**ÖZET**

Bu ödev kapsamında, dört farklı ağ uygulamasının Python da gerçekleştirilerek istemci ve sunucu mimarisinin, veri iletişiminin nasıl gerçekleştirildiği uygulamalı olarak pekiştirilmesi amaçlanmıştır.

Proje gerçekleştirilirken Python versiyonu olarak version 3.6.8 kullanılmıştır.

**Bunlar:**

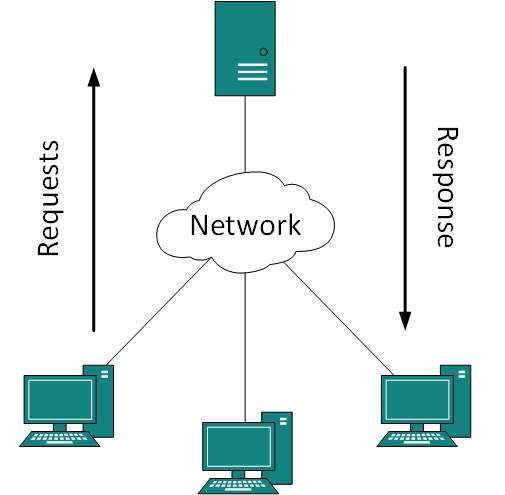
1-) Echo İstemci ve Sunucunun gerçeklenmesi.

2-) Chat İstemci ve Sunucunun gerçeklenmesi.

3-) Web (Textual veya metinsel) İstemci ve Sunucunun gerçeklenmesi.

4-) Whatsapp benzeri bir metinsel iletişim siteminin gerçeklenmesi.

**GİRİŞ**

****

**SUNUCU NEDİR?**

Sunucu, herhangi bir ağ ortamında diğer cihazların veya istemci bilgisayarların çeşitli isteklerine hizmet vermek amacı ile tasarlanmış özel bilgisayarlardır. Sunucu, kullanıcının verilere online olarak ulaşabilmesini sağlayan, bu verileri onların kullanımına sunan sistemlerdir. Bir sunucunun en önemli özelliği veri akışını güvenli, kesintisiz ve istikrarlı bir performans ile sağlayabilmesidir.

**İSTEMCİ NEDİR?**

İstemci, sistemdeki verilere ulaşmak için sunucudan istekte bulunan yapıdır. İstemci cihaz sunucu cihazdan izin verilen ölçülerde isteklerde bulunur ve sunucu cihaz da istemci cihaza ya isteğini karşılayacak cevabı gönderir ya da isteğin neden karşılanamadığını belirten bir cevap iletir.

**SOKET NEDİR?**

Soket, uygulamaların üzerinden veri gönderip alabileceği soyutlama katmanıdır. Soketler, uygulamaların ağ üzerindeki diğer uygulamalarla haberleşebilmesini olanaklı kılar. Bir makine üzerindeki uygulama programı tarafından sokete yazılan bilgi, ağ üzerindeki başka bir makine tarafından okunabilir. TCP/IP mimarisi ile kullanılan iki tip soket mevcuttur: Stream soket ve datagram soket. Stream soketler, veri haberleşmesinde uçtan uca protokol olarak TCP’yi kullanır ve böylece güvenilir veri transferi sağlar. Datagram soketler ise veri haberleşmesinde UDP’yi kullanır. Bu tür soketlerde gönderilen verinin karşı tarafa ulaşması garanti edilemez. 65500 byte uzunluğuna kadar veri içeren mesajlar, bu soketlerin kullanımı ile gönderilebilir. TCP/IP soketleri, temelinde 3 unsur tarafından tanımlanır: Internet adresi (IP), kullanılan veri haberleşme türü (TCP veya UDP), port numarası.

Soket programlamada yer alan temel metodlar;

• socket – Soket oluşturur.

• close – Soketi yok eder.

• connect – İki soket arasında bağlantı oluşturur.

• bind – Sunucu soketini bir adres ile etiketler.

• listen – Bir soketi durumları bekleyecek şekilde konfigüre eder.

