

KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
BÖLÜMÜ
YAZILIM LABORATUVARI II.
Proje 1

190202078-HAMİT DOĞAN

190202066-HÜSEYİN CAN TAŞDEMİR

Projenin Amacı:

Porjenin amacı araçlarını takip etmek isteyen bir A firması vardır. Biz A firmasının teknik çözüm ortağı B firmasını temsil ediyoruz. A firmasının ihtiyaçları kapsamında, ilgili sistem çözümünü, temel ihtiyaçlar ve tasarım kısıtları kapsamında oluşturmakla ve demosunu yapmamız amaçlanmıştır.

Programlama Dili:

Programımızı Visual Studio Code da nodejs kullanarak ve Database kısmını ise NOSQL için redisi, Normal SQL için ise MYSQL yapısını kullandık.

Problem Tanımı:

Bu projede bizlerden çok katmanlı bir web uygulaması geliştirilmesi istenmektedir. Model, Manzara/Arayüz ve Yönetim (Model,View,Controller) kalıbının kullanılması (Arayüz, İşlem Mantığı ve Verinin ayrıştırılması), mesaj simsarı kullanarak yazılım fonksiyonel bölümleri arasındaki bağımlılıkların azaltılması, dağıtık bir sistem mimari kullanarak çözüm oluşturma ve değişik tipte veri tabanlarını kullanma ve veri tabanı tasarımı kullanmamız istenilmektedir. Araç takip hizmeti vermek isteyen bir A firması, müşterilerine araçlarına takabilecekleri GPS destekli bir takip donanımı ve ilgili hizmetleri ücret karşılığı sunmak istemektedir. Alınan takip donanım üzerinde GSM modülü bulunmakta olup, araçların mevkileri GSM üzerinden, A firmasının sunucularına aktarılacaktır. A firması, müşterilerine web arayüzü ve mobil uygulamalar üzerinden, ödeme planına göre, değişik güncelleme seviyelerinde, araçların mevki bilgilerini sunacaktır. A firması ilgili donanımı hazır ticari ürün olarak temin edecektir. Ancak A firmasının dağıtık bir sistem kurma ve ilgili servisleri hazırlama ihtiyacı bulunmaktadır. A firmasının yerel bir teknik alt yapı işletmek istemediğinden oluşturulacak çözüm bulut hizmetleri üzerinde kullanılabilir olması gerekmektedir. Biz A firmasının teknik çözüm ortağı B firmasını temsil ediyorsunuz. A firmasının ihtiyaçları kapsamında, ilgili sistem çözümünü, temel ihtiyaçlar ve tasarım kısıtları kapsamında oluşturmakla ve demosunu yapmakla görevliyiz.

Yapılan Araştırmalar:

İlk olarak programalama dili olarak node.js kullanmaya karar verdik. NoSQL veri tabanı için birçok araştırma yaptık. Yaptığımız araştırmalar sonucunda REDİS i kullanmaya karar verdik. Bu noktadan sonra Windows kullandığımız için

redis bilgisayarıma direkt olarak kuramadık ve docker container yapısını kullandık. Docker ı bilgisayarıma kurduktan sonra docker içine redis indirdik kurduk. Birden fazla kaynaktan (araçlardan) akan GPS verileri alınarak bir veri tabanına depolanması istenmişti. Verilerin alınması için bir mesaj simsarcısı (message broker) olarak rabbitMQ kullandık. Daha sonra bu yapılar arasındaki bağlantıları sağlayan kod parçalarını yazdık. Harita olarak Leaflet yapısını kullanmaya karar verdik. Bununla birlikte kendi tasarladığımız bir login ekran yaptık. Son olarak da kullanıcıların bilgilerini tutmak amacıyla database olarak MYSQL yapısını kullandık.

