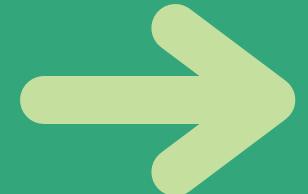


MUHAMMET EMİN HOLOĞLU

İSTANBUL TRAFİK VERİLERİNİN GÖRSELLEŞTİRİLMESİ VE ROTA ÜSTÜ SEÇİMLERLE AĞ YAPISINIŃ OLUŞTURULMASI



GENEL BAKIŞ



**Projem web tabanlı bir yapıdan oluşmaktadır.
Kullanıcının başlangıç ve bitiş noktasını
seçtiği senaryoda bir rota oluşturulur. Yine
kullanıcıdan rota üstü seçimler yapması
beklenir. Bu seçimler rotanın etrafında bir
ağ yapısı şeklinde haritada görüntülenir**



KULLANDIĞIM TEKNOLOJİLER



Python
Backend



Flask
Backend



JavaScript
Backend



HTML
Frontend



CSS
Frontend



Leaflet
Harita&Rota



PapaParse
Veri Önisleme



Google Maps
Places API

İBB VERİ SETİ



İspark konumlarını İBB veri seti havuzundan aldım. Örnek:

İSPARK Lokasyonları

Veri Adresini Kopyala İndir Veri API

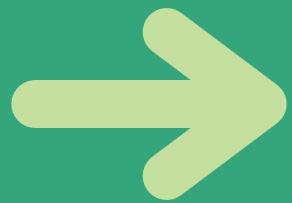
İSPARK tarafından sağlanan parklanma alanlarına ait bilgileri içeren veri setidir. İşbu veri kaynağının içerisinde, Park Adı, Lokasyon Adı, Park Tip Id, Park Tip Tanım, Park Kapasitesi, Çalışma Saatleri, İlçe Adı, Boylam ve Enlem bilgilerini içerir.

Veri Kasifi

Filtre ekle Tam ekran Göm

Tablo Grafik Harita 708 kayıt « 1 » Filtreler Veriyi ara ... Git »

_id	PARK_NAME	LOCATI...	PARK_T...	PARK_T...	CAPACI...	WORKI...	COUNT...	LONGIT...	LATITUDE
1	Vali Konağı Cadd...	Şişli Vali ...	YOL ÜSTÜ	YOL ÜSTÜ	30	08:00-19...	ŞİŞLİ	28.9879...	41.0488...
2	Şakayık Sokak 1	Şişli Şak...	YOL ÜSTÜ	YOL ÜSTÜ	50	08:00-19...	ŞİŞLİ	28.9955...	41.0495...
3	Hüsrev Gerede S...	Şişli Hüs...	YOL ÜSTÜ	YOL ÜSTÜ	50	08:00-19...	ŞİŞLİ	28.9949...	41.0483...
4	Çamlıca Mahalle...	Şişli Çam...	YOL ÜSTÜ	YOL ÜSTÜ	50	08:00-19...	ÇAMLİCA	28.9950...	41.0500...



DİNAMİK VERİ SETİ

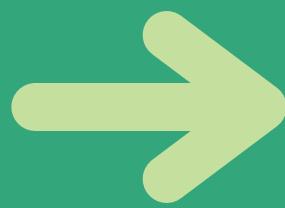
Aslında her soru için kendini
değiştiren ve yenileyen
dinamik veri setlerine sahibim.

```
name,latitude,longitude
Kömürde Kokoreç Hüseyin Usta'nın Yeri,41.03670049999999,28.871571
NE KOKOREÇ & MİDYE,41.040929,28.912444
Yasko Kokoreç,41.0590218,28.9018355
Class Kokoreç,41.0301488,28.8633295
ECEM KOKOREÇ,41.0611131,28.9281486
Öz Şampiyon Kokoreç,40.994295,28.902002
Güneş Kokoreç,41.0093991,28.8675518
Gala Kokoreç,41.0156497,28.9299266
Deniz Kokoreç,41.0446778,28.8599311
Paşa Kokoreç,41.0086063,28.9311422
Adios Kokoreç,41.0143044,28.8626208
ALİ USTA MİDYE KOKOREÇ,41.002302,28.8600963
KÖMÜRDE KOKOREÇ HAMDİ USTA,41.0435733,28.8595883
GENCO PİLAV KOKOREÇ,41.0266988,28.878998
Tikinak Kokoreç,41.0401221,28.9058881
ÇINAR KOKOREÇ-UYKULUK-IZGARA KÖFTE,41.0581464,28.9264129
Mis Kokoreç,41.0248023,28.9399675
Gala Kokoreç Bayrampaşa,41.03722399999999,28.912477
Rüya Kokoreç,41.0626058,28.8974262
Kokoreççi Tekin Usta,40.9927511,28.7862805
Kokoreççi Şeref Usta,41.0257412,28.8069241
Gala Kokoreç İkitelli Atatürk Mahallesi,41.0539035,28.7949032
```

