



Öğrencinin;

ADI: Abdussamet

SOYADI: KACI

NO: 1721221007

BÖLÜM: Bilgisayar Mühendisliği

Projenin;

KONUSU: Soket Programlama ile Çoklu Chat Uygulaması

Dersin;

ADI: BİLGİSAYAR AĞLARI (BLM19304)

EĞİTMEN: Ali Yılmaz ÇAMURCU ve Samet KAYA

İçindekiler

1- Özet	3
2- Proje Konusu.....	3
3- Proje İş Akış Şeması	3
4- Proje Tasarım.....	4
5- Proje Çıktıları ve Başarı Ölçütleri	6
6- Proje Süresince Yapılanlar	7
7- Ek Açıklamalar	8
8- Kaynakça.....	8



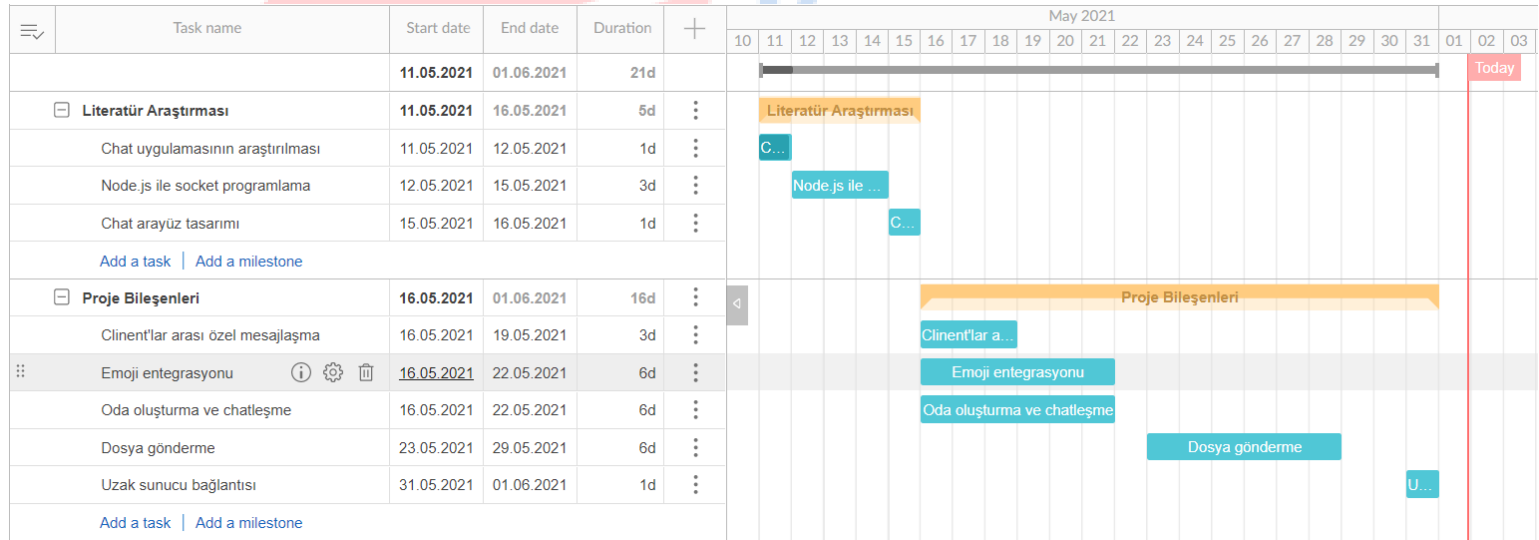
1- Özet

Bu projede bir chat uygulaması geliştirilmiştir. Bu uygulamada Node.js teknolojisi kullanıldı ve socket programlamama kütüphanelerinden yararlanıldı. Arayüz olarak HTML ve CSS kullanıldı. Bu uygulamada client'lar kendi aralarında mesajlaşabilmektedir (yazı, emoji, dosya, vs.). Ayrıca odalar açıp buralarda genel yazışmalar yapabilmektedirler. Server birden fazla client'ın bağlanmasını desteklemektedir.