• accept – Bağlantıyı kabul eder ve bu bağlantı için yeni bir soket oluşturur.

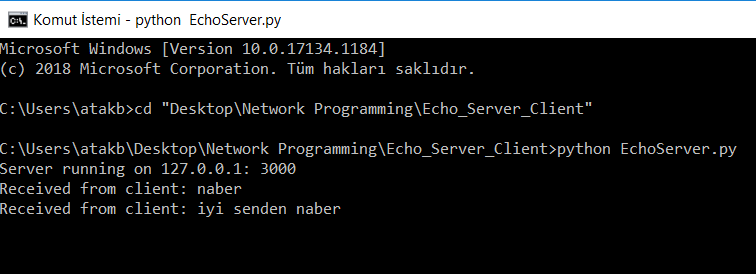
Haberleşme stili parametresi için **SOCK\_** öneki ile başlayan sabitler kullanılır. **SOCK\_STREAM** TCP tabanlı haberleşme stilini tanımlarken, **SOCK\_DGRAM** UDP tabanlı haberleşme stilini ifade eder.

**YAPILAN ÇALIŞMALAR**

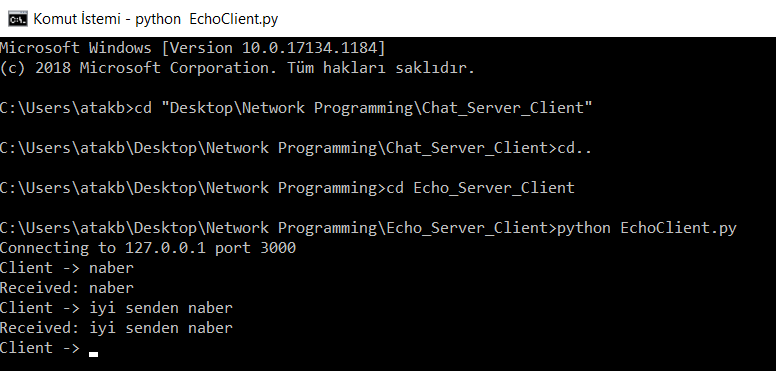
**ECHO İSTEMCİ VE SUNUCU**

Echo sunucu kendine bağlanan istemcilerden gelen bilgiyi aynen geri yansıtır. Bu ödevde, TCP ile istemci tarafında belirlenen port numarasına ve localhost adresine mesaj gönderimi sağlanarak sunucu tarafında da, aynı port numarası ile istemciden gönderilen mesajın tekrar sunucu tarafından istemciye gönderilmesi(Yankı) sağlandı.

