

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4538 - IOS İLE MOBİL UYGULAMA GELİŞTİRME II

PROJE RAPORU

PETOPIA - Hayvan Barınağı Mobil Uygulama Projesi

Sedef Kjamili

21290987

github.com/sedefkjamili

Haziran 2025

GITHUB VE VIDEO BAĞLANTISI

Github Linki:

<https://github.com/sedefkjamili/PetopiaApp.git>

Video Linki:

<https://drive.google.com/file/d/10JmpK84SoS01ouBpKJGTGVjk2NjN28zj/view?usp=sharing>

Veritabanı Video Linki:

https://drive.google.com/file/d/1azlz_nH6PN2oAZi3jLNbs9yIFsWgQQf3/view?usp=sharing

Apk Dosyasının Drive Linki: <https://drive.google.com/file/d/1gVoNou-f-m-rFoyrXQvu6zPH7kjG2Yhz/view?usp=sharing>

Apk Expo Dosyasının Linki:

<https://expo.dev/accounts/sedefkjamili/projects/PetopiaRN/builds/8f12662a-b592-49dd-80a9-a4080554b09f>

Tüm Dosyaların Drive Linki:

https://drive.google.com/drive/folders/1nw_bHYdDb_kDQwIEEz3X91AVQkiM7FOv?usp=sharing

ÖZET

Petopia adlı bu proje, sahihsiz hayvanların dijital platformlarda görünürlüğünü artırmayı ve barınaklardaki sahiplendirme süreçlerini daha verimli hale getirmeyi amaçlamaktadır. Geleneksel yöntemlerin sınırlılıklarını ortadan kaldırarak hem barınak çalışanlarının iş yükünü azaltmak hem de hayvan sever bireylerin hayvanlara kolayca erişmesini sağlamak hedeflenmiştir.

Uygulama, kullanıcı dostu arayüzü sayesinde çeşitli özellikler sunmaktadır. Kullanıcılar hayvan profillerini görüntüleyebilir, yeni ilanlar ekleyebilir, kayıp hayvan bildiriminde bulunabilir ve sistemden bildirim alabilirler. Ayrıca kategoriye göre filtreleme, arama çubuğu ve kullanıcı profili düzenleme gibi etkileşimli modüllerle kullanım kolaylığı sağlanmıştır.

Uygulamanın ön yüzü React Native teknolojisi ile geliştirilmiş olup hem Android hem de iOS cihazlarda çalışacak şekilde optimize edilmiştir. Bu yapı, tek bir kod tabanı üzerinden platformlar arası uyumluluk sağlarken, mobil cihazlarda yüksek performans ve tutarlı kullanıcı deneyimi sunar.

Sunucu tarafında Node.js ve Express.js teknolojileri kullanılmıştır. Veriler MongoDB üzerinde saklanmakta, esnek ve ölçeklenebilir bir veritabanı yapısı sunulmaktadır. Kullanıcı, hayvan ve bildirim verileri Mongoose modelleriyle tanımlanarak sistematik şekilde işlenmektedir.

Kullanıcı güvenliği, JWT (JSON Web Token) tabanlı kimlik doğrulama sistemi ile sağlanmaktadır. Şifreleme işlemleri bcrypt kütüphanesi kullanılarak yapılmakta ve kullanıcı bilgileri güvenli şekilde korunmaktadır. Bu yapı sayesinde hem veri bütünlüğü hem de kullanıcı gizliliği güvence altına alınmaktadır.

İÇİNDEKİLER

GITHUB VE VİDEO BAĞLANTISI.....	i
ÖZET	ii
İÇİNDEKİLER.....	iii
1. GİRİŞ	1
2. TEKNOLOJİK ALTYAPI.....	2
2.1. Frontend – React Native	2
2.2. Backend – Node.js & Express.js	2
2.3. Veritabanı – MongoDB.....	3
2.4. API İletişimi	4
3. VERİ TABANI TASARIMI VE TABLOLAR.....	8
4. UYGULAMANIN GELİŞTİRİLMESİ.....	10
4.1. Karşılama Ekranı – welcomescreen.tsx.....	10
4.2. Ana Sayfa – home.tsx.....	11
4.3. Kayıt Sayfası – signup.tsx	11
4.4. Giriş Sayfası – signin.tsx	12
4.5. Profil Sayfası – profile.tsx.....	13
4.6. Hayvan Ekleme Sayfası – addpet.tsx.....	14
4.7. Bildirimler Sayfası – notifications.tsx	15
4.8. Hastalıklar Sayfası – diseases.tsx	16
4.9. Hayvan Detay Sayfası – [id].tsx	17
4.10. İletişim Sayfası – contact.tsx	18
4.11. Hakkımızda Sayfası – about.tsx.....	19
SONUÇ	21
KAYNAKLAR	22

1. GİRİŞ

Hayvan barınakları, sahipsiz hayvanlara bakım sağlayan ve sahiplendirme sürecine aracılık eden önemli kurumlardır. Ancak işlemlerin çoğu hâlâ manuel yürütüldüğü için bilgi akışı yavaşlamakta ve verimlilik düşmektedir.

Petopia adlı mobil uygulama, kullanıcıların barındaki hayvanlara kolaylıkla ulaşmasını, sahiplenme sürecini başlatmasını ve barınak personelinin süreçleri daha etkin yönetmesini sağlamak amacıyla tasarlanmıştır. Uygulama; kullanıcıların hayvan bilgilerine tür, yaş, cinsiyet gibi kriterlere göre ulaşmasını mümkün kılarken; aynı zamanda kullanıcı dostu bir arayüz ve güvenli bir alt yapı sunarak sahiplenme sürecini daha erişilebilir hale getirmektedir.

Uygulamanın mobil platformda geliştirilmiş olması, hem kullanıcıların diledikleri zaman bilgiye erişimini kolaylaştırmakta hem de barınak çalışanlarının hayvan bilgilerini yerinde güncelleyebilmesine olanak tanımaktadır. Sahiplendirme süreci, baştan sona dijital ortamda yürütülebilmekte; başvurular alınmakta, sistem üzerinden takip edilmekte ve gerekli bildirimler kullanıcılara iletilmektedir.