GOOGLE MAPS API

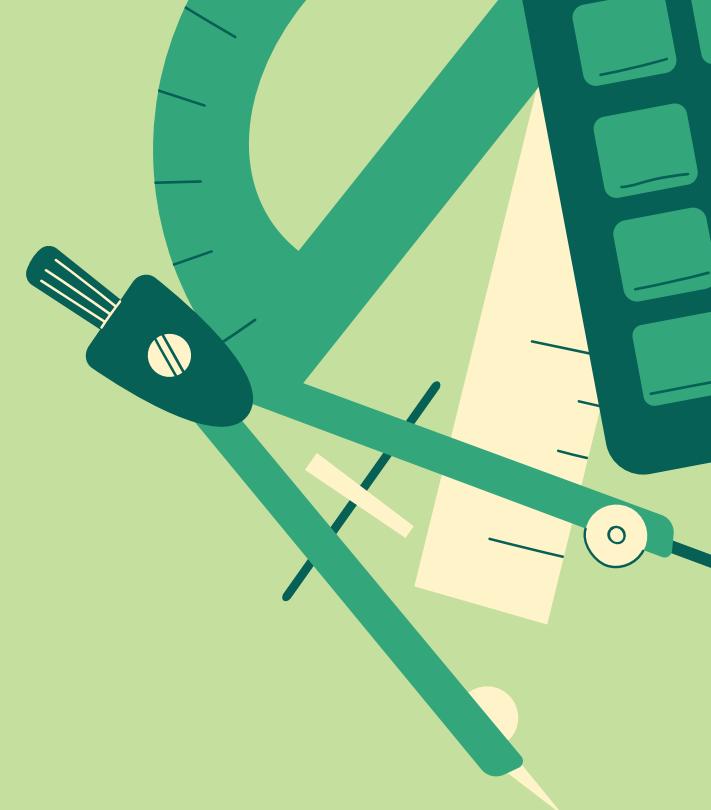
```
def find_nearest_gas_stations(midpoint, keywords, radius=2000):
    # Google Places API ile en yakın akaryakıt istasyonlarını bulan fonksiyon
    endpoint = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"
    gas_stations = []
    for keyword in keywords:
        params = {
            'key': google_api_key,
            'location': f"{midpoint[0]},{midpoint[1]}",
            'radius': radius, # Search within the specified radius
            'keyword': keyword
        }
        response = requests.get(endpoint, params=params)
        data = response.json()
        if 'results' in data:
            for result in data['results']:
                location = result['geometry']['location']
                name = result['name']
                place = {
                    'name': name,
                    'latitude': location['lat'],
                    'longitude': location['lng']
                }
                gas_stations.append(place)
    return gas_stations
```



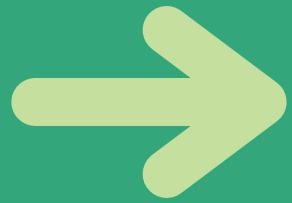


3 NOKTA ARAMA

Arama ve konum bulma performansını artırmak için 3 nokta arama yöntemi kullanılmıştır. Rotanın başlangıç, orta ve bitiş noktaları referans alınarak 3 nokta için arama yapılarak rota üstü seçim olanakları artırılmıştır.



