

Proje Özeti

MeteorViz, Dünya'ya yaklaşan gök cisimlerinin (asteroid, meteor) potansiyel etkilerini görselleştirmek ve simüle etmek için geliştirilmiş bir web tabanlı uygulamadır. NASA ve diğer açık veri kaynaklarından alınan gerçek zamanlı yörünge verilerini kullanarak çarpma senaryolarını, enerji hesaplamalarını, çukur boyutlarını, tsunami ve sismik etkileri interaktif şekilde kullanıcıya sunar.

Amaç

- **Kamuya ve araştırmacılara** görsel ve analitik bir araç sunmak.
- Asteroid çarpması gibi **kritik senaryoları** herkesin kolayca anlamasını sağlamak.
- **Risk analizi ve mitigasyon (sapma)** senaryolarını test edebilmek.
- NASA Space Apps gibi yarışmalarda **inovatif, açık kaynaklı** bir çözüm üretmek.

Hedefler

1. **Gerçek Zamanlı Veri:** NASA NeoWs, USGS ve NOAA API'lerinden verileri çekmek.
2. **Simülasyon Motoru:**
 - Yörünge tahmini
 - Etki enerjisi ve çukur boyutu hesaplama
 - Tsunami ve sismik etkilerin basit modelleri
3. **Mitigasyon Senaryoları:** Δv uygulayarak yörünge sapması simülasyonu.
4. **Web Arayüzü:**
 - Harita tabanlı görselleştirme (çarpma noktaları, etkiler)
 - Grafikler, istatistikler ve raporlar.
5. **Açık Kaynak Kod:** Kodun GitHub'da paylaşılması.

Kullanılacak Teknolojiler

Backend

- **Python + Flask / FastAPI** → API ve iş mantığı
- **Requests** → API entegrasyonu

- **Numpy, Scipy** → Hesaplamalar
- **Pandas / GeoPandas** → Veri işleme
- **Poliastro** (opsiyonel) → Yörünge simülasyonları

Frontend

- **React.js** veya **Next.js** → Modern web arayüzü
- **Leaflet.js** veya **Mapbox** → Harita görselleştirme
- **Chart.js / D3.js** → Grafikler

Veri Kaynakları

- **NASA NeoWs API** → Asteroit verileri
- **USGS / NOAA** → Topografya, tsunami ve deprem verileri

MeteorViz'de olacak özellikler net biçimde:

1. Veri Toplama ve İşleme

- NASA NeoWs API'den asteroid verileri
- Asteroid boyutu, hızı, kütlesi, çarpma olasılığı verileri
- Tarih filtreleme ve arşiv verisi çekme

2. Çarpma Simülasyonu

- Yörünge tahmini
- Çarpma enerjisi, krater çapı, etki yarıçapı hesaplama
- Tsunami ve deprem etkisi için basit modeller

3. Mitigasyon Senaryoları

- Δv (hız değişimi) ile yörünge sapması denemeleri
- Farklı sapma zamanları ve etkileri görselleştirme

4. Görselleştirme

- Harita tabanlı çarpma noktası ve etki alanı gösterimi
- Etki büyüklüğü, risk seviyesi, potansiyel zarar grafikleri
- Zaman çizelgesi: asteroid yaklaşımı → çarpma → sonrası

5. Raporlama

- Simülasyon sonuçlarını tablo ve grafik formatında alma
- PDF/JSON indirme seçeneği

6. Kullanıcı Arayüzü

- Basit senaryo oluşturma paneli
- Tarih seçimi, asteroid seçimi, sapma parametresi girme
- Anlık görselleştirme güncellemeleri

7. Çok Kullanıcılı Sistem (opsiyonel)

- Farklı kullanıcıların kendi senaryolarını kaydetmesi
- Paylaşılabilir senaryo linkleri

Görev Dağılımı

Backend

Amaç: API, veri işleme, simülasyon ve raporlama

Görevler

1. Proje Altyapısı

- Flask veya FastAPI kurulum
- Temel proje dizin yapısı

2. Veri Toplama Modülü

- NASA NeoWs API'den veri çekme
- JSON → Pandas/GeoPandas veri işleme

3. Simülasyon Motoru

- Çarpma enerjisi, krater boyutu hesaplama
- Δv uygulama ve yeni yörünge hesaplama

4. API Geliştirme

- /asteroids → asteroid verileri
- /simulate → çarpma simülasyonu
- /mitigation → sapma senaryoları

- /report → PDF/JSON sonuç çıktısı

5. Veri Depolama (Opsiyonel)

- SQLite veya PostgreSQL basit entegrasyonu

6. Dokümantasyon

- API endpoint açıklamaları
 - Hesaplama formülleri
-

Frontend

Amaç: Görselleştirme, arayüz ve kullanıcı etkileşimi

Görevler

1. Proje Altyapısı

- HTML
- Tailwind CSS ile basit stil

2. Arayüz Tasarımı

- Anasayfa
- Senaryo oluşturma paneli
- Harita ve grafik ekranları

3. Veri Görselleştirme

- Leaflet.js / Mapbox → Harita üzerinde etki alanı
- Chart.js / D3.js → Enerji, etki büyüklüğü grafikleri

4. Backend Entegrasyonu

- API çağrıları (axios/fetch)
- Simülasyon ve rapor verilerini çekme

5. Sonuç Raporlama

- PDF/JSON indirme butonları
- Simülasyon sonuçlarını ekranda gösterme

6. Kullanıcı Deneyimi

- Yükleniyor animasyonları

- Hata mesajları ve doğrulama