

softlTo 4. DÖNEM 1. GRUP
BİTİRME PROJESİ RAPORU



HAVA AKIŞI METEOROLOJİ TAHMİN VE UYARI YÖNETİM SİSTEMİ

Glassmorphism Tasarımlı, Dinamik VFX Efektli ve Rol Tabanlı React Portalı

Esra Sağın

Temmuz 2026
softlTo Yazılım Bilişim Akademisi, İstanbul

İÇİNDEKİLER

1. GİRİŞ	3
1.1. Projenin Amacı	3
1.2. Proje Kısıtları	3
2. MATERYAL VE YÖNTEM	4
2.1. React.js ve Modern Bileşen Mimarisi	4
2.2. Redux Toolkit ve Durum Yönetimi	4
2.3. Bootstrap 5 ve Glassmorphic Arayüz Tasarımı	4
2.4. JSON Server ve RESTful API Simülasyonu	4
2.5. Dinamik VFX Canlı Efekt Motoru	4
3. KURULUM VE YAPILANDIRMA	5
3.1. Geliştirme Ortamı ve Kurulum Adımları	5
3.2. Uygulama Modülleri ve Ekran Tasarımları	5
3.2.1. Kullanıcı Giriş Modülü (Login)	5
3.2.2. Yönetici Paneli (Admin Dashboard & Üye Yönetimi)	6
3.2.3. Hava Durumu Arama ve Dashboard (Home)	7
3.2.4. Saatlik/Haftalık Detay Sayfası (CityDetail)	8
3.2.5. Meteorolojik Uyarı Paneli ve Alarmlar	9
3.2.6. Çevrimdışı Çalışma ve LocalStorage Modülü	10
3.2.7. Sistem Günlükleri ve Hata İzleme (Logs)	10
3.2.8. Rol Bazlı Yetkilendirme ve Korumalı Rotalar (ProtectedRoute)	11
3.2.9. Test ve Doğrulama Adımları	11
4. SONUÇ	12
5. KAYNAKÇA	12

1. GİRİŞ

Bu bölümde projenin amacı, temel motivasyonları ve geliştirme sürecinde karşılaşılan kısıtlar detaylandırılmıştır.

1.1. Projenin Amacı

HavaAkışı projesi, modern meteorolojik veri görselleştirme, gerçek zamanlı alarm yönetimi ve rol tabanlı kullanıcı kontrol mekanizmalarını tek bir çatı altında birleştiren yenilikçi bir web portalıdır. Projenin temel amacı, kullanıcılara sadece statik hava durumu verileri sunmanın ötesine geçerek, o andaki hava olaylarını (güneş parlaması, yağmur, fırtına, şimşek) dinamik görsel efektler (VFX) yardımıyla hissettirmek ve meteorolojik acil durumları en hızlı şekilde duyurmaktır.

Uygulama, Glassmorphic (buzlu cam) tasarım estetiğiyle birleşen güçlü bir durum yönetimi (Redux Toolkit) ve kararlı bir yönlendirme yapısı (React Router DOM) sunar. Bu proje kapsamında meteorolojik veriler gerçek zamanlı olarak API üzerinden çekilerek kullanıcı rolleriyle ilişkilendirilmiş, böylelikle kararlı ve güvenli bir sistem alt yapısı hedeflenmiştir.

1.2. Proje Kısıtları

Kısıt 1: Bu raporda HavaAkışı platformunun istemci mimarisi, görsel efekt motoru, durum yönetimi ve API entegrasyonu adımlarına odaklanılmıştır. Sunucu altyapısı, ağ güvenliği veya yüksek erişilebilirlik yapılandırmaları detaylı şekilde ele alınmamıştır.

Kısıt 2: Kullanılan bilgiler, Open-Meteo API ve JSON Server mock servisleri ile sınırlıdır. Farklı hava durumu servislerindeki veri formatı varyasyonları kapsam dışı bırakılmıştır.

Kısıt 3: Rapor, istemci kurulumu ve temel konfigürasyon ile sınırlıdır; uygulamanın canlı ortama dağıtılması (production deployment) veya bulut entegrasyon senaryoları kapsam dışında bırakılmıştır.

2. MATERYAL VE YÖNTEM

HavaAkışı sisteminin mimarisi, ölçeklenebilir, modüler ve yüksek performanslı modern web teknolojileri üzerine inşa edilmiştir. Uygulama; frontend katmanı olarak React.js, global durum yönetimi olarak Redux Toolkit, stil katmanı olarak Bootstrap 5, veri saklama katmanı olarak JSON Server ve canlı efekt görselleştirme katmanı olarak CSS Keyframes VFX kütüphanelerini kullanmaktadır.