Genel Yapı:

Projeye başlarken ilk olarak NOSQL yapısını araştırdık ve bunlardan bir tanesi olan Redis kullanmaya karar verdik. Daha sonra redis i araştırdık. Ben ve takım arkadaşım Windows kullandığımız için redis bilgisayarıma direkt olarak kuramadık ve bunun için öncelikle bilgisayarıma docker yapısını kurduk. Cmd komut ekranında docker ı çalıştırdıktan sonra redisin kurulumunu yaptık. Daha sonra rabbitmq yu erlang ile beraber bilgisayarıma kurduk. Visual Studio Code u indirip kurulumunu yaptık. Aynı şekilde nodejs i de indirip kurduktan sonra Visual Studio Code ortamında nodejs kodlamak için gerekli bağlantıları yaptık. Bu bağlantılardan sonra gerekli kütüphaneleri (redis,nodemon,Express gibi) terminal ekranından npm install komutu ile indirdik. Daha sonra Publisher clası ile veri gönderme olayını yapıp, Consumer clası ile de veri alma işlemini anlık olarak gerçekleştirdik. Proje pdfinde kullanmamız gereken taximovemeentconcatenated verilerini indirip, bu csv verilerinin ihtiyacımız olan kısmını alıp, conver to json ile stringe çevirip, date, latitude, longitude ve id formatında düzenleyip kodumuzda data.json clası oluşturup, o clasın içine yapıştırdık. Daha sonra Express yapısı(ajax ve app.get methodları) ile htmldeki verileri localhosta nasıl alacağımızı araştırıp bday:time yapısı ile localhosta data.json daki herhangi bir tarihi girdiğimizde terminale o verileri çektik. Bu veri ilk başta sadece tarih olarak geliyordu. Sonra bunu isterlerde olduğu gibi girilen tarihten önceki son yarım saat içerisindeki tarihleri getirecek şekilde ayarladık. Gerekli kodu yazıp tarih verisini girdikten sonra o tarihteki aracın id sini ve lokasyonlarını çektik. Ardından leaflet ile haritamızı oluşturup çektiğimiz verilerin lokasyonlarını gönderip markerlayarak haritamıza bastırdık. Geriye kalan login ekran kısmını da kendimiz dizayn ederek kullanıcı bilgilerini ve her kullanıcıya ait 2 tane carid oluşturarak projemizi tamamladık.

Yalancı Kod:

1.npm run -y ile package.json classını oluşturup gerekli kütüphaneler kuruldu.

2.Publisher class oluşturup connect_rabbitmq() yapısını kurduk.

3.Consumer class ile connect_rabbitmq() ve redis.createClient() ile redis bağlantısını yaptık.

3.1.client.set() ile Publisher dan gönderilen verileri aldık.

4.data.json clasının içine araç verilerini kopyaladık.

4.1. date:

Latitude:

Longitdue:

İd:

5.public adında bir folder oluşturup içine foto1.jpg, foto2.jpg, foto3.jpg, index.html, index.js, index2.html ve style.css oluşturduk.

5.1. foto1.jpg, foto2.jpg, foto3.jpg login ekran için tasarım resimleri.

5.2.index.html login ekranı için kullanıcı adı ve password kısmı butonları.