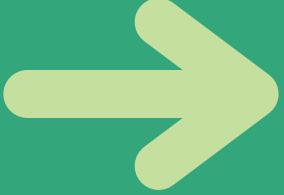
```
if include_gas_stations:  
    # En yakın akaryakıt istasyonlarını bul: orta noktası, başlangıç ve bitiş  
    nearest_gas_stations_midpoint = find_nearest_gas_stations(midpoint, keywords)  
    nearest_gas_stations_point1 = find_nearest_gas_stations(point1, keywords)  
    nearest_gas_stations_point2 = find_nearest_gas_stations(point2, keywords)  
  
    # Sonuçları birleştir  
    nearest_gas_stations = nearest_gas_stations_midpoint + nearest_gas_stations_point1 + nearest_gas_stations_point2  
    # Tekrarları sil  
    unique_gas_stations = {station['name']: station for station in nearest_gas_stations}.values()  
  
    # CSV'ye kaydet  
    save_to_csv(unique_gas_stations)  
else:  
    clear_csv('static/gas_stations.csv')
```



Leaflet Routing Machine

Leaflet Routing Machine (LRM), Leaflet harita kütüphanesi için bir yönlendirme (routing) eklentisidir. Bu eklenti, kullanıcıların harita üzerinde iki veya daha fazla nokta arasında rota oluşturmasını ve yön tariflerini görmesini sağlar. İşte Leaflet Routing Machine'in bazı özellikleri:

- 1. Rota Hesaplama:** Kullanıcıların başlangıç ve varış noktalarını belirlemesine ve bu noktalar arasında en iyi rotayı hesaplamasına olanak tanır.
- 2. Desteklenen Yönlendirme Hizmetleri:** OpenStreetMap (OSM) tabanlı yönlendirme servisleri (örneğin OSRM, Mapbox Directions, GraphHopper) ile çalışabilir.
- 3. Yol Tarifi:** Harita üzerinde adım adım yol tarifi sağlar. Bu tarifler yazılı olarak görüntülenebilir ve harita üzerinde işaretlenebilir.
- 4. Çoklu Nokta Desteği:** Birden fazla ara nokta ekleyerek karmaşık rotalar oluşturulabilir.
- 5. Kullanıcı Arayüzü:** Kullanıcı dostu bir arayüz sağlar. Sürükle ve bırak özelliği ile başlangıç ve varış noktaları kolayca ayarlanabilir.
- 6. Eklenti Uyumluluğu:** Leaflet'in diğer eklentileri ile uyumlu çalışabilir, bu da daha kapsamlı ve özel harita uygulamaları geliştirilmesini sağlar.



```
function drawLinesToGasStations(gasStations) {
    gasStations.forEach(function (gasStation) {
        var coordinates = [parseFloat(gasStation.latitude), parseFloat(gasStation.longitude)];
        if (isNaN(coordinates[0]) || isNaN(coordinates[1])) {
            console.error("Geçersiz koordinatlar:", gasStation);
            return;
        }
        var marker = L.marker(coordinates, { icon: gasStationIcon }).addTo(leafletMap).bindPopup(gasStation.name);
        L.Routing.control({
            waypoints: [
                L.latLng(point1),
                L.latLng(coordinates)
            ],
            lineOptions: {
                styles: [altRouteStyle]
            },
            routeWhileDragging: false,
            show: false,
            addWaypoints: false,
            createMarker: function () { return null; }
        }).on('routesfound', function (e) {
            var routes = e.routes;
            var summary = routes[0].summary;
            console.log('Benzin istasyonu rotası bulundu:', summary);
        }).on('routingerror', function (e) {
            console.error('Benzin istasyonu rotası bulunamadı:', e.error);
        }).addTo(leafletMap);
    });
}
```

HAVERSINE FORMÜLÜ

```
function getDistance(point1, point2) {  
    var lat1 = point1[0]; // İlk noktanın enlemi  
    var lon1 = point1[1]; // İlk noktanın boylamı  
    var lat2 = point2[0]; // İkinci noktanın enlemi  
    var lon2 = point2[1]; // İkinci noktanın boylamı  
    var R = 6371; // Dünya'nın yarıçapı (km)  
  
    // Enlem farkını radyan cinsine çevir  
    var dLat = (lat2 - lat1) * Math.PI / 180;  
  
    // Boylam farkını radyan cinsine çevir  
    var dLon = (lon2 - lon1) * Math.PI / 180;  
  
    // Haversine formülünün hesaplanması  
    var a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +  
        Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 * Math.PI / 180) *  
        Math.sin(dLon / 2) * Math.sin(dLon / 2);  
  
    // Büyük çemberin merkez açısı  
    var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));  
  
    // Mesafeyi hesapla  
    var distance = R * c;  
  
    return distance; // Hesaplanan mesafeyi döndür  
}
```