**ECHO SUNUCU GÖRÜNÜMÜ**



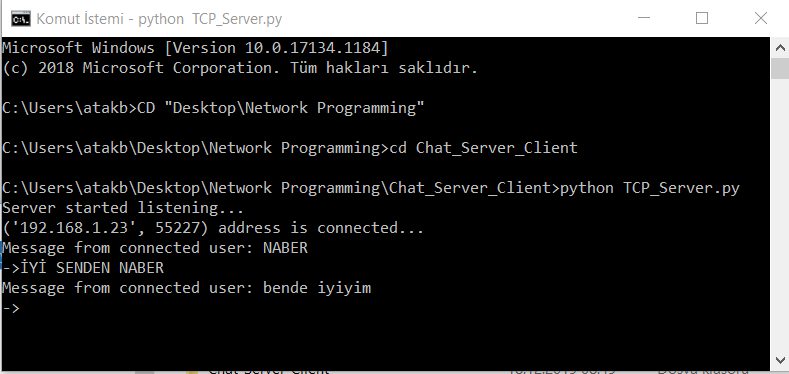
**ECHO İSTEMCİ GÖRÜNÜMÜ**



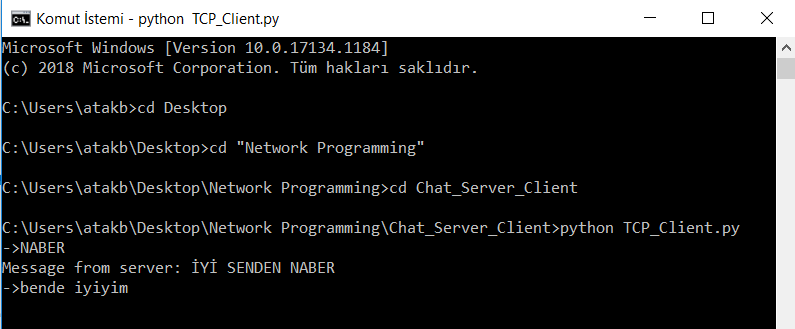
**CHAT İSTEMCİ VE SUNUCU**

Chat istemci ve sunucunun gerçeklenmesi ödevinde ise, TCP ile istemcinin sunucuya bağlanarak sunucuya bir mesaj göndermesi ve mesajı alan sunucunun istemciye mesaj göndermesini sağlayan bir mimari gerçekleştirildi.

**CHAT SUNUCU GÖRÜNÜMÜ**



**CHAT İSTEMCİ GÖRÜNÜMÜ**



**WEB İSTEMCİ VE SUNUCU**

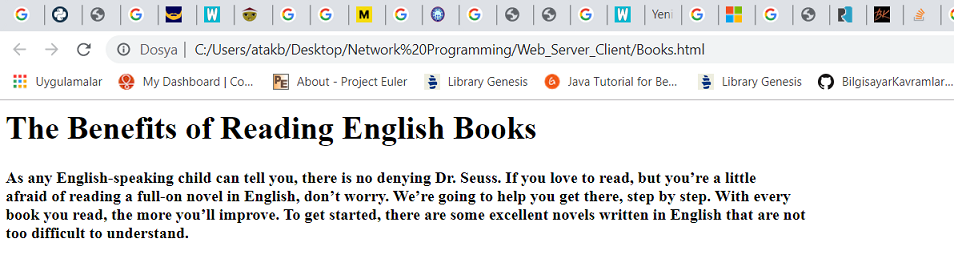
İstemci ­ sunucu haberleşmesinde, kullanıcıdan bilgiler bir HTML form yardımıyla alınır ve sunucuya iletilir. Veriler sunucuya iletilirken HTTP istek metotları kullanılır. En yaygın kullanılan metotlar şunlardır:

● **GET metodu :** Verilerin belirli bir kaynaktan istenmesi bu yolla olur. İstek URL vasıtası ile gönderilir. İstek gönderilirken URL içerisinde parametre gönderilebilir. GET istekleri tarayıcı geçmişinde saklanabilir ve tarayıcıya yer imi olarak eklenebilir. Bu yöntemle hassas bilgiler kesinlikle gönderilmez, çünkü URL de gözükürler. GET yönteminin URL boyutundan dolayı boyut kısıtlaması vardır.

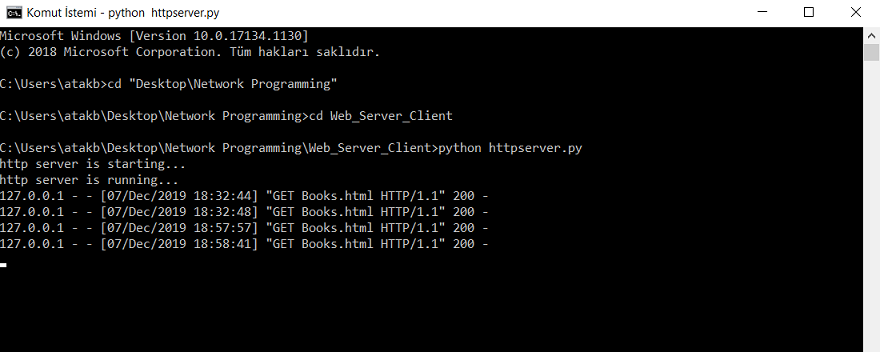
**● POST metodu** : Sunucuya işlenmek üzere verilerin gönderilmesi yöntemidir. POST metodu tarayıcı geçmişinde kalmaz ve yer imi olarak eklenemez. Boyut kısıtlaması yoktur

