahiplendirme Süreci



Şekil 5.2. Sahiplendirme süreci iş akışı

Başvuru onaylanarak sahiplendirme tamamlandığında ilgili hayvanın adoptionStatus alanı otomatik olarak Sahiplendirildi değerine güncellenir. Public formda aynı e-posta ile aynı hayvana tekrar başvuru yapılması engellenmektedir.

5.7 Sağlık ve Aşı Takibi

Sağlık modülü genel muayene, aşı, ameliyat, tedavi, parazit uygulaması ve acil müdahale kayıtlarını yönetir. Yaklaşan kontroller tarih durumuna göre gruplandırılır; geciken kontroller

kırmızı, yaklaşan işlemler sarı uyarı ile gösterilir. Personel kayıt ekleyip güncelleyebilir, silme işlemi yalnızca yöneticiye açıktır.

5.8 Gönüllü ve Görev Yönetimi

Gönüllüler; müsaitlik, yetenek, başlangıç tarihi ve durum bilgileri ile kaydedilir. Görevler kategori, öncelik, son tarih ve atanan gönüllü bilgisi içerir. Gönüllüler yalnızca kendilerine atanan görevleri görür ve durumlarını Devam Ediyor veya Tamamlandı olarak değiştirebilir.

GÖNÜLLÜ BİLGİSİ	MÜSAİTLİK	YETENEKLER	GÖREV SAYISI	KATILIM TARİHİ	DURUM	İŞLEMLER
ayse@patiyuva.org - 0555 123 4567	-	-	0 Görev	-	Aktif	İzleni
ahmet@patiyuva.org - 0555 789 4321	-	-	0 Görev	-	Aktif	İzleni
merve@patiyuva.org - 0555 987 6543	-	-	0 Görev	-	İzleni	İzleni
kaan@patiyuva.org - 0555 456 7890	-	-	0 Görev	-	Aktif	İzleni
elvin@patiyuva.org - 0555 789 0123	-	-	0 Görev	-	Aktif	İzleni
fatma@patiyuva.org - 0555 890 1234	-	-	0 Görev	-	Aktif	İzleni

5.9 Bağış Yönetimi

Bağış modülü nakit ve ayni bağışların kayıt altına alınmasını sağlar. Bağışçı adı, bağış türü, miktar, ürün, ödeme yöntemi ve tarih bilgileri tutulur. Grafikler son altı aylık bağışları, bağış türü dağılımını ve en yüksek bağışçıları göstermektedir.

PatiYuva

Dashboard

Hayvanlar

Sahiplendirme

Saglik Kayıtları

Gonulluler

Gorevler

Bağışlar

Stok Yonetimi

Kapasite Yonetimi

Bildirimler

Raporlar ve Analiz

Profil

Ayarlar

Merve Dağı

Bağış Yönetimi

Barnamızın giderlerini karşılamak ve patili dostlarımızı daha iyi bir yaşam sunmak amacıyla yapılan tüm ayı ve nakdi bağışları izleyin.

+ Yeni Bağış Kaydı Ekle

TOPLAM NAKİT BAĞIŞ
₺9.700

BU AYKI NAKİT BAĞIŞ
₺9.700

TOPLAM AYNI BAĞIŞ
0 İşlem

SON BAĞIŞ TARİHİ
2026-07-16

BAĞIŞ LİSTESİ & İŞLEMLER

ANALİZ & RAPORLAMA

Bağışçı adı, tür veya ürün ara...

Tüm Bağış Türleri

En Yeni İlk

BAĞIŞCI ADI	BAĞIŞ TÜRÜ	DETAY / MİKTAR	TARİH	ÖDEME / YÖNTEM	İŞLEMLER
Mehmet Yılmaz	Nakit	₺500	2026-07-16	-	
All Koçer	Nakit	₺750	2026-07-16	-	
Zeynep Kaya	Nakit	₺1.500	2026-07-15	-	
Hakan Bulut	Nakit	₺2.000	2026-07-14	-	
Meltem Şen	Nakit	₺1.000	2026-07-13	-	
Gizem Aksoy	Nakit	₺350	2026-07-12	-	

5.10 Stok ve Stok Hareketleri

Inventory modülü mama, ilaç, temizlik malzemesi, tıbbi malzeme ve diğer ürünlerin stok seviyelerini takip eder. Miktarı minimum seviyenin altına düşen ürünler kritik olarak işaretlenir. Son kullanma tarihi yaklaşan ürünler sarı, tarihi geçen ürünler kırmızı renkle gösterilir.