AĞ YAPISI

İSPARK

AKARYAKIT İSTASYONLARI

RESTORANLAR

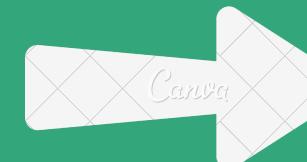
HASTANELER

SPESİFİK YERLER

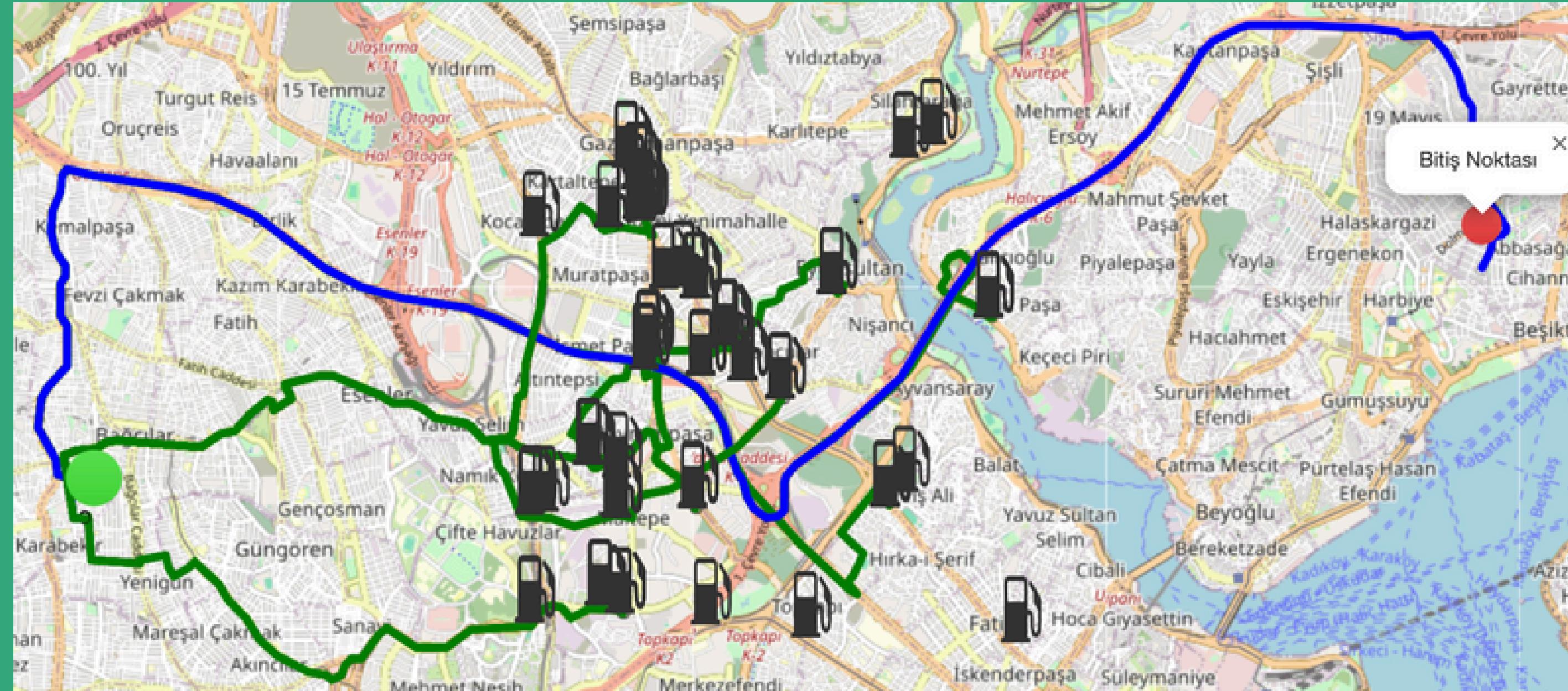


DÜĞÜM

ROTALAR



KENAR



Kullanım

1

Kullanıcı rotanın başlangıç ve bitiş noktalarını seçer.

2

Rota üstü seçimler işaretlenir veya metin olarak girilir.

3

Haritada başlangıç ve bitiş arasına mavi renkte rota çizilir.