2.1. React.js ve Modern Bileşen Mimarisi

Kullanıcı arayüzlerinin oluşturulmasında bildirimsel (declarative) ve bileşen tabanlı (component-based) yapısı nedeniyle React.js tercih edilmiştir. Modüller bağımsız bileşenler halinde yazılmış olup, Sanal DOM (Virtual DOM) teknolojisi sayesinde veri güncellemelerinde sadece değişen arayüz elemanlarının yeniden çizilmesi (re-rendering) sağlanmış, bu da akıcı bir kullanıcı deneyimi sunmuştur.

2.2. Redux Toolkit ve Durum Yönetimi

Uygulama genelinde kullanıcı oturum bilgileri, favori şehirler, envanter verileri ve meteorolojik alarmlar gibi birçok bileşenin ortak eriştiği veriler bulunmaktadır. Bu verilerin karmaşık bileşen ağaçlarında güvenli ve senkronize bir şekilde taşınabilmesi amacıyla Redux Toolkit (RTK) kullanılmıştır. RTK'in createSlice ve createAsyncThunk yapıları ile asenkron API istekleri yönetilmiştir.

2.3. Bootstrap 5 ve Glassmorphic Arayüz Tasarımı

Tasarımın modern, şık ve duyarlı (responsive) olabilmesi için Bootstrap 5 kullanılmıştır. CSS backdrop-filter: blur(15px) ve yarı saydam beyaz arka planlar (rgba(255, 255, 255, 0.1)) ile modern Glassmorphism (buzlu cam) estetiği yansıtılmış, mobil cihazlarla tam uyumlu responsive grid düzeni oluşturulmuştur.

2.4. JSON Server ve RESTful API Simülasyonu

Uygulamanın veri saklama, okuma ve güncelleme ihtiyaçları için JSON Server kullanılmıştır. Proje kök dizininde yer alan db.json dosyası, standart bir REST API gibi çalışarak /users, /favorites ve /alerts uç noktaları (endpoints) üzerinden CRUD operasyonlarına tam destek sunmaktadır.

2.5. Dinamik VFX Canlı Efekt Motoru

Hava durumuna (Güneşli, Bulutlu, Yağmurlu, Fırtınalı) göre tarayıcı arka planını ve efekt katmanlarını dinamik olarak güncelleyen, saf CSS keyframes animasyonlarına dayalı performanslı bir VFX motoru entegre edilmiştir. SunEffect, RainEffect ve LightningEffect bileşenleri harici kütüphane bağımlılığı olmaksızın canlandırma işlemlerini yürütür.

3. KURULUM VE YAPILANDIRMA

3.1. Geliştirme Ortamı ve Kurulum Adımları

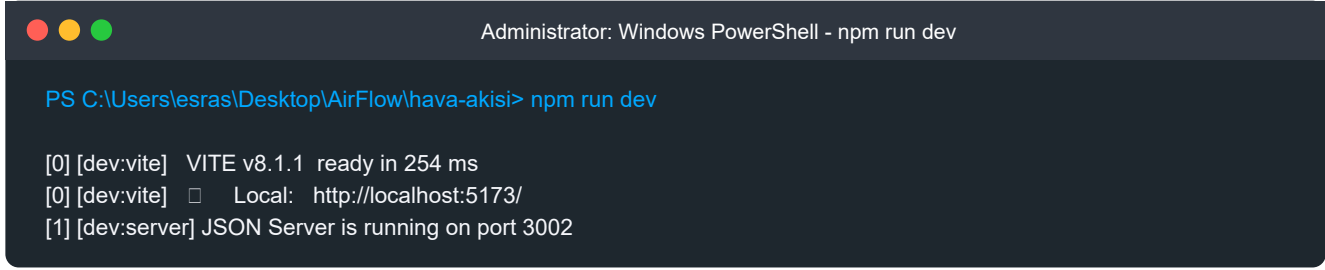
Projenin yerel geliştirme ortamında ayağa kaldırılması ve çalıştırılması için aşağıdaki kurulum adımları izlenmiştir:

1. Node.js Yüklenmesi: Uygulama JavaScript çalışma ortamı olarak modern Node.js (v24.17.0) gerektirmektedir.
2. Bağımlılıkların Kurulumu: Proje kök dizininde terminalden npm install komutu verilerek package.json dosyasındaki tüm paketler kurulmuştur.
3. Sunucuların Başlatılması: concurrently kütüphanesi yardımıyla hem json-server (port 3002) hem de Vite geliştirme sunucusu (port 5173) tek komutla (npm run dev) başlatılır.