Stok giriş, stok çıkış, kullanım, bağıştan gelen ve fire hareketleri ayrı koleksiyonda tutulur. Hareket oluşturulduğunda ürün miktarı otomatik güncellenir ve stok miktarının sıfırın altına düşmesine izin verilmez.

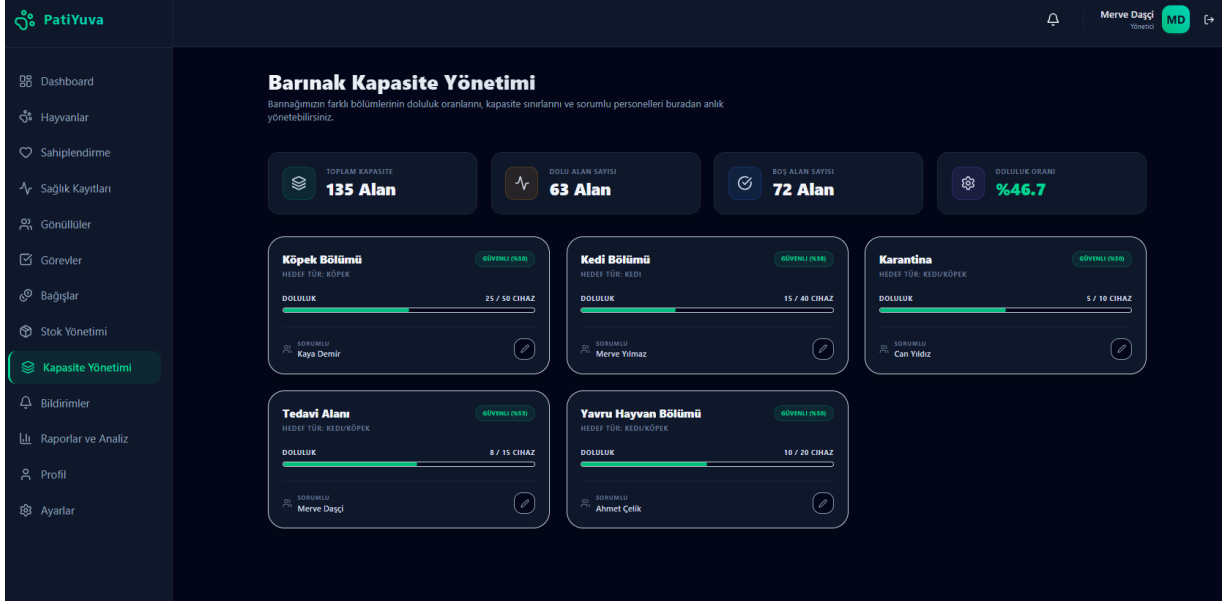
ÜRÜN ADI	KATEGORİ	MELCUT STOK	KRİTİK LİMİT	SON KULLANMA TARİHİ	SON GÜNCELLEME	İŞLEMLER
Kuru Köpek Maması (Yetişkin)	Mama	15 Çuval (15kg)	5 Çuval (15kg)	Belirtilmemiş	-	Gör Düzenle Sil
Kuru Kedi Maması (Yetişkin)	Mama	3 Çuval (10kg)	6 Çuval (10kg)	Belirtilmemiş	-	Gör Düzenle Sil
Kedi Konserve Yaş Mama	Mama	120 Adet	50 Adet	Belirtilmemiş	-	Gör Düzenle Sil
Köpek Konserve Yaş Mama	Mama	80 Adet	40 Adet	Belirtilmemiş	-	Gör Düzenle Sil
Antiseptik Solüsyon	Tıbbi	2 Litre	5 Litre	Belirtilmemiş	-	Gör Düzenle Sil
Steril Sarğı Bezi	Tıbbi	50 Adet	20 Adet	Belirtilmemiş	-	Gör Düzenle Sil
Klorak Dezenfektan	Temizlik	10 Bidon (5L)	4 Bidon (5L)	Belirtilmemiş	-	Gör Düzenle Sil
Mikrofiber Temizlik Bezi	Temizlik	25 Adet	10 Adet	Belirtilmemiş	-	Gör Düzenle Sil