4

Rotanın etrafında seçilen rota üstü mekanlar ağ yapısı oluşturur ve alternatif rotalar çizilir.

5

Ekranın sağ tarafında yol tarifi gösterilir.

İstanbul canlı trafik yol durumu renklendirilir, rota tahmini süresi hesaplanır.



Senaryolar



1

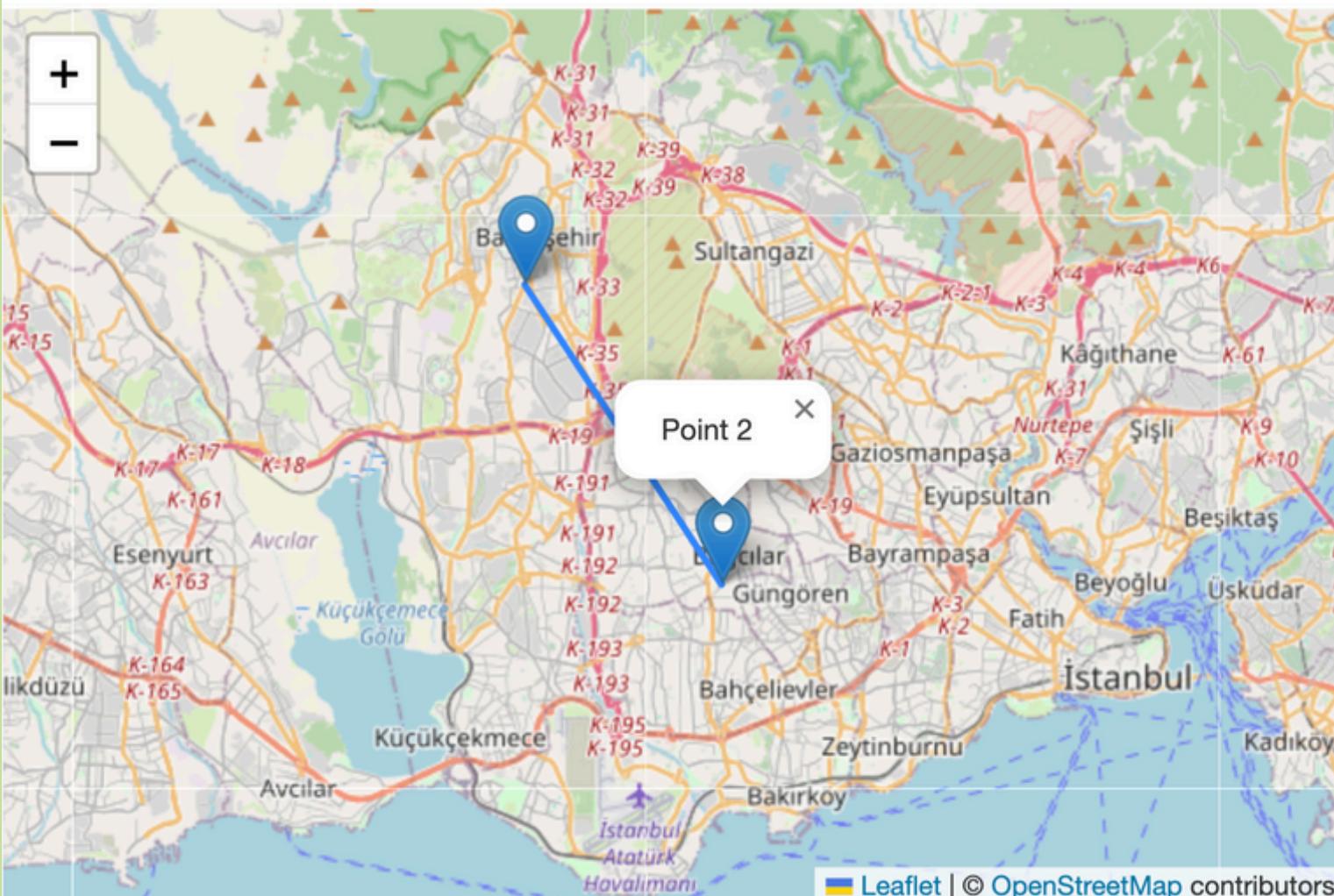