Bu proje iki temel hedefe odaklanmaktadır:

- **Kullanıcı Deneyimi:** Hayvan sahiplenmek isteyen bireylerin hızlı, kolay ve etkili bir şekilde ihtiyaç duydukları hayvana ulaşmasını sağlayan, sade ve yönlendirici bir kullanıcı arayüzü sunulmuştur. Kullanıcılar diledikleri hayvanın bilgilerini detaylı olarak inceleyebilmekte ve sahiplendirme talebinde bulunabilmektedir.
- **Barınak Yönetimi:** Barınak personelinin iş yükünü azaltmak, hayvan kayıtlarını düzenli ve güncel tutmak ve başvuru sürecini dijital olarak takip edebilmek amacıyla mobil cihazlardan erişilebilen yönetim paneli geliştirilmiştir. Bu sayede süreçler daha verimli, takip edilebilir ve şeffaf bir yapıya kavuşturulmuştur.

Petopia ile amaçlanan; barınaklardaki hayvanların daha hızlı sahiplendirilmesi, kayıp hayvanların daha kolay bulunması ve barınak yönetiminin dijitalleşme sayesinde daha etkili hale getirilmesidir.

2. TEKNOLOJİK ALTYAPI

Petopia uygulaması, modern mobil geliştirme teknolojileri kullanılarak inşa edilmiştir. Geliştirme sürecinde hem kullanıcı deneyimini ön planda tutan hem de güvenilir ve ölçeklenebilir bir sistem kurulmasına imkân tanıyan yazılım araçları tercih edilmiştir.

2.1. Frontend – React Native

Uygulamanın kullanıcı arayüzü, çok platformlu mobil geliştirmeye olanak tanıyan **React Native** teknolojisi ile hazırlanmıştır. Bu sayede tek bir kod tabanı üzerinden hem Android hem iOS cihazlarda çalışan bir uygulama elde edilmiştir. Arayüzde Expo Router kullanılarak sayfalar arası geçişler sağlanmış, kullanıcı deneyimini iyileştirmek için animasyonlar ve özelleştirilmiş stiller uygulanmıştır.

React Native'in avantajları:

- Tek kodla iki platform desteği (Android/iOS)
- Geniş topluluk ve bileşen desteği
- Hızlı geliştirme ve test döngüsü
- Kullanıcı dostu ve modern arayüz tasarımları



Şekil 2.1. React Native Uygulama Logosu

2.2. Backend – Node.js & Express.js

Uygulamanın sunucu tarafı, **Node.js** üzerinde çalışan ve RESTful API'ler sağlayan **Express.js** framework'ü kullanılarak geliştirilmiştir. Kullanıcı kayıt ve giriş işlemleri, hayvan ve bildirim verilerinin işlenmesi gibi tüm sunucu işlemleri bu yapı üzerinden yürütülmektedir.

Backend tarafı şu işlemleri kapsar:

- Kullanıcı oturum yönetimi (JWT ile token doğrulama)
- Yeni hayvan kaydı ekleme
- Bildirim oluşturma ve listeleme
- İletişim formu veri kaydı



Şekil 2.2. Node Js Uygulama Logosu

2.3. Veritabanı – MongoDB

Veri yönetimi için **MongoDB** NoSQL veritabanı kullanılmıştır. MongoDB, belgeler şeklinde veri saklama modeliyle esnek bir yapı sunar. Uygulamada kullanılan veritabanı modelleri Mongoose kütüphanesi ile tanımlanmıştır.

Veritabanında aşağıdaki koleksiyonlar yer alır:

- **User:** Kullanıcı bilgileri ve giriş bilgileri
 - **Pet:** Hayvan adı, türü, açıklaması ve fotoğraf URI'si
 - **Notification:** Sistem tarafından gönderilen bildirimler
- MongoDB'nin tercih edilme sebepleri:
- JSON formatında esnek veri modeli
 - Ölçeklenebilirlik ve performans
 - Mongoose ile kolay modelleme ve doğrulama



Şekil 2.3. Mango DB Uygulama Logosu

2.4. API İletişimi

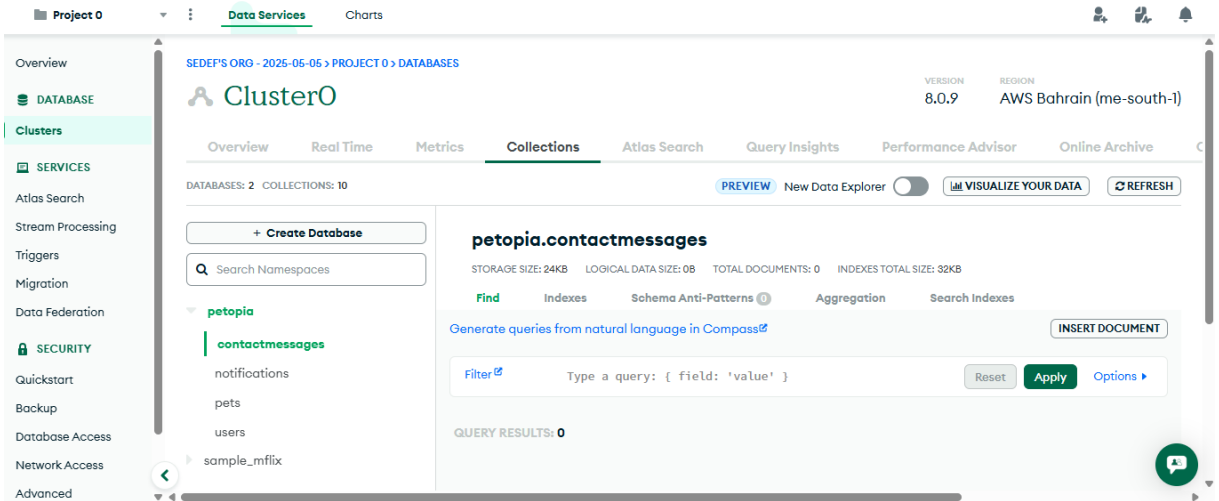
Petopia uygulamasında istemci (mobil uygulama) ile sunucu (backend) arasındaki veri alışverişi, RESTful API'ler aracılığıyla sağlanmaktadır. Bu yapı, kullanıcı işlemlerinin ve veri yönetiminin hızlı, güvenli ve organize bir şekilde gerçekleştirilmesine olanak tanımaktadır.

