

# Kargo Dağıtım Sistemi

Buğrahan TOPAL 190201050, Bahadır IŞIK 190201049

## 1. ÖZET

Google Map API üzerinden manuel yada harita üstünden işaretlenen kargo noktalarına en kısa yol üstünden giden kargocu uygulaması.

## 2. GİRİŞ

Bu projedeki ilk amacımız Google Map API'yi kodumuza entegre etmek oldu. Daha sonrasında girilen noktalar kod içine aktarılmıştır.

Kod içine aktarılan noktalar sayesinde kargo konumlarımızın locasyonu belli olur . Belli olan locasyonları Google map API sayesinde birbirlerine olan uzaklıklar hesaplanır bu uzaklıklar Floyd – Warshall matrisine aktarılır ve en kısa yol bulma algoritması çalışır. Daha sonra en kısa yol çizdirilir.

## 3. YÖNTEM

### A) Floyd-Warshall :

Floyd-warshall algoritması bize birbirine olan uzaklıkları bilinen konumların en kısa yolunu bize sunar.

### B) Google Map API:

Google Map API bize görsel arayüz ve bu arayüz sayesinde götürülecek kargoların konumlarını alma bu konumları yol olarak çizdirme yöntemlerini sağlar.

### C) Electronjs:

Yazdığımız Javascript kodlarını masaüstü uygulamasına çevirir.

### D) Mongodb:

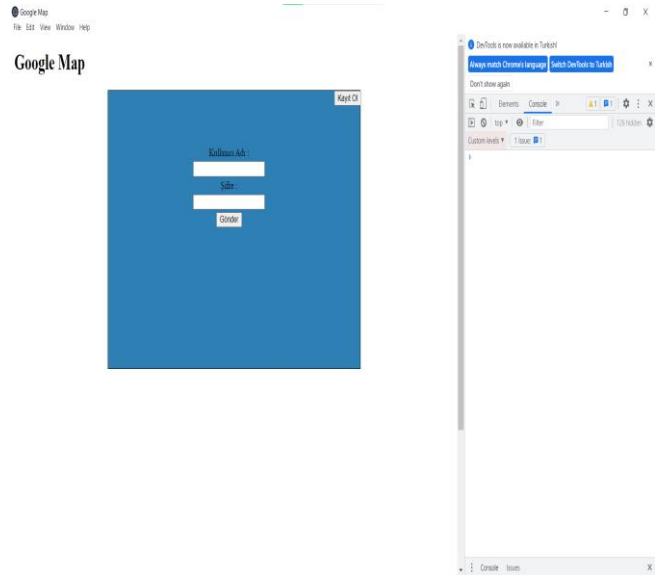
Kullanıcı adı ,Parola gibi gereksinimleri bulut üzerine kaydetmemizi sağlar.

### E) Thread Yöntemi:

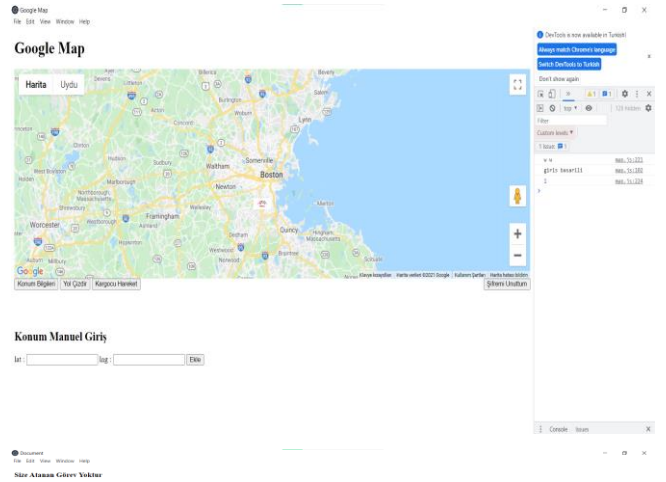
Yaptığımız uygulamanın ikinci ekrana entegre olmasını sağlıyor.