İki Nokta Seç

1. Nokta:

41.09623344872029,28.79104614257813

2. Nokta:

41.026857136502365,28.85147094726563



Senaryolar



2

Rota Üstü Seçim Yap

- İSPARK
- Benzin İstasyonu
- Restoran
- Hastane

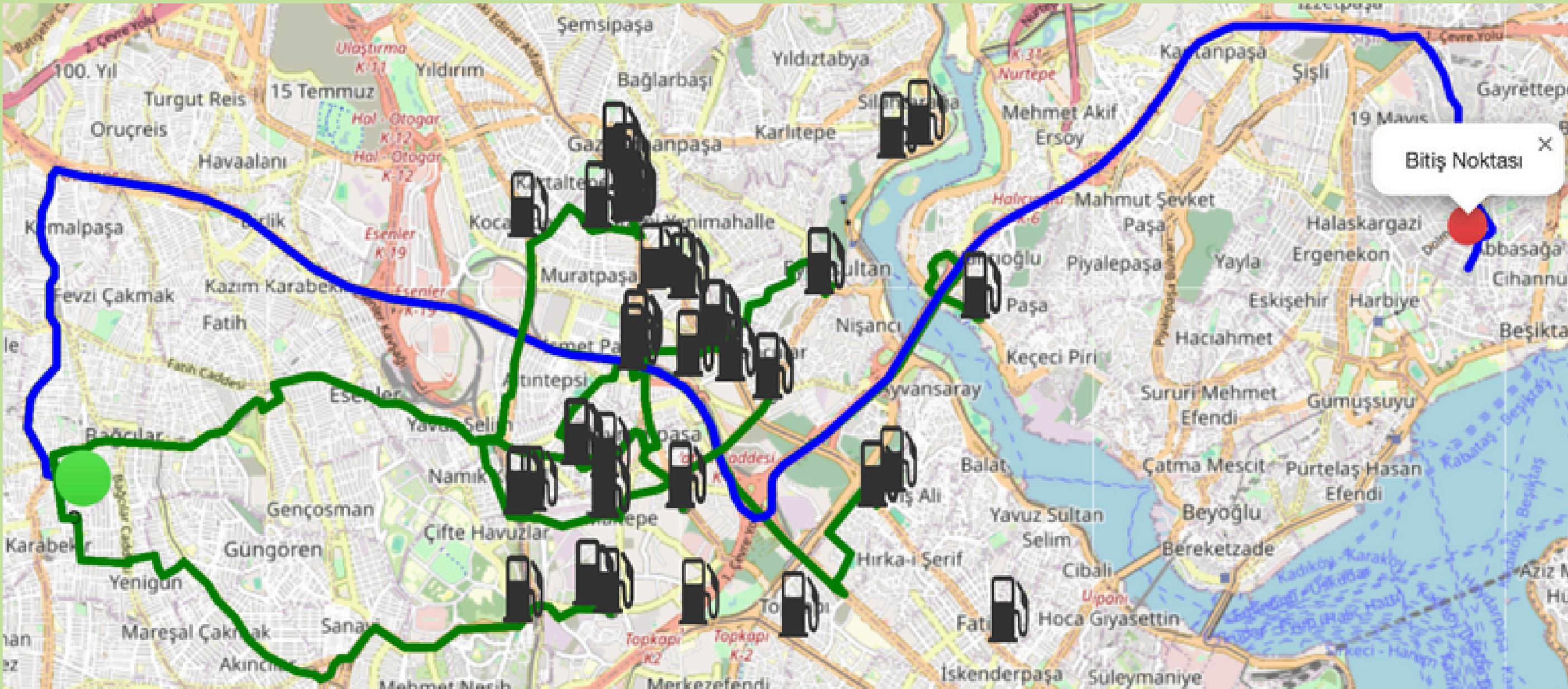
Farklı bir yer girin:

Haritayı Oluştur

Senaryolar



3



Senaryolar



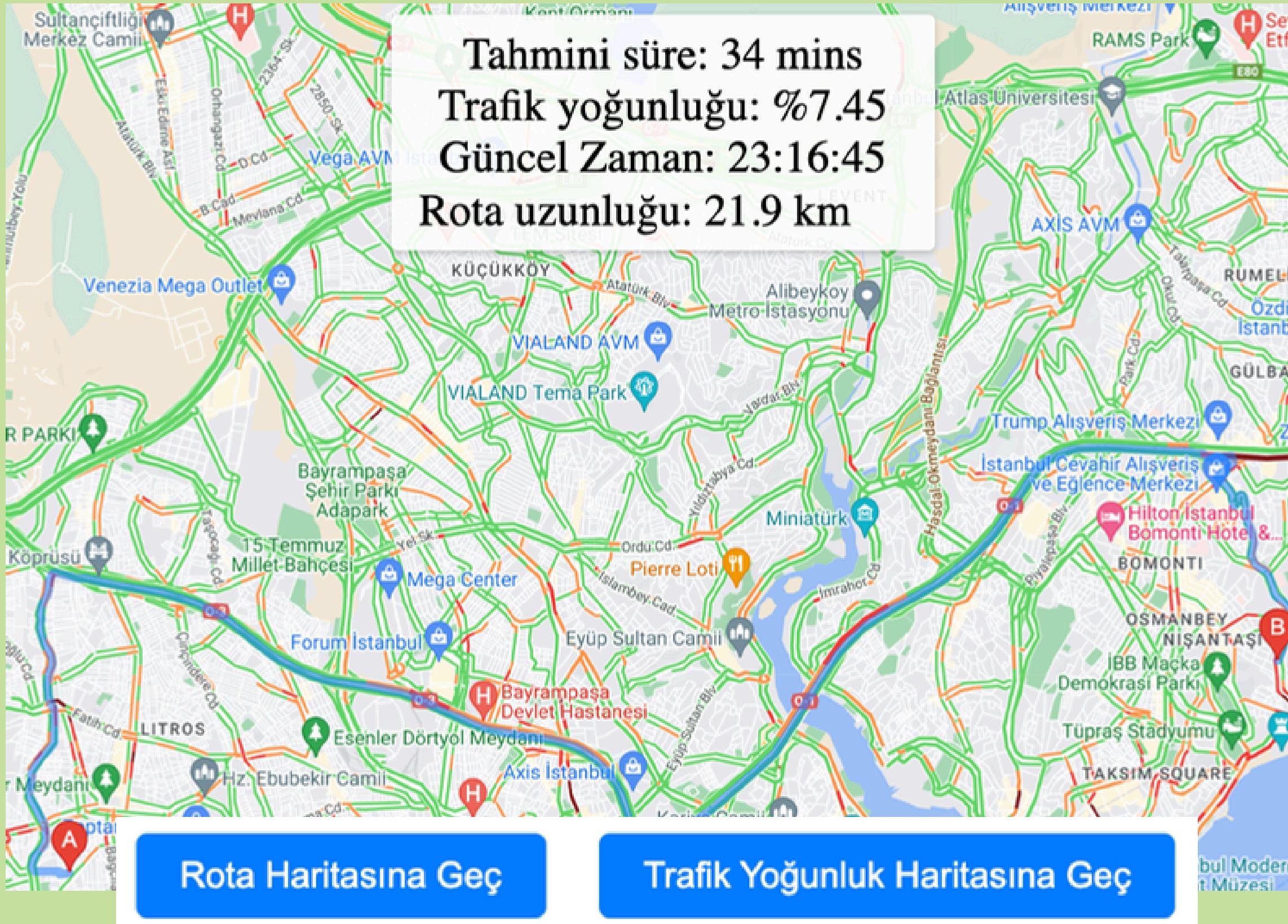
4

O-3, O-3	
19.2 km, 22 min	
Head west	250 m
Turn right onto Birlik Caddesi	250 m
Exit the traffic circle onto Birlik Caddesi	700 m
Make a slight left onto Birlik Caddesi	150 m
Turn right onto Mehmet Akif Bulvarı	1.5 km
Go straight onto O-3 Güney Yanyolu	200 m
Take the ramp on the left towards Ankara	150 m
Merge left towards O-3: Aksaray	6 km
Take exit K-3 towards Okmeydanı	250 m
Keep left towards Okmeydanı	900 m
Merge left towards Ankara	900 m
Keep left towards O-2: Çağlayan	2.5 km
Take exit K-7 towards O-2: Kağıthane	350 m
Keep right towards Okmeydanı	350 m

Senaryolar



5



TEŞEKKÜRLER. ^

MUHAMMET EMİN HOLOĞLU