API'ler üzerinden gerçekleştirilen başlıca işlemler şunlardır:

- Kullanıcı kayıt ve giriş işlemleri (JWT token ile kimlik doğrulama)
- Hayvan ekleme, listeleme ve detay görüntüleme
- Bildirim oluşturma ve görüntüleme
- Kayıp hayvan bildirimi ve iletişim formu gönderimi
-

Tüm istekler JSON formatında iletilmekte; hatalı veya eksik girişlerde kullanıcıya uygun geri bildirim mesajları sunulmaktadır. Güvenlik açısından, yalnızca kimliği doğrulanmış kullanıcıların belirli API uç noktalarına erişebilmesi sağlanmıştır.

Uç noktalar, Express.js framework'ü kullanılarak yapılandırılmıştır ve istemciden gelen talepler HTTP metodlarına (GET, POST, PUT, DELETE) uygun şekilde yanıtlanmaktadır. Bu yapı, uygulamanın modüler, sürdürülebilir ve geliştirilebilir olmasını sağlamaktadır.



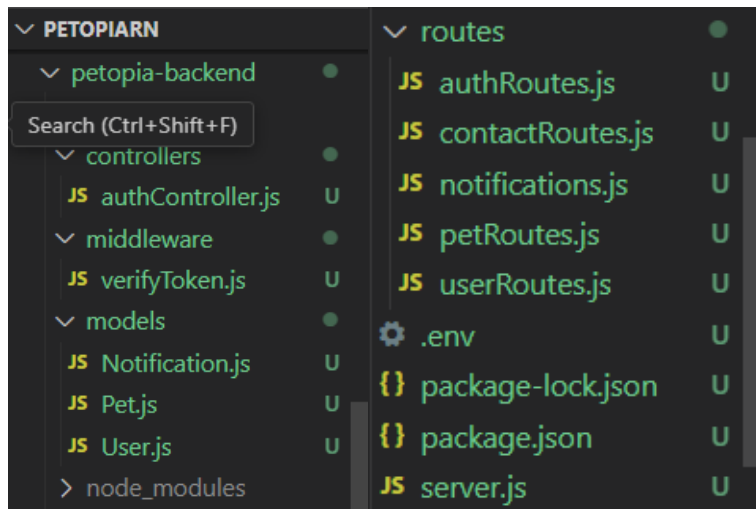
Şekil 2.4. Veri Tabanı Table Yapısı

Proje Dizin Yapısı:

Backend tarafı, modüler ve sürdürülebilir bir mimariyle organize edilmiştir. Ana klasör olan petopia-backend içerisinde görevlerine göre ayrılmış alt klasörler yer almaktadır:

- **controllers/:** Uygulamanın iş mantığını yöneten dosyaları içerir. Örneğin, authController.js dosyasında kullanıcı kayıt ve giriş işlemleri yönetilmektedir.
- **middleware/:** Orta katman fonksiyonlarının bulunduğu dizindir. verifyToken.js dosyası, JWT doğrulaması yaparak korumalı rotalara erişimi denetler.
- **models/:** MongoDB koleksiyonlarını tanımlayan Mongoose şemaları burada yer alır. Kullanıcı, hayvan ve bildirim modelleri ayrı ayrı yapılandırılmıştır.
- **routes/:** API uç noktalarının tanımlandığı klasördür. Kullanıcı, hayvan, bildirim ve iletişim gibi her işlev için ayrı bir yönlendirme dosyası mevcuttur.
- **server.js:** Uygulamanın giriş noktasıdır. Express sunucusu burada başlatılır ve gerekli middleware ile rotalar burada bağlanır.
- **.env:** Ortam değişkenlerinin saklandığı dosyadır. Gizli anahtarlar ve bağlantı adresleri bu dosyada yönetilir.

Bu yapı sayesinde, proje hem okunabilirlik hem de ölçeklenebilirlik açısından güçlü bir mimariye sahiptir. Her sorumluluğun ayrı bir dosyada tanımlanmış olması, bakım ve geliştirme süreçlerini kolaylaştırmaktadır.



Şekil 2.5. Backend Dosyasının Directory Yapısı

2.5. Frontend: React Native

Petopia uygulamasının kullanıcı arayüzü, mobil cihazlar için çapraz platform desteği sunan React Native teknolojisi ile geliştirilmiştir. Bu yapı sayesinde hem Android hem de iOS işletim sistemlerinde aynı kod tabanı üzerinden sorunsuz bir kullanıcı deneyimi sunulmaktadır. Proje yapısı app/ dizini altında organize edilmiştir ve Expo Router kullanılarak sayfa yönlendirmeleri sağlanmaktadır.

Proje Dizin Yapısı:

- **context/:** Uygulama genelinde oturum yönetimi için kullanılan AuthContext.tsx bileşenini içerir. Bu yapı sayesinde kullanıcı oturumu (login/logout) global olarak yönetilir.
- **profiles/:** Hayvan detay profillerini yöneten dinamik sayfa ([id].tsx) burada bulunur.
- **userprofiles/:** Kullanıcıya ait profil düzenleme ve bilgi güncelleme ekranları bu klasör altında yer alır.
- **sayfalar (tsx):**
 - **home.tsx:** Ana sayfa – yeni eklenen hayvanlar ve kayıp hayvanlar burada listelenir.
 - **addpet.tsx:** Yeni hayvan ekleme ekranı.
 - **profile.tsx:** Kullanıcının uygulama içi profil yönetim alanı.
 - **notifications.tsx:** Kullanıcıya sistemden gelen bildirimlerin listelendiği ekran.
 - **lostpets.tsx, diseases.tsx:** Kaybolan hayvanlar ve hayvan hastalıkları ile ilgili bilgi ekranları.
 - **contact.tsx, about.tsx, welcomescreen.tsx:** Bilgilendirme ve yönlendirme amaçlı temel ekranlar.
 - **signin.tsx, signup.tsx, forgotpassword.tsx:** Kimlik doğrulama ve şifre yenileme işlemleri için kullanıcı arayüzleri.