5.11 Bildirim Sistemi

Bildirim modülü kritik stok, yaklaşan veteriner kontrolü, geciken aşı, yeni sahiplendirme başvurusu ve görev ataması gibi olayları kullanıcıya iletir. Topbar içindeki bildirim panelinde okunmamış bildirim sayısı, son bildirimler ve tümünü okundu olarak işaretleme seçenekleri bulunur.

5.12 Barınak Kapasite Yönetimi

Barınak bölümleri kapasite, dolu alan ve sorumlu personel bilgileri ile tanımlanır. Köpek bölümü, kedi bölümü, karantina, tedavi alanı ve yavru hayvan bölümü ayrı ayrı izlenebilir. Doluluk oranı yüzde 80 üzerinde sarı, yüzde 95 üzerinde kırmızı uyarı verir.



5.13 Bakım Planlama Sistemi

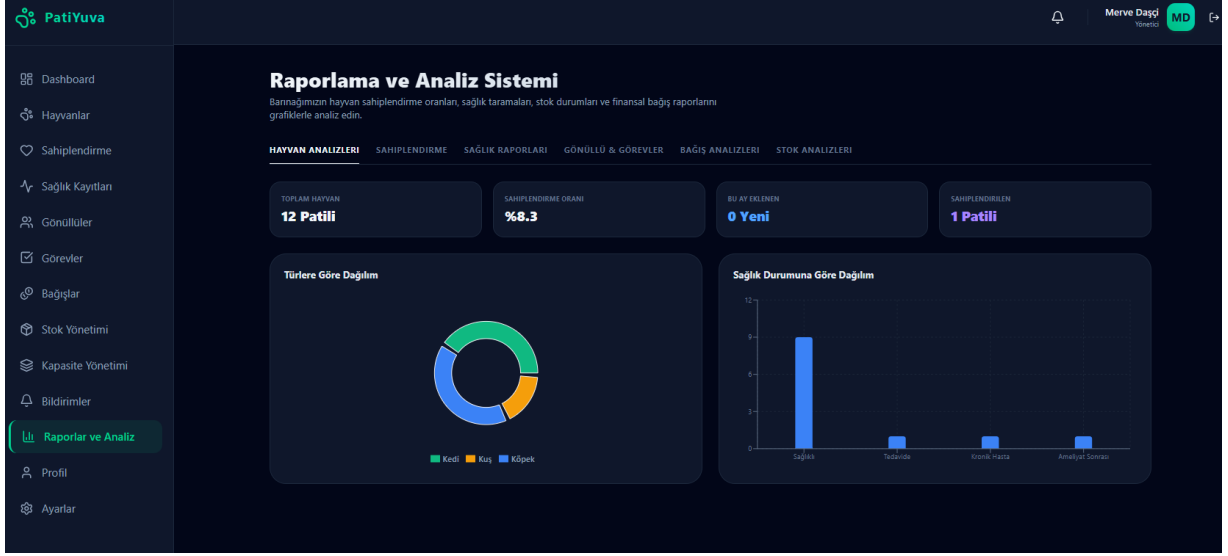
Bakım planlarında mama saati, mama türü, su kontrolü, egzersiz zamanı, ilaç saati, aşı tarihi, veteriner kontrolü ve özel bakım notları yer alır. Dashboard üzerinde Bugünkü Bakım Planları bölümü, o gün tamamlanması gereken işlemleri otomatik olarak listeler.