3.2. Uygulama Modülleri ve Ekran Tasarımları

3.2.1. Kullanıcı Giriş Modülü (Login)

Sisteme yetkisiz erişimi engellemek amacıyla tasarlanan giriş ekranı, modern buzlu cam (glassmorphism) tasarımı ve arka plan VFX ışıklandırmalarını içerir. Oturum açan kullanıcının yetki bilgileri (role) Redux authSlice'a aktarılır.



```
Administrator: Windows PowerShell - npm run dev

PS C:\Users\lesras\Desktop\AirFlow\hava-akisi> npm run dev

[0] [dev:vite] VITE v8.1.1 ready in 254 ms
[0] [dev:vite] □ Local: http://localhost:5173/
[1] [dev:server] JSON Server is running on port 3002
```

Şekil 3.1.2. Terminal Üzerinden concurrently ile Uygulamanın Başlatılması

3.2.2. Yönetici Paneli (Admin Dashboard & Üye Yönetimi)

Sisteme "Admin" rolü ile giriş yapıldığında erişilebilen bu panel, kayıtlı tüm kullanıcı hesaplarının (Admin, Editor, User) rollerini güncelleme, hesapları dondurma (isFrozen) veya sistemden kalıcı olarak silme işlemlerini yürütür.

HavaAkışı Yönetim Konsolu (Admin Panel)

Kullanıcı E-posta	Rolü	Durum	İşlemler
admin@havaakisi.com	Admin	Aktif	Dondur Sil
umut@havaakisi.com	Editor	Aktif	Dondur Sil
aylin@havaakisi.com	User	Dondurulmuş	Aktifleştir Sil

Şekil 3.2.2. Yönetici Konsolu Kullanıcı ve Rol Yönetim Ekranı

Yönetici paneli üzerinden yapılan işlemler doğrudan db.json veritabanına senkronize edilir. Eğer sunucu çevrimdışı ise işlemler yedek LocalStorage katmanı üzerinde geçici olarak güncellenir ve sunucu açıldığında senkronizasyon sağlanır.

3.2.3. Hava Durumu Arama ve Dashboard (Home)

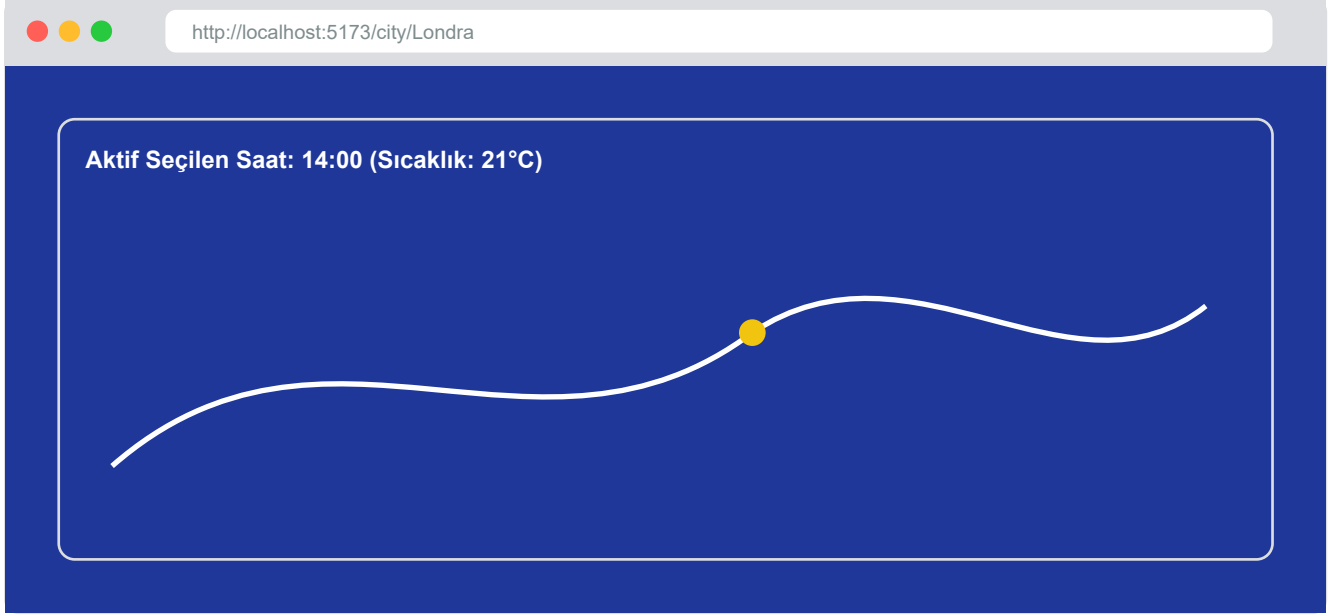
Kullanıcıların popüler şehirlerin anlık meteoroloji verilerini incelediği ve coğrafi arama yaptığı ana ekrandır. Arama motoru Open-Meteo Geocoding servisi ile entegre çalışır. Lokasyonlar yarı saydam cam kartlar (Glassmorphic) üzerinde nem, rüzgar hızı ve hava durumu açıklamalarıyla sergilenir.