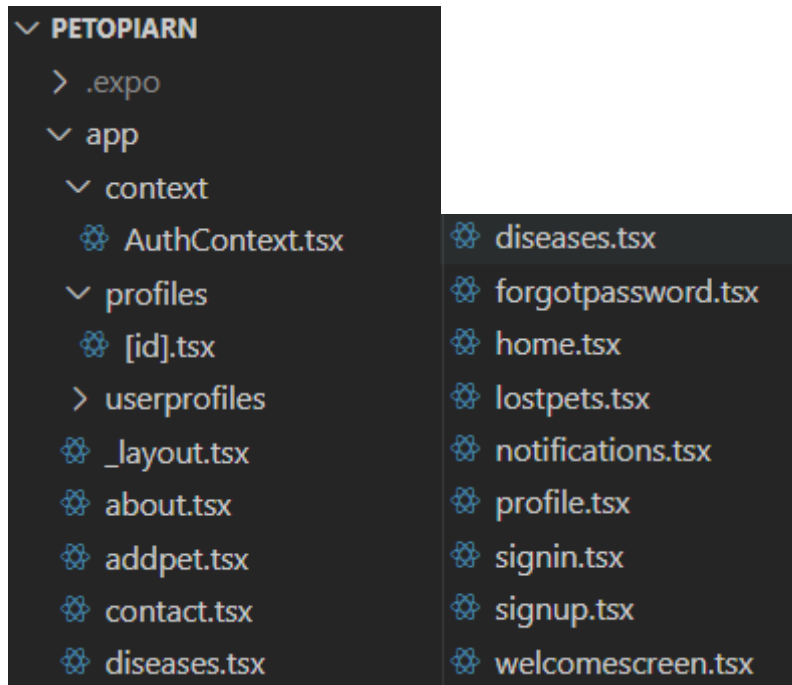
Kullanılan Teknolojiler ve Bileşenler:

- **React Native:** Temel UI ve mobil uygulama mantığı için.
- **Expo:** Uygulamanın hızlı geliştirme, test ve dağıtım sürecinde kullanılmıştır.

- **Expo Router:** Sayfalar arası yönlendirme, dinamik rota oluşturma ve geçiş animasyonları için.
- **useState, useEffect, useContext** gibi **React** hook'ları ile durum ve veri yönetimi.
- **Animated API:** Kayar menü gibi geçiş efektlerinde kullanıcı deneyimini iyileştirmek için kullanılmıştır.
- **Fetch API: Backend (Node.js)** ile haberleşme için **RESTful API** isteklerinde kullanılır.
- **JWT** ile oturum yönetimi: **AuthContext.tsx** ile birlikte, kullanıcının giriş durumunu yönetmek için **AsyncStorage** üzerinde token saklanır.

Arayüz Özellikleri:

- Modern ve kullanıcı dostu tasarım
- Mobil uyumlu, responsive yapı
- Kategori bazlı filtreleme ve arama çubuğu (özellikle home.tsx'te)
- Şifre korumalı giriş sistemi
- Gerçek zamanlı bildirim görüntüleme



Şekil 2.6. Frontend Dosyasının Directory Yapısı

3. VERİ TABANI TASARIMI VE TABLOLAR

Bu bölümde, Petopia uygulamasının veri tabanı tasarımı ve ilgili tablolarının yapısı detaylı bir şekilde açıklanacaktır.

- **Kullanıcılar Kayıt Tablosu:**

The screenshot shows the 'petopia.users' database table. The left sidebar lists namespaces: 'petopia' (expanded), 'contactmessages', 'notifications', 'pets', 'users' (selected), and 'sample_mflix'. The main panel displays the table's metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 280B, TOTAL DOCUMENTS: 2, and INDEXES TO. Below this are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns (0)', and 'Aggregation'. A search bar with the placeholder 'Type a query: { field: 'value' }' is present. The 'QUERY RESULTS: 1-2 OF 2' section shows a single document with the following fields: '_id: ObjectId('6818cb308643843b2584d470')', 'name: "Test User"', 'email: "test@example.com"', and 'password: "123456"'.

Şekil 3.1. Kullanıcılar Kayıt Tablosu: Kullanıcıların hesap bilgilerini depolar.

- **Hayvanlar Tablosu:**

The screenshot shows the 'petopia.pets' database table. The left sidebar lists namespaces: 'petopia' (expanded), 'contactmessages', 'notifications', 'pets' (selected), 'users', and 'sample_mflix'. The main panel displays the table's metadata: STORAGE SIZE: 3.22MB, LOGICAL DATA SIZE: 1.4MB, TOTAL DOCUMENTS: 9, and INDEXES TOTAL SIZE: 36KB. Below this are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns (0)', 'Aggregation', and 'Search Indexes'. A search bar with the placeholder 'Type a query: { field: 'value' }' is present, along with 'Reset' and 'Apply' buttons. The 'QUERY RESULTS: 1-2 OF 2' section shows a single document with the following fields: '_id: ObjectId('68320f1a429f8eeffae95c3d')', 'name: "Kiwi"', 'type: "Bird"', 'description: "A bird."', 'imageUri: "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAkGBxMSEhUQEBI..."', 'createdAt: 2025-05-24T18:25:30.257+00:00', 'updatedAt: 2025-05-24T18:25:30.257+00:00', and '__v: 0'.

Şekil 3.2. Hayvanlar Tablosu: Hayvanlarla ilgili bilgileri içerir.

- İletişim Gönderim Tablosu:

The screenshot shows the 'petopia.contactmessages' table in a database interface. The left sidebar lists the database 'petopia' and its collections: 'contactmessages', 'notifications', 'pets', 'users', and 'sample_mflix'. The main panel displays the table's metadata: STORAGE SIZE: 24KB, LOGICAL DATA SIZE: 0B, TOTAL DOCUMENTS: 0, and INDEXES TOTAL SIZE: 32KB. Below the metadata are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar with a 'Filter' button and a 'Reset' button is present. The query results section shows 'QUERY RESULTS: 1-1 OF 1' and a single document with the following fields: '_id: ObjectId('68376f2060eae3f1508104c2')', 'name: "Sedef"', 'email: "sedefkjamili@gmail.com"', 'message: "I found a dog in the part, I need to adress it in your app!"', 'createdAt: 2025-05-28T20:16:32.293+00:00', and '__v: 0'.

Şekil 3.3. İletişim Gönderim Tablosu: İletişim kurarken form verilerini saklar.