2- Proje Konusu

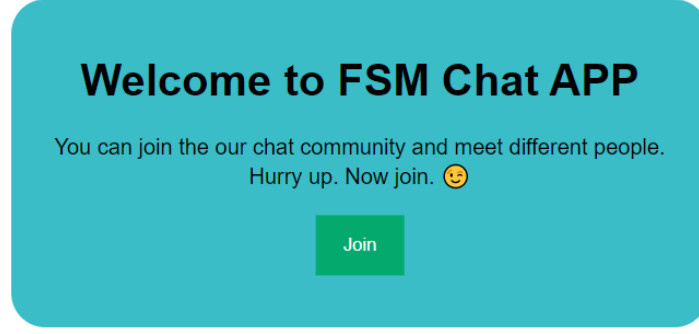
Projenin konusu çoklu client desteği olan bir server yazmaktır. Bu server client'lara bir çok hizmet sunmaktadır. Client'lar server'a bir isim ile bağlanırlar ve bağlanan client online olarak görünen tüm kullanıcıları görmektedir. Ayrıca client'lar kendi aralarında özel olarak mesajlaşabilirler. Dahası odalar oluşturup buralarda toplu bir şekilde konuşabilirler. Client'lar birbirlerine text, emoji ve dosya gibi birçok şeyi gönderebilirler. Ayrıca Heroku platformunda uzaktan bir server'da çalışır halde bulunmaktadır.

3- Proje İş Akış Şeması



Projenin iş akış şeması yukarıda görüldüğü gibidir. Bu proje temelde araştırma ve projenin bileşenleri yani parça parça istenilenlerin yerine getirilmesi ile gerçekleştirilmiştir.