Web (Textual veya metinsel) istemci ve sunucunun gerçeklenmesi ödevinde, ilk olarak **GET** methodu ile çağırılıp içeriğini göreceğimiz örnek bir **HTML** sayfası oluşturuldu. Web üzerinde iletişim kurulacağı için istemci ve sunucu dosyalarına Python’ın **http** kütüphanesi eklendi. Sunucu çalıştırıldığın da, istemci tarafından sunucun içeriğini görme isteği geldiğinde içeriği göstermek için sunucuya oluşturulan örnek **HTML** sayfasının dizini verildi. Eğer bir hata olmazsa **200**(İstek başarılı alınmış ve cevap başarılı verilmiştir.) kodu ve sunucunun içeriği döndürüldü. Eğer **HTML** sayfası bulunamazsa **404**(İstek yapılan kaynağın (veya sayfanın) bulunamadığını belirtir.) hata kodu döndürüldü.

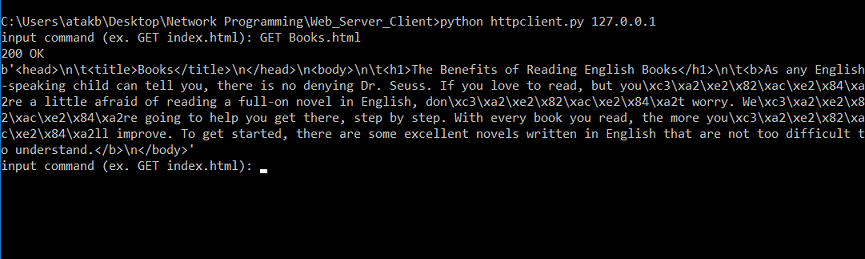
**HTML SAYFA GÖRÜNÜMÜ**



**WEB SUNUCU GÖRÜNÜMÜ**



**WEB İSTEMCİ GÖRÜNÜMÜ**

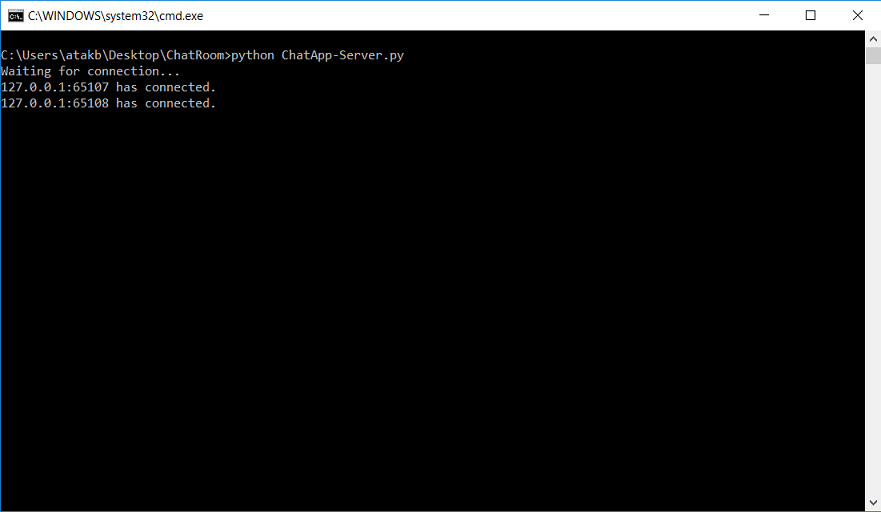


**WHATSAPP BENZERİ BİR METİNSEL İLETİŞİM SİSTEMİNİN GERÇEKLENMESİ**

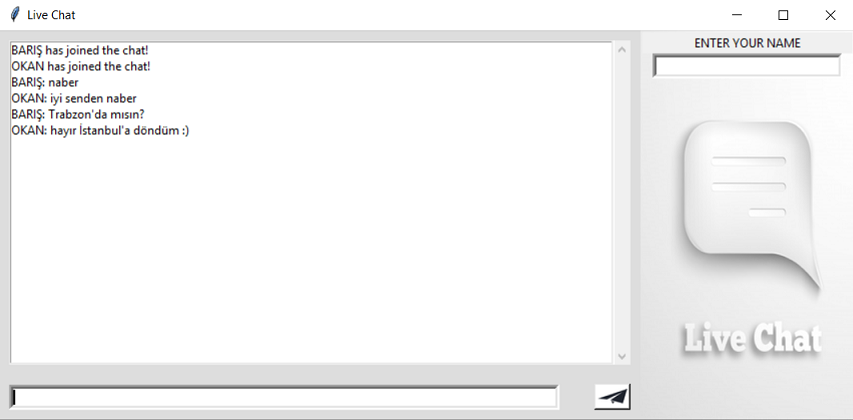
Whatsapp benzeri bir metinsel iletişim sisteminin gerçeklenmesi ödevinde ise, arayüze sahip çoklu istemciden oluşan bir chat uygulaması yapılması isteniyor. Çoklu istemci mimarisini sağlamak için Python’ın **threading** kütüphanesi ve arayüz tasarımı için Python’ın **tkinter** kütüphanesi kullanılmıştır.