- Bildirim Tablosu:

The screenshot shows the 'petopia.notifications' table in a database interface. The left sidebar lists the database 'petopia' and its collections: 'contactmessages', 'notifications', 'pets', 'users', and 'sample_mflix'. The main panel displays the table's metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 231B, TOTAL DOCUMENTS: 2, and INDEXES TOTAL SIZE: 36KB. Below the metadata are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar with a 'Filter' button and a 'Reset' button is present. The query results section shows 'QUERY RESULTS: 1-2 OF 2' and a single document with the following fields: '_id: ObjectId('6832228179af5d1f4e18c2e3')', 'title: "New Pet Added!"', 'body: "Kiwi the Bird has been added."', 'createdAt: 2025-05-24T19:48:17.692+00:00', and '__v: 0'.

Şekil 3.4. Bildirim Gönderim Tablosu: Yeni Hayvan Eklenince Verileri Saklar.

4. UYGULAMANIN GELİŞTİRİLMESİ

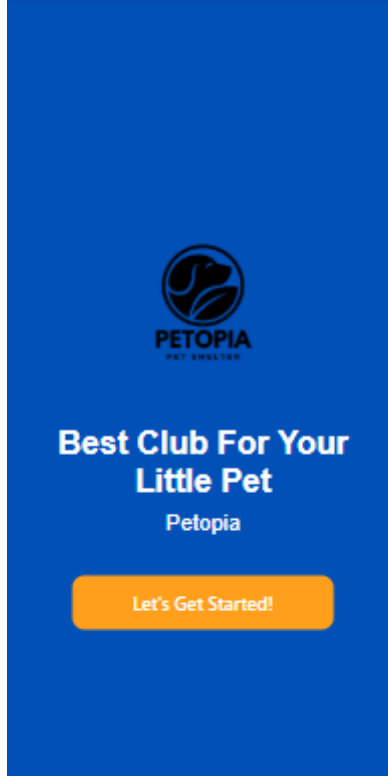
Bu bölümde, Petopia mobil uygulamasının ekran bazında tüm işlevleri açıklanmakta, her sayfanın ne amaçla geliştirildiği ve nasıl çalıştığı detaylandırılmaktadır.

4.1. Karşılama Ekranı – welcomescreen.tsx

Uygulamanın açılışında kullanıcıyı karşılayan bu ekran, sade ve dikkat çekici bir tasarıma sahiptir. Ana amacı, kullanıcıyı uygulamaya yönlendirmek ve ilk etkileşimi oluşturmaktır. Sayfa ortasında Petopia logosu ve slogan yer almakta; alt kısımda ise kullanıcıyı “Ana Sayfa”ya yönlendiren bir buton bulunmaktadır.

- **Öne çıkan özellikler:**

- Sabit logo ve marka tanıtımı
- "Let's Get Started" butonu ile yönlendirme (/home)
- Modern, minimal ve mobil uyumlu tasarım



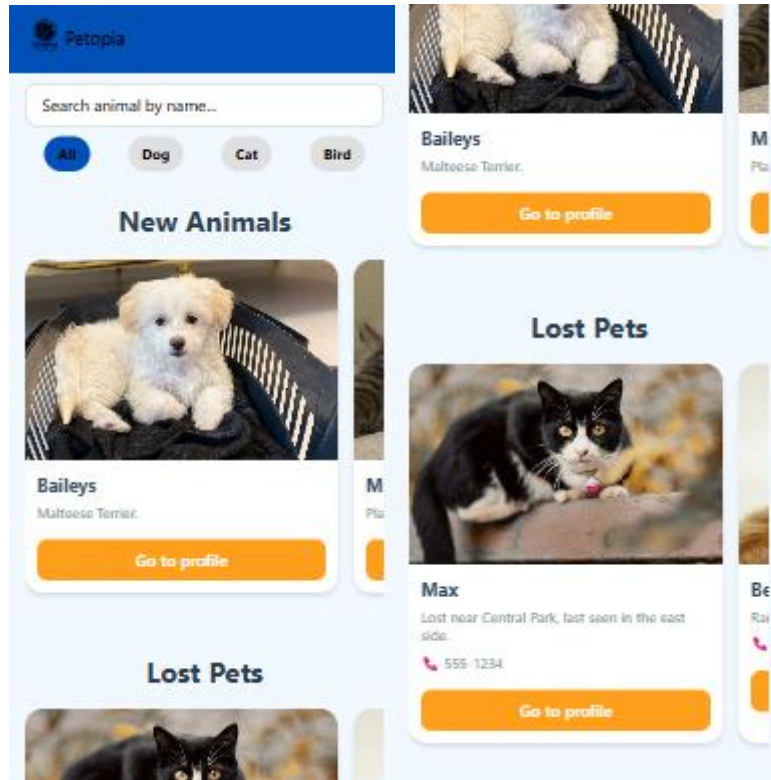
Şekil 4.1. Karşılama Ekranın Görüntüsü.

4.2. Ana Sayfa – home.tsx

Ana sayfa, kullanıcıların yeni eklenen hayvanları ve kayıp hayvanları hızlıca görebildiği ana kontrol panelidir. Arama çubuğu ve kategori butonları sayesinde hayvanlar filtrelenebilir. Hayvan kartlarına tıklayarak detay sayfasına geçiş yapılabilir.

Öne çıkan özellikler:

- Kategori bazlı filtreleme (Kedi, Köpek, Kuş)
- Arama çubuğu ile isimle arama
- Kayıp hayvan listesi
- Navigasyon menüsü ve profil erişimi



Şekil 4.2. Ana Sayfanın Görüntüsü.

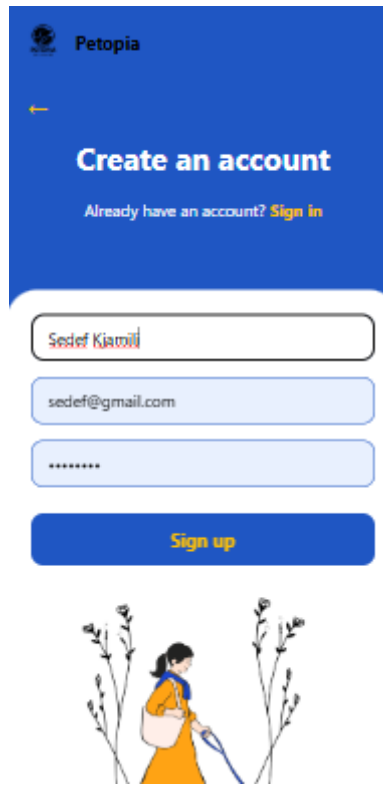
4.3. Kayıt Sayfası – signup.tsx

Bu ekran, yeni kullanıcıların sisteme kayıt olmasını sağlar. Kullanıcılardan ad-soyad, e-posta adresi ve şifre bilgileri istenir. Form eksiksiz doldurulduğunda kayıt işlemi gerçekleştirilir ve başarılı bir şekilde tamamlandığında kullanıcı signin.tsx sayfasına yönlendirilir.