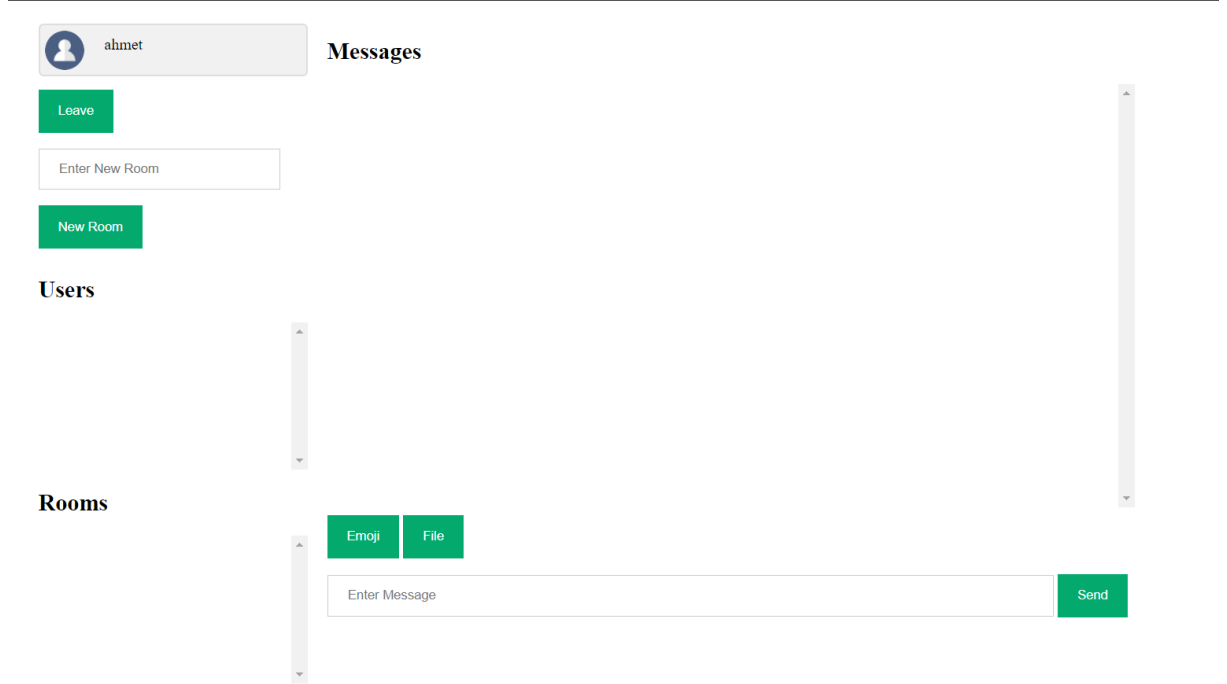
4- Proje Tasarım



Proje ilk açılırken yukarıdaki ekran görünmektedir. Bu ekran kullanıcıyı karşılamaktadır ve bir adet buton bulunmaktadır. Kullanıcı bu butona bastığında aşağıdaki ekran çıkmaktadır.

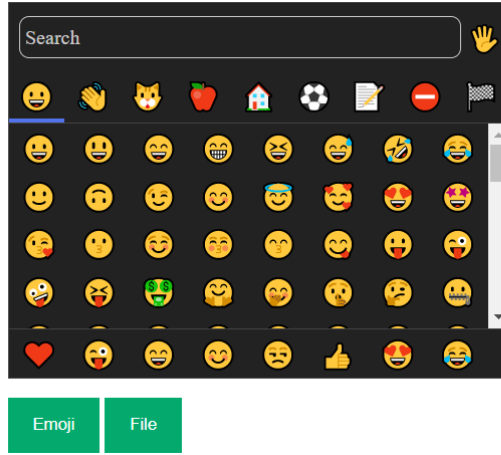


Bu ekranda da kullanıcı bir adet kullanıcı adı girer ve login butonuna basarak sisteme kayıt olur. Böylelikle chat uygulamasına giriş yapmış olur. Uygulamaya girdikten sonra ise aşağıdaki gibi bir ekran bizi karşılamaktadır.

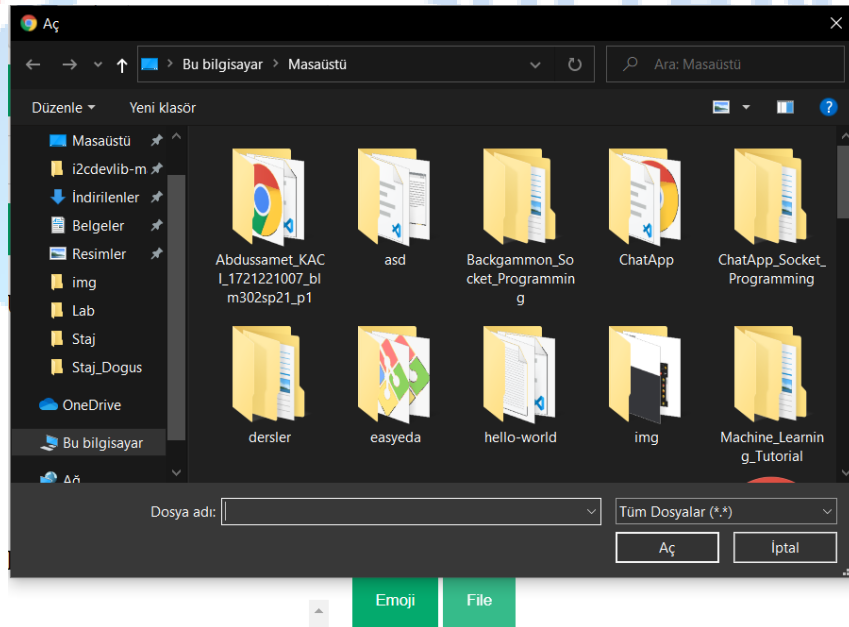


Bu ekranda sol en üst bölümde o andaki kullanıcının ismi görünmektedir. Hemen aşağısında chatten ayrılmak için bir buton bulunmakta ve onun altında ise odalar oluşturmak için bir alan bulunmaktadır. Sisteme giriş yapıldığında Users bölümünde o andaki online kullanıcılar görünmektedir ve oluşturulan odalar ise Rooms alanının altında sıralanmaktadır. Messages bölümünde ise kullanıcı üzerine tıkladığı kullanıcının veya bir odanın içerisindeki mesajları göstermektedir.