5.14 Aktivite Geçmişi

Hayvan ekleme veya güncelleme, başvuru durum değişikliği, sağlık kaydı ekleme, görev tamamlama, stok hareketi ve profil güncelleme gibi önemli işlemler activityLogs koleksiyonuna kaydedilir. Her kayıt işlemi yapan kullanıcıyı, modülü, açıklamayı ve zamanı içerir.

5.15 Raporlama ve Analiz

Reports modülü hayvan, sahiplendirme, sağlık, gönüllü, bağış ve stok verilerini analiz eder. Tür dağılımı, sahiplendirme oranı, aylık bağışlar, kritik stok sayısı, yaklaşan kontroller ve en aktif gönüllüler gibi ölçümler Recharts bileşenleriyle görselleştirilir.



5.16 Profil ve Sistem Ayarları

Kullanıcılar kendi profil bilgilerini ve şifrelerini güncelleyebilir. Güncelleme sonrası Redux state ve localStorage verileri eş zamanlı yenilenir. Sistem ayarları yalnızca yönetici tarafından değiştirilebilir ve barınak iletişim bilgileri, bildirim tercihleri, tema ve sayfalama seçeneklerini içerir.

Profil Ayarlarım
Hesap bilgilerinizi, profil fotoğrafınızı ve giriş şifrenizi buradan yönetebilirsiniz.

Kişisel Bilgiler
Adınız, telefon numaranız ve profil fotoğrafınızı güncelleyin.

AD SOYAD
Merve Daşçı

TELEFON
0555 111 2233

PROFİL FOTOĞRAFI URL
https://example.com/avatar.jpg

Bilgileri Kaydet

HESAP GEÇMİŞİ

- Başlangıç Tarihi: 10.01.2026
- Son Giriş Tarihi: Bugün (2026-07-16)
- Yeni Deneyimi: Admin

Şifre Değiştir
Güvenliğiniz için şifrenizi belirli aralıklarla değiştirin.

MEVCUT ŞİFRE

YENİ ŞİFRE
En az 6 karakter

YENİ ŞİFRE TEKRAR
Tekrar girin

Şifreyi Güncelle

6. KURULUM VE ÇALIŞTIRMA

6.1 Gereksinimler

- Node.js ve npm
- Git
- Modern bir web tarayıcısı
- Kod editörü (Visual Studio Code önerilir)

6.2 Projenin Kurulması

Proje GitHub üzerinden indirildikten veya klonlandıktan sonra proje kök dizininde terminal açılır ve bağımlılıklar yüklenir.

```
git clone https://github.com/merve-dasci/patiyuva.git  
  
cd patiyuva  
  
npm install
```

6.3 JSON Server'ın Çalıştırılması

Uygulamanın verileri db.json dosyasında bulunmaktadır. REST API simülasyonunu başlatmak için aşağıdaki komut çalıştırılır. Sunucu 3001 portunda açılır.

```
npm run server
```

6.4 Frontend Uygulamasının Çalıştırılması

Yeni bir terminal penceresinde Vite geliştirme sunucusu başlatılır. Uygulama varsayılan olarak <http://localhost:5173> adresinde görüntülenir.

```
npm run dev
```

6.5 Production Build

Yayın öncesinde derleme hatalarını kontrol etmek ve optimize edilmiş production çıktısı oluşturmak için `npm run build` komutu çalıştırılır. Derleme sonucunda dist klasörü oluşur.

```
npm run build
```

6.6 Demo Kullanıcılar

Rol	E-posta	Şifre
Yönetici	admin@patiyuva.com	123456
Personel	personel@patiyuva.com	123456
Gönüllü	gonullu@patiyuva.com	123456