## 4. DENEYSEL SONUÇLAR

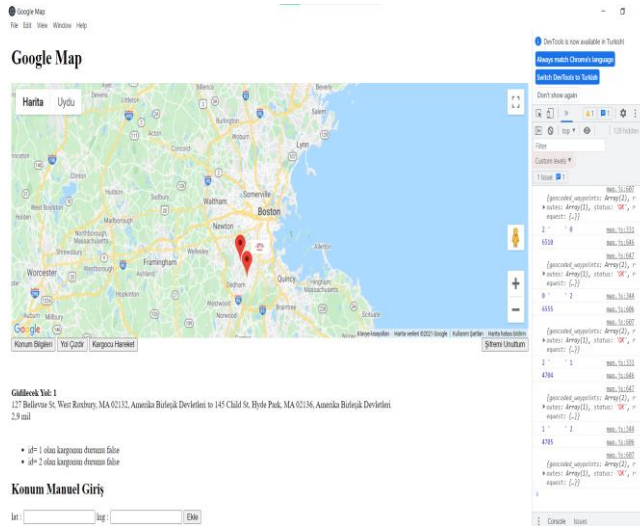
### A) Ana Giriş Ekranı :



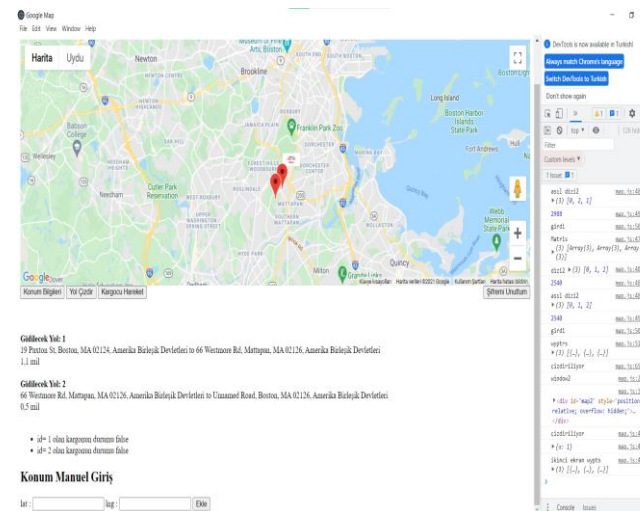
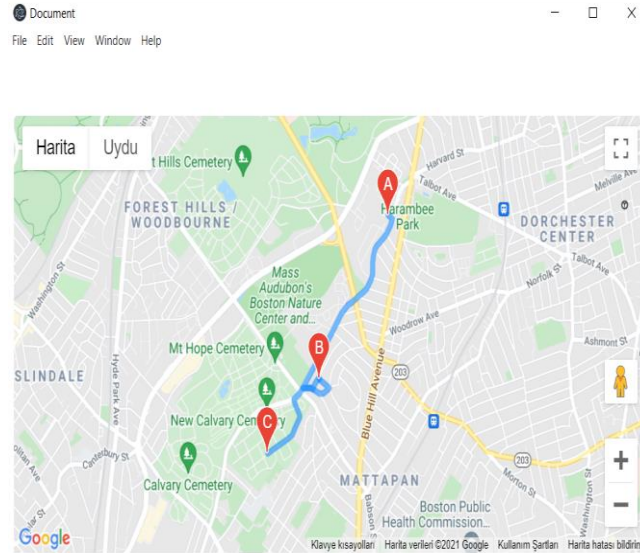
### B) Giriş Sonrası Görünen Ekranlar:



## C) Konum Alması:



## D) En Kısa Yolu Çizdirme:



## 5. YALANCI KOD

- Başla
- Google Map API ekle.
- Marker oluşturun.
- Oluşan markera başlangıç konumu ekle.
- AddMarker adlı fonksiyonu oluşturun
- Bu fonksiyon ile ekran üstünden basılan konumları al.
- Alınan konumları başlangıç noktası ile bir dizi içine at.
- Konum al butonu için fonksiyon yaz.
- Google Map API'nin iki konum arasındaki uzaklık bulma fonksiyonunu yaz.
- Butona basıldığında uzaklıklar arasındaki konumları bul
- Bu fonksiyon ile birilen tüm noktaların birbirine olan uzaklığını bul.
- Bu uzaklıkları bir dizinin içinde tut.
- Yolu çizdir butonun fonksiyonunu yaz.
- Uzaklık dizisini Floyd-Warshall matrisinin içine at.
- Matrisi 00 11 22 vb noktalarını 0 yap.
- Daha sonra nokta 1in nokta 2 ye olan uzaklığını matris [1][2] de tut.
- Faktöriyel hesabı yapmak için bir adet factorialize fonksiyonunu yaz.
- Bu fonksiyon ile faktöriyel hesabı yap. (nokta sayısı kadar)
- Nokta sayısı 2 den küçük ve 2 ye eşit ise wypts dizisini oluşturun ve dizinin ilk iki elemanını ata.
- Nokta sayısı 2 den çok ise faktöriyel hesabı sayesinde en kısa yolu bul.
- Yol çizdirme fonksiyonunu yaz.
- Yol çizdir butonun en altında bu fonksiyonu kullan ve yolu çizdir.
- Html dosyası sayesinde arayüzü oluşturun.
- Kullanıcı adı ve parolası iste.
- Kullanıcının girişi yok ise kayıt butonu ekle.
- Kayıt olmak isteyen kullanıcının girdiği kullanıcı adın ve parolayı al.

- Database\_kayıt adlı fonksiyonu yaz.
- Bu fonksiyon sayesinde kullanıcıdan gelen veriyi bulut sistemine ekle.
- Daha sonra kayıt yerinden çıkış yapan kullanıcının giriş ekranına geri döndür.
- Kullanıcı kullanıcı adı ve şifresini giriş ekranına yazsın.
- Girilen verileri giriş butonu ile al.
- Alınan verileri bulut sistemi üzerinde kontrol et ve eşleşen veri var ise girişini sağla.
- Şifresini değiştirmek isteyen kullanıcılar için şifre değiştir butonu ekle.
- Bu butona basan kullanıcıyı şifre değiştir ekranına götür.
- Şifresini değiştirmek isteyen kullanıcıdan kullanıcı adını ve şifresini al daha sonra yeni şifresini al.
- Ve bulut sistemi üzerinden şifresini değiştir.
- Giriş ekranından giren kullanıcının gördüğü ekrana manuel konum ekleyen bir yapı kur.
- Bu yapıda girilen konum bilgisini al.
- Bu yapıyı yanında bulunan buton sayesinde konumunu ekranda göster.
- Kargocu gittiği adreslere kargo bıraktığında kargonun durumu teslim edildi olacak şekilde html üzerinden bir yapı kur.
- Kargocu her hareket ettiğinde bu yapıyı değiştir.
- Kargocu hareket et fonksiyonu oluştur.
- Bu fonksiyon kargocuyu hareket ettirir böylece kargocu kaldığı konumdan en kısa yolu bularak diğer konumlara gider.
- İkinci bir ekran oluştur.
- Thread mantığı ile iki ekran arasındaki koordinasyonu sağla.

## 8.SONUÇ

Bu proje sayesinde API kullanmayı öğrendik. Thread mantığının nasıl olması gerektiğini öğrendik. Elektronjsin nasıl kullanılması gerektiğini öğrendik. Javascripti öğrendik.

## 9. KAYNAKÇA

- <https://www.mongodb.com>
- <https://www.youtube.com/watch?v=Zxf1mnP5zcu>
- [https://www.youtube.com/watch?v=qsM9ylyt7xw&list=PL\\_f2F0Oyaj48jgl98pHuoyxoTgi7gXJ-z](https://www.youtube.com/watch?v=qsM9ylyt7xw&list=PL_f2F0Oyaj48jgl98pHuoyxoTgi7gXJ-z)
- [https://developers.google.com/maps/documentation/javascript/directions#maps\\_directions\\_waypoint\\_s-javascript](https://developers.google.com/maps/documentation/javascript/directions#maps_directions_waypoint_s-javascript)
- <https://www.youtube.com/playlist?list=PLtWy6rdN1yBntIyLeNo-fCVR6AaK7vdyI>
- <https://app.patika.dev/moduller/javascript>
- <https://www.youtube.com/watch?v=8NiYo5Ise-k&t=440s>