Messages



Emoji butonuna basıldığında emojiyi seçebileceğimiz bir alan açılmaktadır. Burada kullanıcı istediği emojiyi seçip karşı tarafa gönderebilmektedir.



File butonuna bastığımızda ise dosya seçebileceğimiz bir pencere açılmaktadır. Buradan istediğimiz dosyayı seçip gönderebiliriz.

5- Proje Çıktıları ve Başarı Ölçütleri

The screenshot shows a chat application interface. On the left, there is a sidebar with three sections: 'Messages' at the top, 'Users' in the middle, and 'Rooms' at the bottom. The 'Messages' section shows a user profile for 'ahmet' and a 'Leave' button. Below it is a text input field labeled 'Enter New Room' and a 'New Room' button. The 'Users' section shows a list of users: 'mehmet' and 'ayşe'. The 'Rooms' section shows a list of rooms: 'Python'. On the right, there is a 'Messages' section with a 'Send' button and a text input field labeled 'Enter Message'. Above the input field are two buttons: 'Emoji' and 'File'.

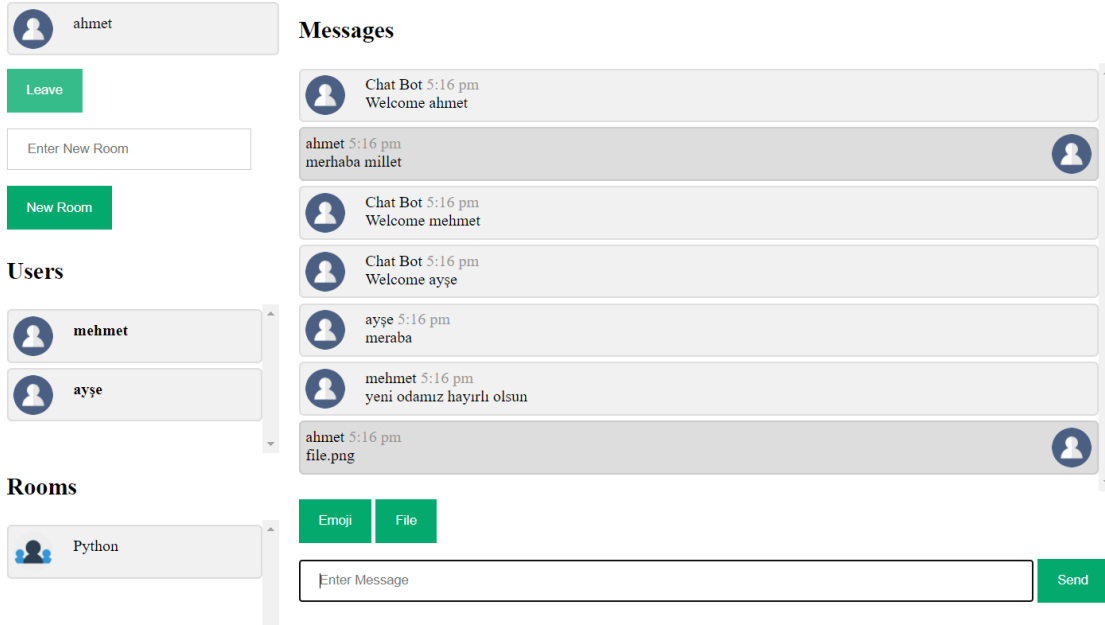
Uygulamada görüldüğü üzere online kullanıcılar ekranda görünmektedir. Kullanıcılar tarafından açılan odalar da bu pencerede gösterilmektedir.