Giriş yapılmasını teşvik etmek amacıyla alt kısımda “Already have an account?” yönlendirmesi yer almakta, bu bağlantı ile kullanıcı mevcut bir hesabı varsa doğrudan giriş ekranına geçebilmektedir.

Öne çıkan özellikler:

- Backend’e POST isteği ile kayıt (API: /api/auth/signup)
- Eksik alan kontrolü ve kullanıcıya uyarı mesajları
- Başarılı kayıt sonrası yönlendirme
- Modern tasarımlı giriş bağlantısı ve marka öğeleri



Şekil 4.3. Kayıt Sayfasının Görüntüsü.

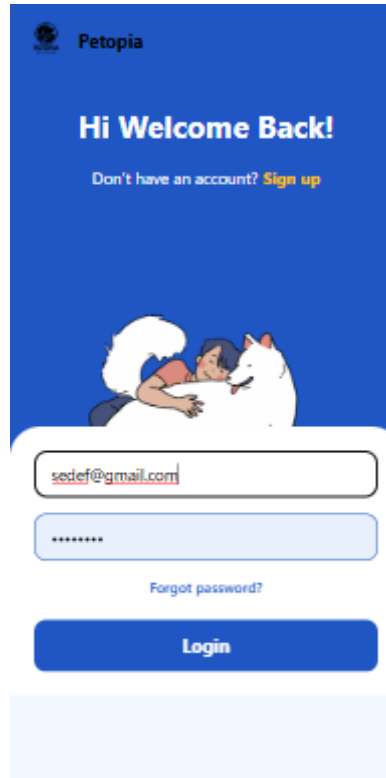
4.4. Giriş Sayfası – signin.tsx

Bu ekran, kullanıcıların sisteme giriş yapmasını sağlayan oturum açma arayüzüdür. Kullanıcılar e-posta ve şifre bilgilerini girerek sisteme erişim sağlar. Giriş işlemi başarıyla tamamlandığında, kullanıcının JWT token’ı yerel belleğe (AsyncStorage) kaydedilir ve kullanıcı profile.tsx sayfasına yönlendirilir.

Eksik veya hatalı bilgiler girildiğinde kullanıcıya anlamlı hata mesajları gösterilir. Ayrıca, şifresini unutan kullanıcılar için “Forgot password?” bağlantısı, yeni kullanıcılar için ise signup.tsx sayfasına yönlendiren “Sign up” bağlantısı sunulmaktadır.

Öne çıkan özellikler:

- JWT tabanlı oturum yönetimi
- Form doğrulama ve kullanıcı bilgilendirme
- Şifre yenileme bağlantısı (forgotpassword.tsx)
- Navigasyon menüsü üzerinden diğer sayfalara geçiş



Şekil 4.4. Giriş Yap Sayfasının Görüntüsü.

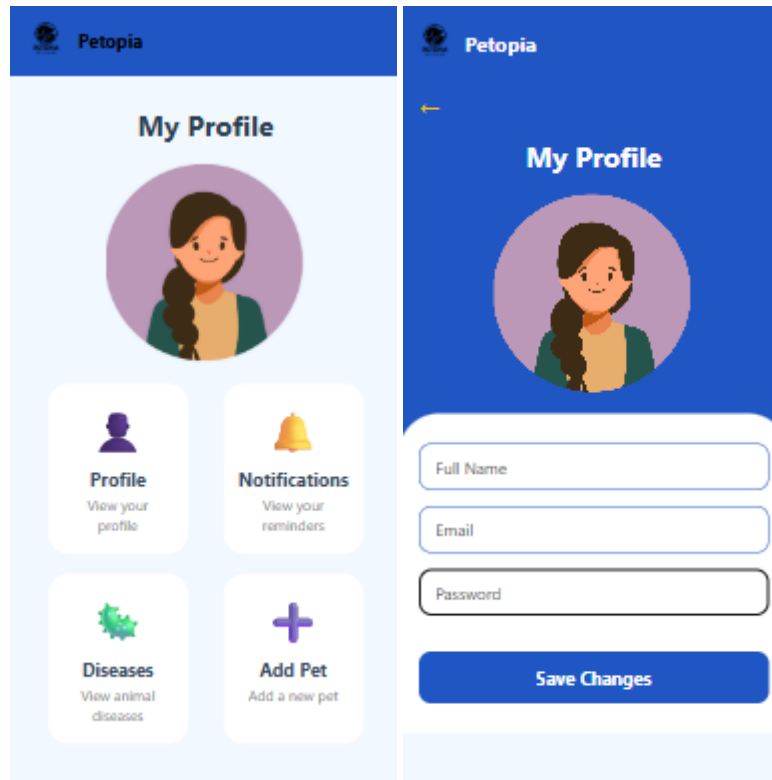
4.5. Profil Sayfası – profile.tsx

Profil ekranı, kullanıcıya ait ana kontrol paneli görevindedir. Giriş yaptıktan sonra kullanıcı bu sayfaya yönlendirilir. Buradan uygulamadaki çeşitli özelliklere (bildirimler, hastalık bilgileri, hayvan ekleme vb.) erişim sağlanır. Aynı zamanda oturum kapatma işlemi de bu ekrandan gerçekleştirilebilir.

Kullanıcı arayüzü kart bazlı bir düzenle tasarlanmış olup, her kart ilgili sayfaya yönlendirme yapar. Uygulama boyunca yapılan işlemlerin merkezi bağlantı noktası burasıdır.

Öne çıkan özellikler:

- Kullanıcının oturum durumu kontrolü (JWT token geçerliliği)
- Bildirimler, hastalıklar, hayvan ekleme gibi işlemlere erişim
- Kart tabanlı modern tasarım
- Çıkış butonu ile oturumu kapatma ve login ekranına yönlendirme



Şekil 4.5. Profil Sayfasının Görüntüsü.