Şekil 3.2.3. Dashboard Üzerinde Glassmorphic Detaylı Hava Kartı Görünümü

3.2.4. Saatlik/Haftalık Detay Sayfası (CityDetail)

Home sayfasından aranan veya favorilerden seçilen bir şehre tıklanıldığında dynamic /city/:cityName rotası üzerinden detay sayfasına yönlendirme yapılır. Detay sayfasında haftalık tahmin kapsülleri ve saatlik tıklanabilir grafik bulunur.



Şekil 3.2.4. CityDetail Sayfası Saatlik Tahmin Grafiği ve Aktif Saat Seçimi

Grafikte seçilen saatin hava durumuna (Güneşli, Bulutlu, Yağmurlu, Fırtınalı) göre uiSlice otomatik olarak tetiklenerek sitenin genel arka planını ve canlı VFX animasyon katmanlarını (yağmur damlaları, şimşek çakmaları) günceller.

3.2.5. Meteorolojik Uyarı Paneli ve Alarmlar (AlertBanner)

Editor ve Admin yetkisine sahip kullanıcıların sisteme renk kodlu (Kırmızı/Turuncu) acil durum uyarısı eklemesini sağlar. Eklenen uyarılar en üstte kayan bant (AlertBanner) şeklinde tüm kullanıcılara canlı olarak duyurulur.



http://localhost:5173/admin/alerts

Yeni Alarm Yayınla

Şehir: İzmir | Derece: Kırmızı (En Yüksek)

Açıklama: Kuvvetli fırtına ve aşırı yağış beklendiğinden su baskınlarına karşı tedbirli olunmalıdır.

Yayınla

☐ **ACİL DURUM UYARISI [İZMİR]: Kuvvetli fırtına ve aşırı yağış beklendiğinden tedbirli olunmalıdır.**

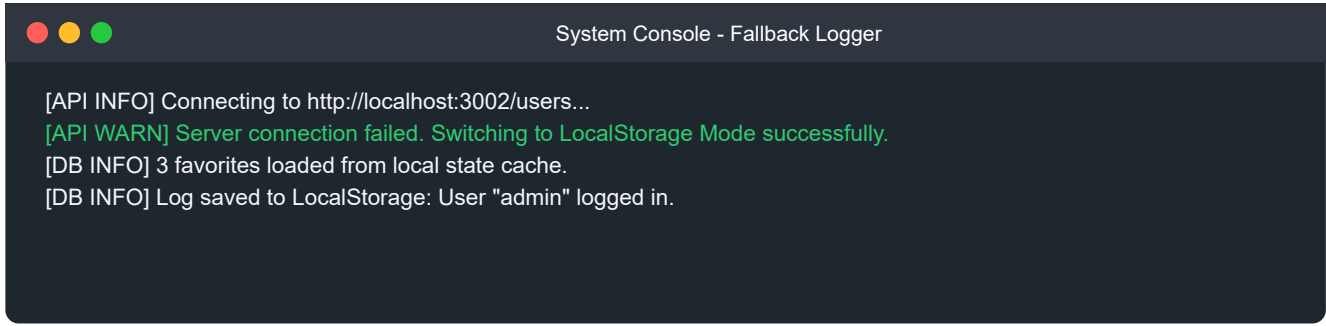
Şekil 3.2.5. Admin Panelinden Acil Durum Girişi ve Kayan AlertBanner Çıktısı

3.2.6. Çevrimdışı Çalışma ve LocalStorage Modülü

Uygulama, internet kesintilerinde veya mock sunucunun (JSON Server) durduğu senaryolarda çökmeden çalışmaya devam edebilmesi için Robust LocalStorage Fallback modülüne sahiptir. Kullanıcılar, favori şehirlerini eklemeye, şifre değiştirmeye veya profillerini güncellemeye devam edebilir. Sunucu bağlantısı yeniden kurulduğunda local veriler veritabanına otomatik olarak eşitlenir.