The screenshot shows the same chat application interface as before, but now with a conversation. The 'Messages' section on the right shows a list of messages. The first message is from 'ahmet' at 5:14 pm, saying 'merhaba mehmet ben ahmet nasılsın'. The second message is from 'ahmet' at 5:14 pm, saying 'iyisindir inşallah'. The third message is from 'mehmet' at 5:14 pm, saying 'evet iyiyim sağol'. The fourth message is from 'ahmet' at 5:15 pm, saying 'deneme.txt'. The 'Users' section shows the same list of users. The 'Rooms' section shows the same list of rooms. The 'Messages' section on the right has a 'Send' button and a text input field labeled 'Enter Message'. Above the input field are two buttons: 'Emoji' and 'File'.

Bu ekranda da görüldüğü üzere iki kullanıcı arasındaki mesajlaşma başarılı bir şekilde gösterilmektedir. Kullanıcılar birbirlerine mesaj, emoji ve dosya gönderebilmektedir. Gönderilen dosya üzerine tıklanıldığında dosya kullanılan cihaza indirilmektedir.



Bu ekranda da görüldüğü üzere dosya tarayıcı üzerinden indirildiği gösterilmektedir.



Bu ekranda da bir grup içerisinde bir kaç kullanıcının mesajlaştığı gösterilmektedir. Odaya katılan kullanıcıyı ilk olarak bir bot karşılamaktadır ve kullanıcıya hoşgeldin demektedir. Daha sonra kullanıcılar burada toplu şekilde mesajlaşıp dosya gönderebilmektedirler.

6- Proje Süresince Yapılanlar

Projede ilk olarak server'da bulunması gereken hizmetlerin karşılanması için nasıl bir yapı kurulması gerektiği üzerine düşündüm. Daha sonra bu projede bizden istediğimiz teknoloji kullanılışında serbest bırakıldığı için hangi teknolojinin benim açımdan kullanılması daha rahat olacağına karar vermek için araştırma yaptım. Java ve Node.js arasında kaldım ve ikisi arasındaki avantaj ve dezavantajları sıraladım.

JavaSwing'de rahat bir şekilde istediğim tasarımı yapamamakla birlikte, bir önceki projede Java ile yaptığım server'a 2 den fazla client bağlanması sonucunda bazen server çökebiliyordu yetersiz memory'den dolayı. Ancak Node.js kullanılması durumunda HTML ve CSS ile arayüz hazırlanacağı için istediğim gibi bir ayaüzü daha kolay tasarlayabilecektim ve bu teknoloji üzerinde durmaya karar verdim. Ayrıca Node.js'de hafıza problemleri de yaşamıyordum.

Daha sonra Node.js de nasıl socket programlama yapılır onu araştırdım. Bunun üzerine bir video bitirip örnek bir uygulama geliştirdim. Daha sonra konsept arayüzleri inceledim ve kendim bir arayüz geliştirdim.

Bu uygulamayı baz alarak önce client'ler arasındaki mesajlaşmayı gerçekleştirdim. Burada client öncelikle server'a mesaj atmaktaydı. Bu mesajda mesajın içeriği ve kime gideceği bilgisi bulunmaktaydı. Daha sonra server ilgili client'a bu mesajı göndermektedir.

Bu adımdan sonra emoji desteği getirdim. Burada bir Node.js paketi olan emoji-picker'ı kullandım. NPM sayfasında dökümantasyonunu okuyup nasıl kullanacağımı öğrendim ve projeye entegre ettim.