Programı her defasında komut isteminde(CMD) de çalıştırmamak için **server** ve **client** adında iki adet **.bat** uzantılı doysa oluşturarak, içerisine oluşturulan **.py** uzantılı dosyalar atanmıştır. Böylece program direk çalıştırılabilir hale getirilmiştir.

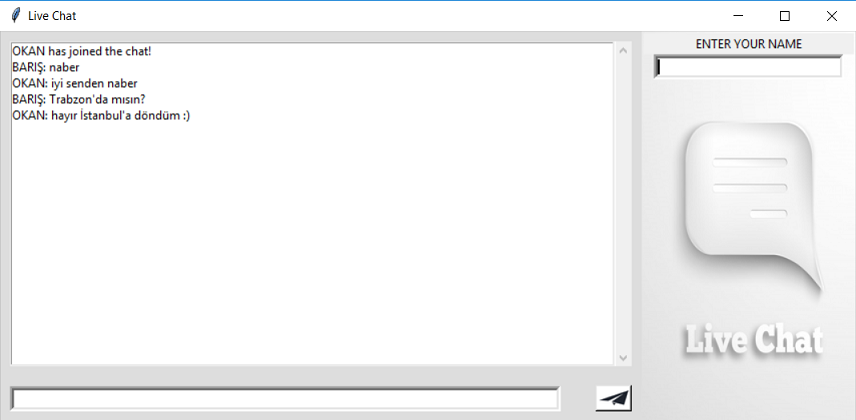
**SUNUCU GÖRÜNÜMÜ**



**İSTEMCİ-1 GÖRÜNÜMÜ**



**İSTEMCİ-2 GÖRÜNÜMÜ**



**SONUÇLAR VE DEĞERLENDİRME**

Ağ programlama teknik ve uygulamalı olarak pekiştirildi. Tüm projeler gerçekleştirilip test edildi ve başarı ile sonuç verdi.

**KAYNAKLAR**

<https://docs.python.org/3/library/socket.html>

<https://www.geeksforgeeks.org/socket-programming-python/>

<https://www.edureka.co/blog/socket-programming-python/>

<https://pythonprogramming.net/client-server-python-sockets/>

<https://www.journaldev.com/15906/python-socket-programming-server-client>

<https://www.godo.dev/tutorials/python-http-server-client/>

<https://www.pubnub.com/blog/socket-programming-in-python-client-server-p2p/>

<https://codinginfinite.com/python-chat-application-tutorial-source-code/>

<https://medium.com/swlh/lets-write-a-chat-app-in-python-f6783a9ac170>

<https://docs.python.org/2/library/tkinter.html>

<http://www.ktu.edu.tr/dosyalar/bilgisayar_b3f47.pdf>

<https://bakikaracay.com/6542-2>