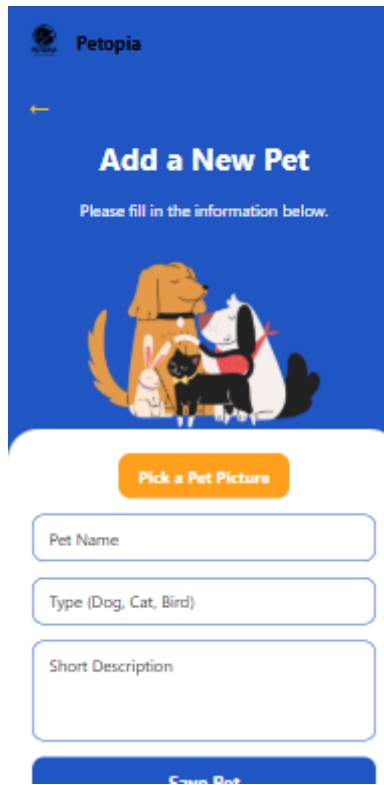
4.6. Hayvan Ekleme Sayfası – addpet.tsx

Bu sayfa, kullanıcıların sisteme yeni bir hayvan eklemesini sağlar. Ad, tür, açıklama ve fotoğraf gibi temel bilgiler doldurularak form gönderildiğinde, sunucuya bir POST isteği yapılır ve veriler MongoDB'ye kaydedilir. Ayrıca, başarılı ekleme sonrasında sistem tarafından otomatik olarak bir bildirim oluşturulur.

Kullanıcılar galeri üzerinden hayvan fotoğrafı seçebilir. Eksik alanlarla form gönderilmeye çalışıldığında uyarı mesajları ile bilgilendirme yapılır. Kayıt işlemi tamamlandığında kullanıcı profile.tsx sayfasına yönlendirilir.

Öne çıkan özellikler:

- Resim seçimi (Expo ImagePicker ile)
- Form doğrulama ve eksik alan kontrolü
- API'ye hayvan ekleme ve bildirim oluşturma (POST /api/pets/add)
- Başarılı işlem sonrası yönlendirme ve geri bildirim



Şekil 4.6. Hayvan Ekle Sayfasının Görüntüsü.

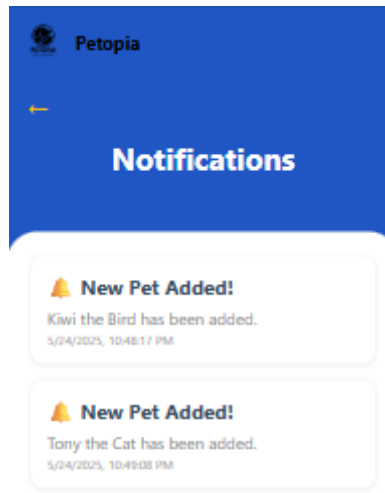
4.7. Bildirimler Sayfası – notifications.tsx

Bu sayfa, kullanıcıya sistem tarafından gönderilen bildirimleri listelemek üzere geliştirilmiştir. Yeni bir hayvan eklendiğinde veya sistemsel bir olay gerçekleştiğinde otomatik olarak oluşturulan bildirimler bu alanda sıralanır.

Bildirimler, en yeni olandan başlayarak sırayla gösterilir. Kullanıcı uygulamaya giriş yaptığında burada yer alan mesajları okuyabilir. Eğer sistemde bildirim yoksa "No notifications yet" şeklinde kullanıcı bilgilendirilir.

Öne çıkan özellikler:

- Backend'den veri çekme (GET /api/notifications)
- Bildirimleri tarihe göre ters sıralı listeleme
- Boş liste durumunda kullanıcıya bilgilendirme
- Temiz ve okunabilir kart tasarımı



Şekil 4.7. Bildirim Sayfasının Görüntüsü.

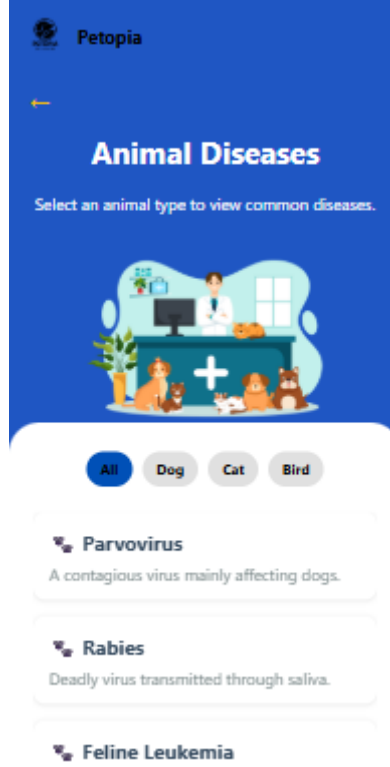
4.8. Hastalıklar Sayfası – diseases.tsx

Bu ekran, evcil hayvanlarda yaygın görülen hastalıklar hakkında bilgi sunmak amacıyla geliştirilmiştir. Kullanıcılar, sayfa başındaki filtre butonlarını kullanarak köpek, kedi veya kuş gibi hayvan türlerine göre hastalıkları filtreleyebilir.

Her hastalık kartında, hastalığın adı ve kısa bir açıklaması yer almaktadır. Böylece kullanıcılar temel bilgiye kolayca erişebilir. Filtreleme işlemi yerel veri üzerinden çalışır ve ekran performans açısından oldukça akıcıdır.

Öne çıkan özellikler:

- Hayvan türüne göre filtreleme (All, Dog, Cat, Bird)
- Temiz ve okunabilir kart yapısı
- Sabit veri kullanımı ile hızlı erişim
- Kullanıcı dostu kategori seçimi



Şekil 4.8. Hayvan Hastalıkları Sayfasının Görüntüsü.

4.9. Hayvan Detay Sayfası – [id].tsx

Bu dinamik sayfa, kullanıcıların bir hayvana ait detaylı bilgiye ulaşmasını sağlar. home.tsx sayfasındaki kartlardan birine tıklanarak bu sayfaya yönlendirilir. Hayvanın adı, türü, yaşı, kilosu, ihtiyaçları, fotoğrafı ve iletişim numarası gibi bilgiler gösterilir. Ayrıca “Adopt me” butonuna tıklayan kullanıcıya iletişim numarası bildirilir. Bu ekran, sahiplendirme kararını verecek kullanıcılar için son derece kritik bir bilgilendirme alanıdır.