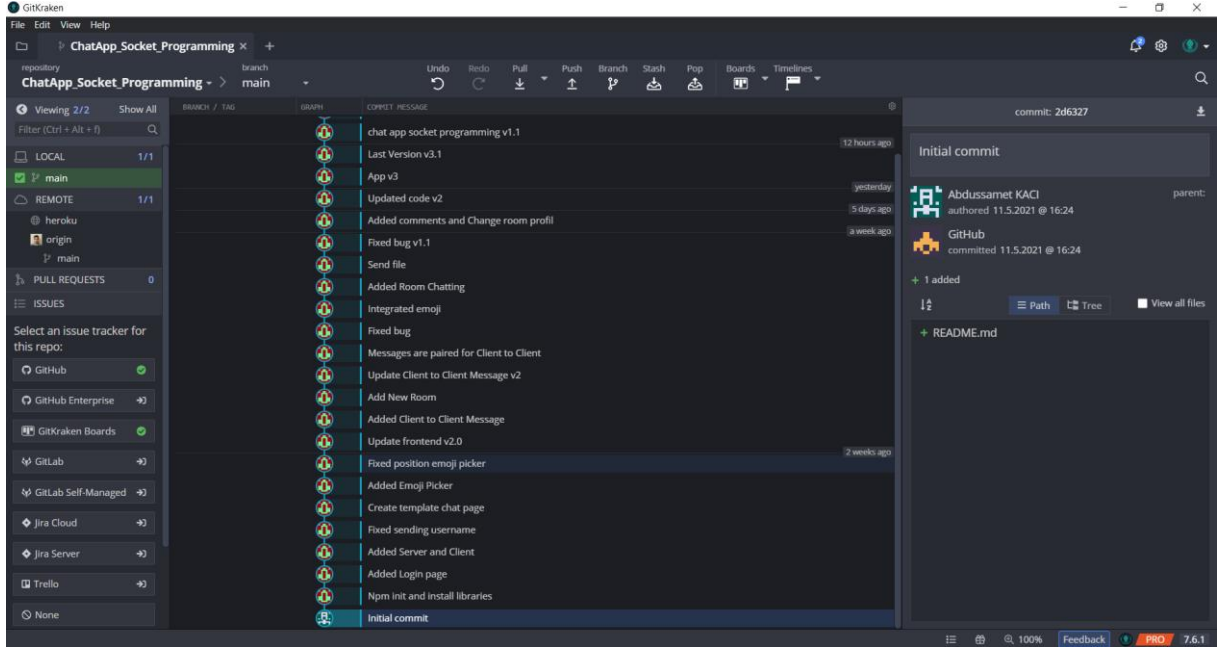
Daha sonra client'ların oda açmasını sağladım. Odalardaki mesajların oda içerisinde bulunan herkese ulaştırılmasını sağladım.

En son olarak da dosya aktarımını gerçekleştirdim. Burada mesaj ile aynı mantıkta gönderilmekte. Ancak dosyalar genellikle büyük boyutlara sahip olduklarından dolayı server'da kayıt altında tutulmaktadır. Kullanıcıya sadece dosya bilgisi gönderilmektedir. Kullanıcı eğer isterse dosya mesajı üzerine tıklayıp o dosyayı cihazına indirebilir.

En son olarak bizden AWS benzeri bir uzak sunucudan projemizin çalıştırılması istenmekteydi. Ben de git ile uyumlu ücretsiz bir servis olan (500 MB depolamaya kadar) Heroku'yu kullandım. Buraya kendi Node.js projemi kurdum ve uygulamamı uzaktan çalıştırdım.

Proje Linki: <https://chatappsocketprogramming.herokuapp.com/>

Ayrıca proje için verilen hafta içerisinde bir repo oluşturup düzenli olarak commit attım.



7- Ek Açıklamalar

Bu projede herhangi bir eksik bulunmamaktadır. İstenilen her şey eksiksiz bir şekilde yerine getirilmiştir.

8- Kaynakça

<https://socket.io/docs/v3/emit-cheatsheet/index.html>

<https://socket.io/docs/v3/rooms/>

<https://www.npmjs.com/package/emoji-picker-element>

<https://www.youtube.com/watch?v=jD7Fnbl76Hg>

Lab Videoları

Lab Ders Kaynakları