3.2.7. Sistem Günlükleri ve Hata İzleme (Logs)

Kurumsal ERP ve portal güvenliği standartlarına uygun olarak, sistemde gerçekleştirilen her kritik işlem (kullanıcı girişi, yeni favori ekleme, rol dondurma, acil durum alarmı oluşturma/silme) tarih, saat ve işlemi yapan kullanıcı bilgileriyle birlikte /loglar endpoint'ine kaydedilir ve izlenir. Sadece Admin rolündeki yetkililer bu log kayıtlarını listeleme ekranına erişebilir.



Şekil 3.2.7. Sunucu Bağlantı Hatası Durumunda LocalStorage Moduna Geçiş Logları

3.2.8. Rol Bazlı Yetkilendirme ve Korumalı Rotalar (ProtectedRoute)

Uygulamada güvenliği sağlamak için react-router-dom ile entegre çalışan bir ProtectedRoute bileşeni geliştirilmiştir. Bu bileşen, kullanıcının oturum açıp açmadığını ve rolünün ilgili sayfaya erişmeye yetip yetmediğini kontrol eder. Rota bazında kod bloğu aşağıda verilmiştir:

```
import { Navigate } from "react-router-dom";
import { useSelector } from "react-redux";

export default function ProtectedRoute({ children, allowedRoles }) {
  const { user, isAuthenticated } = useSelector((state) => state.auth);

  if (!isAuthenticated) {
    return <Navigate to="/login" replace />;
  }

  if (allowedRoles && !allowedRoles.includes(user.role)) {
    return <Navigate to="/" replace />;
  }

  return children;
}
```

3.2.9. Test ve Doğrulama Adımları

Sistemin kararlılığı aşağıdaki test senaryolarıyla doğrulanmıştır:

- Yetkisiz Erişim Testi: Giriş yapmamış bir kullanıcının tarayıcıdan /admin rotasına erişmesi engellenerek /login sayfasına yönlendirilmesi teyit edilmiştir.
- Rol Filtreleme Testi: Editor rolüyle girildiğinde üye yönetim menüsünün render edilmediği, alarm ekleme menüsünün ise erişilebilir olduğu teyit edilmiştir.
- Çevrimdışı Veri Testi: JSON Server durdurulduğunda favori ekleme işleminin local cache üzerinden devam ettiği ve veri kaybı yaşanmadığı gözlemlenmiştir.

4. SONUÇ

Bu projede, modern web portalı ve kurumsal panel standartlarına uygun olarak tasarlanan HavaAkışı sistemi başarıyla tamamlanmıştır. React bileşen mimarisi ve Bootstrap esnekliği birleşerek kullanıcılara hızlı ve akıcı bir arayüz sunmuştur. Redux Toolkit entegrasyonu durum yönetimini kararlı hale getirirken, robust LocalStorage yedekleme sistemi sayesinde kesintisiz veri hizmeti sağlanmıştır. Proje, rol bazlı erişim denetimi (RBAC) ile donatılmış, genişletilebilir ve güvenli bir meteoroloji tahmin platformu prototipi olarak başarıyla test edilmiştir.

5. KAYNAKÇA

- [1] React.js Resmi Dokümantasyonu ve State Yönetim Kılavuzları - <https://react.dev/>
- [2] Redux Toolkit Store Konfigürasyonu ve Slice Kılavuzu - <https://redux-toolkit.js.org/>
- [3] Bootstrap 5 Responsive Tasarım ve Arayüz Grid Sistemleri - <https://getbootstrap.com/>
- [4] Open-Meteo Canlı Meteoroloji ve Coğrafi Kodlama API Dokümanları - <https://open-meteo.com/>
- [5] JSON Server Mock REST API Sunucu Kurulum Dokümanı - <https://github.com/typicode/json-server/>