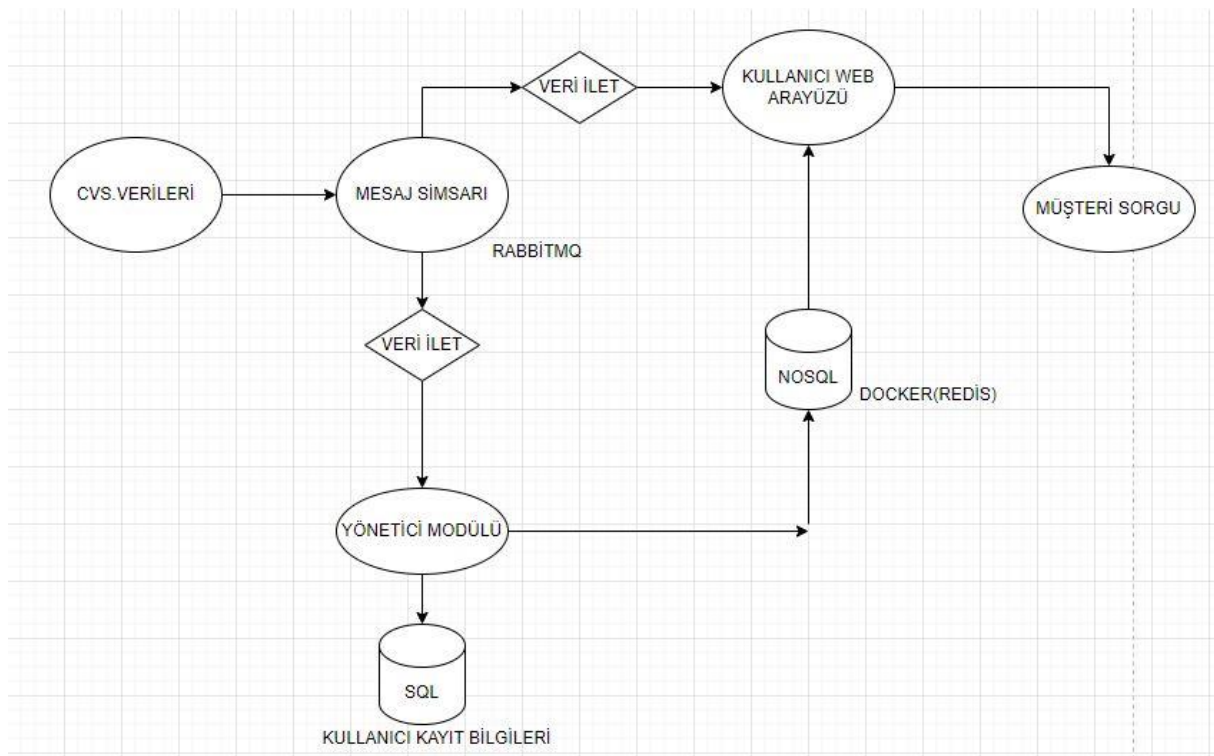
5.3.index.js harita için gerekli maxzoom,subdomains.

5.4.index2.html saat ve harita için gerekli butonlar, ajax ile post methodu için gönderilmesi gereken fonksiyonlar, marker fonksiyonu.

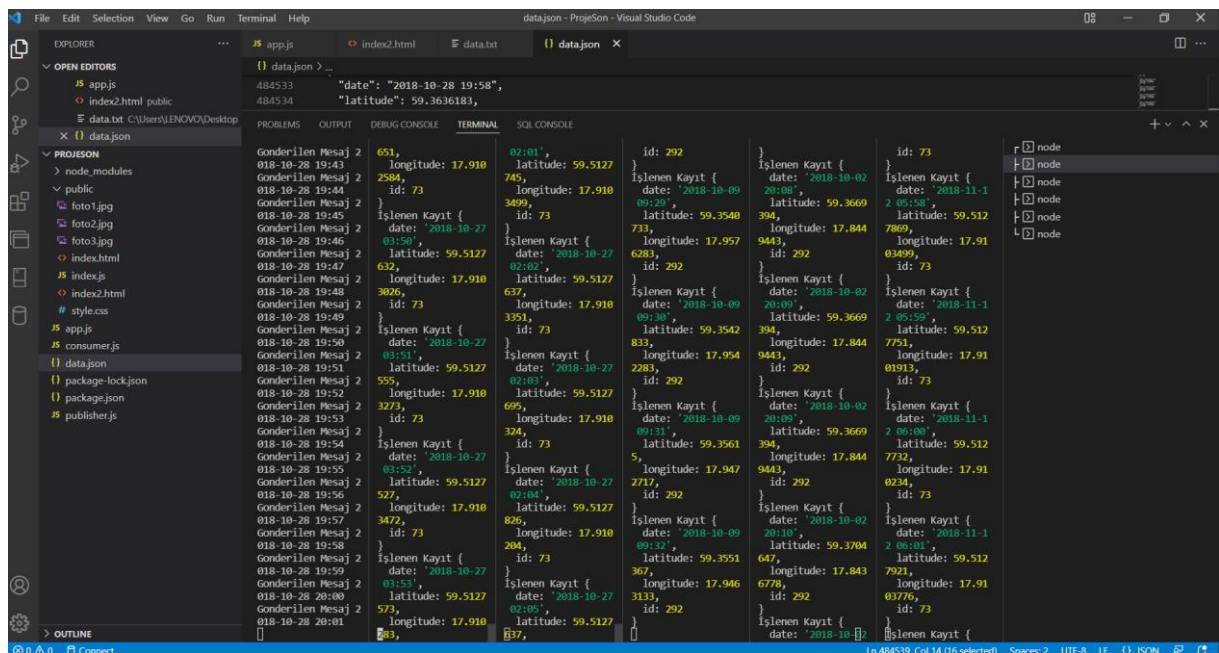
5.5. style.css arka plan , butonlar vs özellikler için width,height,font-size vs özellikler.