Tablo 6.1. Demo kullanıcı giriş bilgileri

7. TEST VE DOĞRULAMA

7.1 Fonksiyonel Testler

Test Alanı	Kontrol	Sonuç
Giriş ve oturum	Üç demo rolüyle giriş, yenilemede oturumun korunması	Başarılı
Rol yetkileri	Yetkisiz route erişiminin engellenmesi	Başarılı
Hayvan CRUD	Ekleme, güncelleme, silme ve detay görüntüleme	Başarılı
Arama ve filtre	Modül bazlı arama alanları ve filtrelerin birlikte çalışması	Başarılı
Sağlık kayıtları	Kayıt ekleme, güncelleme ve hayvan detayı entegrasyonu	Başarılı
Stok hareketi	Giriş/çıkış sonrası miktarın güncellenmesi	Başarılı
Sahiplendirme	Başvuru oluşturma ve durum güncelleme	Başarılı
Bildirimler	Okundu ve tümünü okundu işlemleri	Başarılı

Tablo 7.1. Fonksiyonel test sonuçları

7.2 Responsive Tasarım Testleri

Public sayfalar, dashboard, tablolar, kartlar, modal pencereler ve formlar masaüstü ve mobil ekran boyutlarında kontrol edilmiştir. Mobil görünümde sidebar açılır menü olarak çalışmakta, kartlar tek sütuna düşmekte ve sayfalarda yatay taşma oluşmaması hedeflenmektedir.

7.3 Hata ve Boş Veri Durumları

API bağlantısı kurulamadığında kullanıcıya anlaşılır hata mesajı gösterilmektedir. Veri bulunmayan listelerde EmptyState bileşeni kullanılmakta, asenkron istekler sırasında loading göstergeleri görüntülenmektedir. Silme işlemlerinde window.confirm yerine özel onay modalı kullanılmaktadır.

7.4 Veri Kalıcılığı

Ekleme, güncelleme ve silme işlemleri JSON Server üzerinden db.json dosyasına kaydedilmektedir. Kullanıcı oturumu localStorage ile korunmakta, profil değişiklikleri Redux state ile localStorage üzerinde eş zamanlı güncellenmektedir.

8. SONUÇ

PatiYuva projesi ile hayvan barınaklarının hayvan, sağlık, sahiplendirme, gönüllü, görev, bağış, stok, kapasite ve bakım süreçlerini tek bir platformdan yönetebileceği kapsamlı bir web uygulaması geliştirilmiştir. Uygulama yalnızca yönetim panelinden oluşmamakta, ziyaretçilere açık sahiplendirme kataloğu ve başvuru sistemi de sunmaktadır.

Redux Toolkit ile merkezi durum yönetimi, JSON Server ve Axios ile REST tabanlı veri iletişimi, React Router DOM ile rol bazlı yönlendirme ve Tailwind CSS ile responsive çift tema yapısı uygulanmıştır. Modüllerin feature-based klasör yapısı altında geliştirilmesi, projenin okunabilirliğini ve genişletilebilirliğini artırmıştır.

Akıllı Operasyon Merkezi, kritik stoklar, yaklaşan veteriner kontrolleri, geciken görevler ve günlük bakım planları gibi önemli verileri tek ekranda bir araya getirerek barınak operasyonlarının daha hızlı değerlendirilmesini sağlamaktadır. Aktivite geçmişi ve bildirim sistemi ise yapılan işlemlerin izlenebilirliğini desteklemektedir.

Gelecek geliştirmelerde gerçek bir backend servisi, kimlik doğrulama altyapısı, dosya ve fotoğraf yükleme, e-posta bildirimleri, QR kod destekli bakım işlemleri, çoklu barınak desteği ve çevrim içi bağış sistemi eklenebilir. Bu geliştirmelerle PatiYuva'nın gerçek kullanım ortamına uygun daha kapsamlı bir ürüne dönüştürülmesi mümkündür.

9. KAYNAKÇA

1. React Documentation, <https://react.dev/>
2. Redux Toolkit Documentation, <https://redux-toolkit.js.org/>
3. React Router Documentation, <https://reactrouter.com/>
4. Tailwind CSS Documentation, <https://tailwindcss.com/docs>
5. Vite Documentation, <https://vite.dev/guide/>
6. Axios Documentation, <https://axios-http.com/docs/intro>
7. JSON Server Repository, <https://github.com/typicode/json-server>
8. Recharts Documentation, <https://recharts.org/>
9. Lucide React Documentation, <https://lucide.dev/guide/packages/lucide-react>