Öne çıkan özellikler:

- Dinamik URL ile içerik yükleme (parametre üzerinden)

- Hayvanın detaylı bilgileri ve görseli
- İletişim bilgisi ile kullanıcıya yönlendirme
- Buton ile “adopt me” işlemi için rehberlik



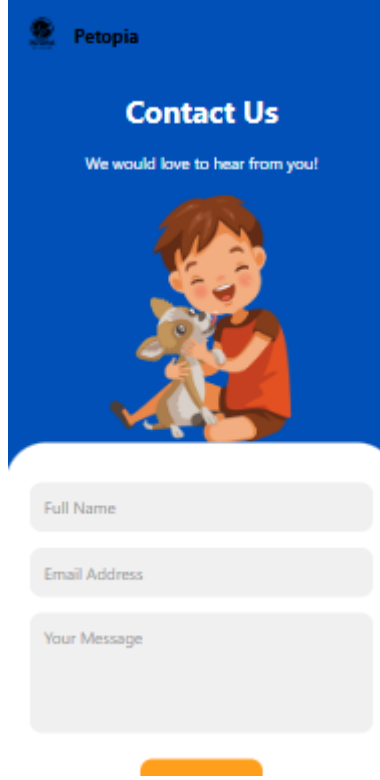
Şekil 4.9. Hayvan Detay Sayfasının Görüntüsü.

4.10. İletişim Sayfası – contact.tsx

Bu sayfa, kullanıcıların uygulama geliştiricileriyle veya barınak yönetimiyle iletişim kurabilmesi amacıyla tasarlanmıştır. Formda ad, e-posta ve mesaj alanları bulunmaktadır. Tüm alanlar zorunludur ve eksik bilgi durumunda kullanıcı uyarılır. Form gönderildiğinde bilgiler backend'e iletilir ve MongoDB'ye kaydedilir. Başarılı gönderim sonrası kullanıcıya bilgi mesajı gösterilir.

Öne çıkan özellikler:

- Basit ve anlaşılır iletişim formu
- Geri bildirim mesajları
- Veritabanına kayıt işlemi (POST /api/contact)
- Görsel destekli, mobil uyumlu tasarım

The image shows a mobile app interface for a contact form. At the top, there is a blue header with the Petopia logo and the text 'Contact Us'. Below the header, there is a cartoon illustration of a boy hugging a dog. The form itself is white and contains three input fields: 'Full Name', 'Email Address', and 'Your Message'. A yellow button is located at the bottom of the form.

Şekil 4.10. İletişim Sayfasının Görüntüsü.

4.11. Hakkımızda Sayfası – about.tsx

Petopia uygulamasının amacı, vizyonu ve temel özelliklerinin tanıtıldığı bilgilendirme ekranıdır. Kullanıcıya sade bir şekilde uygulamanın ne yaptığı, hangi alanlarda hizmet sunduğu anlatılır. Ayrıca platformun kullanıcıya sağladığı başlıca avantajlar da vurgulanır. Sayfa alt kısmında, uygulamanın işlevleri simgelerle desteklenmiş açıklamalar eşliğinde sıralanır.

Öne çıkan özellikler:

- Uygulama hakkında genel bilgilendirme
- Öne çıkan işlevlerin vurgulanması (filtreleme, ilan, iletişim)
- Görsel destekli açıklama yapısı
- Tanıtım odaklı sade tasarım



Şekil 4.11. Hakkımızda Sayfasının Görüntüsü.

SONUÇ

Petopia mobil uygulaması, sahihsiz hayvanların dijital ortamda tanıtılması, sahiplenme süreçlerinin hızlandırılması ve barınak yönetiminin dijitalleştirilmesi hedefleri doğrultusunda başarılı bir şekilde geliştirilmiştir. Uygulama, kullanıcı dostu arayüzü ve güçlü teknik altyapısıyla hem bireysel kullanıcıların hem de barınak çalışanlarının ihtiyaçlarını karşılayacak şekilde tasarlanmıştır.

React Native ile geliştirilen ön yüz, Android ve iOS platformlarında çalışacak şekilde optimize edilmiş; Expo Router ve dinamik sayfa yönetimi sayesinde kullanıcı deneyimi artırılmıştır. Backend tarafında Node.js ve Express.js kullanılarak RESTful API mimarisi oluşturulmuş, MongoDB veritabanı ile esnek ve ölçeklenebilir bir veri yapısı sağlanmıştır. Kullanıcı kimlik doğrulaması için JWT tabanlı güvenli oturum yönetimi uygulanmıştır.

Uygulama içerisinde kayıt, giriş, profil yönetimi, hayvan ekleme, bildirim alma, hastalık bilgileri görüntüleme, kayıp hayvan bildirimi, iletişim ve bilgilendirme gibi pek çok işlevsel ekran entegre edilmiştir. Bu sayede kullanıcıların sistemle etkileşimi artırılmış, sahiplendirme süreci dijital olarak uçtan uca yönetilebilir hale getirilmiştir.

Petopia, toplumsal bir soruna dijital bir çözüm sunarken, aynı zamanda kullanıcı deneyimini ve teknik güvenliği ön planda tutarak mobil uygulama geliştirme alanında etkili bir örnek teşkil etmektedir. Uygulamanın daha geniş kitlelere ulaştırılması ve barınak sistemlerine entegre edilmesi halinde, hayvan refahına doğrudan katkı sağlayacak önemli bir araç olma potansiyeline sahiptir.

KAYNAKLAR

Meta Platforms, Inc. (2024). *React Native Documentation*.

OpenJS Foundation. (2024). *Node.js Documentation*.

Express Team. (2024). *Express.js Guide*.

MongoDB Inc. (2024). *MongoDB Documentation*.

Valeri, A. (2024). *Mongoose ODM Documentation*.

Expo. (2024). *Expo Documentation*.

Auth0. (2024). *JWT: Introduction to JSON Web Tokens*.

npmjs.com. (2024). *bcrypt – A library to help you hash passwords*.

React Native Community. (2024). *AsyncStorage for React Native*.

Expo. (2024). *Image Picker API Reference*.

Mozilla Foundation. (2024). *REST API Design Best Practices*.