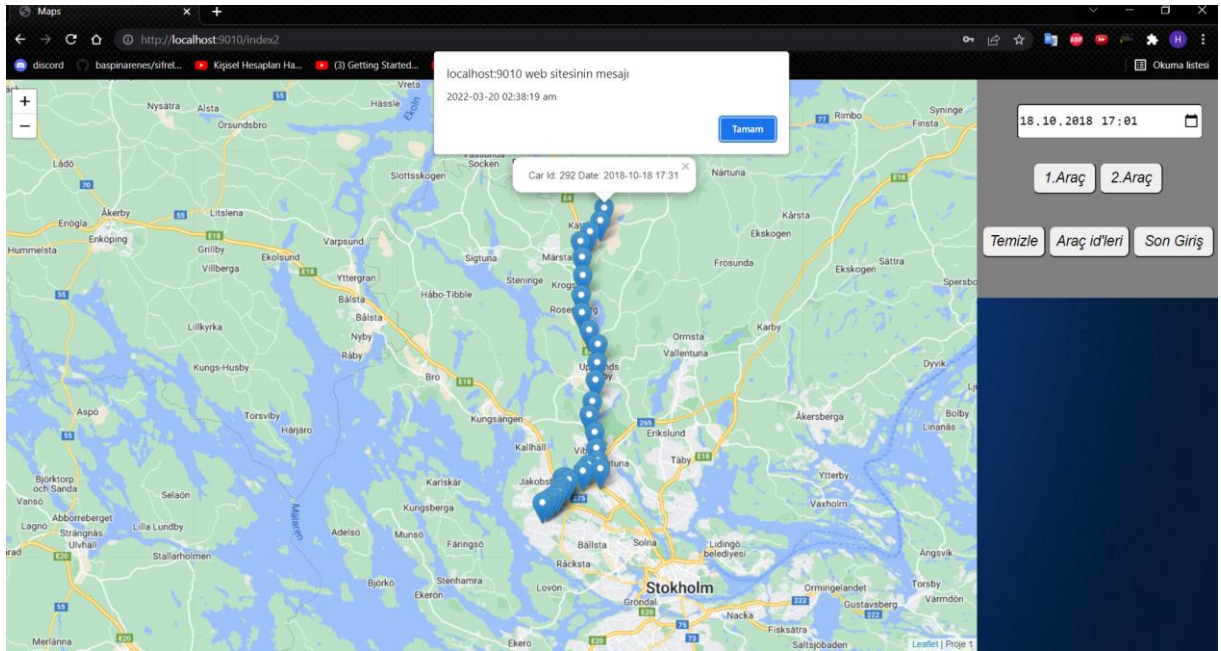
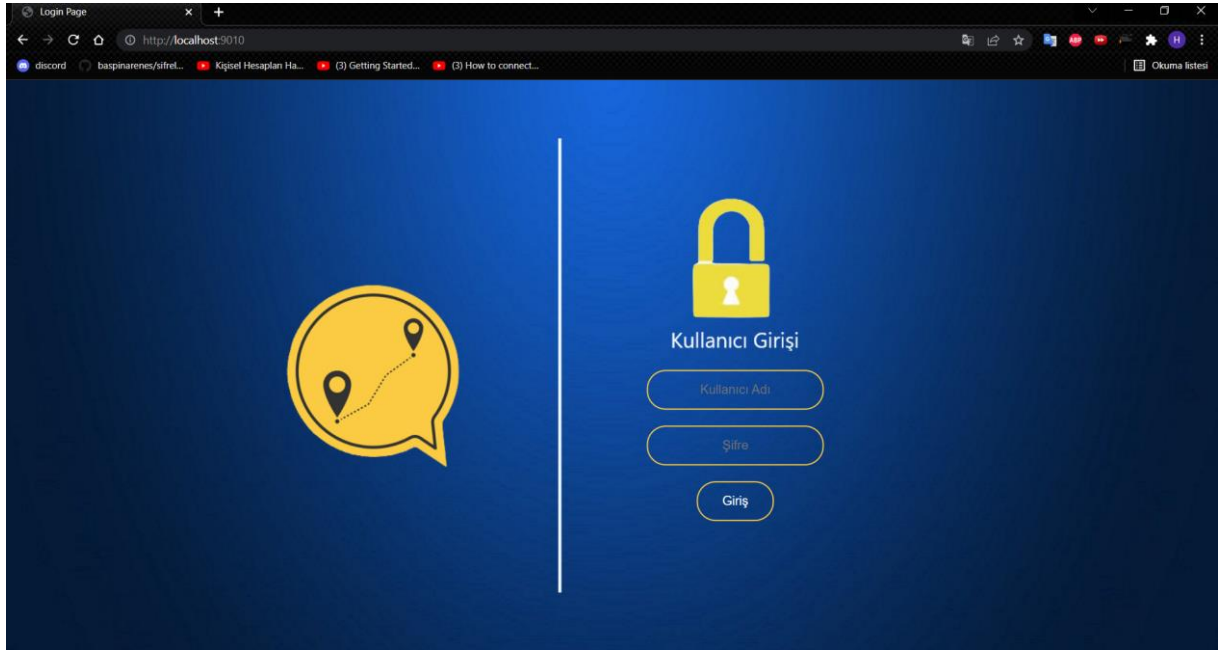
6. app.js Express ve bodyParser tanımlama, veri tipleri ve adları tanımlama, mysql.creatConnection() bağlantısı, app.post ile username ve password bilgilerinin gönderilmesi, app.post ile ilk tarih verisinin alınması, tarihDonustur(veri) fonksiyonu ile gelen tarih verilerini stringe çevirme, verigetir(tarih) fonksiyonu ile alınan tarihin öncesindeki yarım saatlik verileri alma ve app.listen ile localhosta bağlanma.

AKIŞ DİYAGRAMI:



PROJEDEN BAZI GÖRSELLER:





KAYNAKÇA:

https://www.youtube.com/watch?v=wZbX_Qh8zbl&t=3837s

<https://devnot.com/2020/redis-nedir-temel-kullanim-alanlari-nelerdir/>

<https://medium.com/batech/docker-nedir-docker-kavramlar%C4%B1-avantajlar%C4%B1-901b37742ee0>

<https://www.yusufsezer.com.tr/node-js-kurulumu/>

<https://www.youtube.com/watch?v=Ba1HWvEzvW4>

<https://www.youtube.com/watch?v=Mn0rdbJPWEo>

<https://www.youtube.com/watch?v=JLS9gg-oJPQ>

<https://www.youtube.com/watch?v=xvmrW5TNCPY&list=PLueWgsqWuSughEYBPCcN53SvmDJCodNJP>