



**T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY of ENGINEERING
DEPARTMENT of COMPUTER ENGINEERING**

SOCKET PROGRAMMING

CSE 476/575 MOBILE COMMUNICATIONS NETWORKS TERM PROJECT PROGRESS REPORT

STUDENT
Şeyda Nur DEMİR
12 10 44 042

LECTURER
Hasari ÇELEBİ

TEACHING ASSISTANT
Abdullah Salih BAYRAKTAR

KOCAELİ, 2020

1.DESCRPTION

Please complete the below Socket Programming Assignments using **Phyton**.

1. Web Server (Please see attached Lab1 for the details.): **30 points**
2. UDP Pinger (Please see attached Lab2 for the details.): **30 points**
3. Mail Client (Please see attached Lab3 for the details.) : **30 points**
4. Progress Report (Please put all the results of the above 3 assignments including the codes into a single report): **Due Date: December 1, 2020 (5 points)**
5. Final Report (Please put all the results of the above 3 assignments including the codes into a single report): **Due Date: January 4, 2021 (5 points)**

Assignment 1: Web Server

In this assignment, you will develop a simple **Web Server** in Python that is capable of processing only one request.

Specifically, your Web Server will

- (i) create a connection socket when contacted by a client (browser);
- (ii) receive the HTTP request from this connection;
- (iii) parse the request to determine the specific file being requested;
- (iv) get the requested file from the server's file system;
- (v) create an HTTP response message consisting of the requested file preceded by header lines;
- (vi) and send the response over the TCP connection to the requesting browser. If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message.

In the attached **Lab1**, we provide the skeleton code for your server. Your job is to complete the code, run your server, and then test your server by sending requests from browsers running on different hosts. If you run your server on a host that already has a Web Server running on it, then you should use a different port than port 80 for your Web Server.

Assignment 2: UDP Pinger

In this programming assignment, you will write a client **Ping Program** in Python.

Your client will send a simple ping message to a server, receive a corresponding pong message back from the server, and determine the delay between when the client sent the ping message and received the pong message. This delay is called the Round Trip Time (RTT). The functionality provided by the client and server is similar to the functionality provided by standard ping program available in modern operating systems.

However, standard ping programs use the Internet Control Message Protocol (ICMP) (which we will study in Chapter 4). Here we will create a nonstandard (but simple!) UDP-based ping program. Your ping program is to send 10 ping messages to the target server over UDP. For each message, your client is to determine and print the RTT when the corresponding pong message is returned. Because UDP is an unreliable protocol, a packet sent by the client or server may be lost. For this reason, the client cannot wait indefinitely for a reply to a ping message. You should have the client wait up to one second for a reply from the server; if no reply is received, the client should assume that the packet was lost and print a message accordingly.

In this assignment, you will be given the complete code for the server (available in the attached **Lab2**). Your job is to write the client code, which will be very similar to the server code. It is recommended that you first study carefully the server code. You can then write your client code, liberally cutting and pasting lines from the server code.

Assignment 3: Mail Client

The goal of this programming assignment is to create a simple **Mail Client** that sends email to any recipient.

Your client will need to establish a TCP connection with a mail server (e.g., a Google mail server), dialogue with the mail server using the SMTP protocol, send an email message to a recipient (e.g., your friend) via the mail server, and finally close the TCP connection with the mail server.

For this assignment, the **Lab3** document provides the skeleton code for your client. Your job is to complete the code and test your client by sending email to different user accounts. You may also try sending through different servers (for example, through a Google mail server and through your university mail server).

2.RULES

Obey the rules given above.

No cheating.

3.SUBMISSION

3.1.Files

Name your zip file as StudentNumber.zip.

3.2. Deadline

Submit your project progress report due to 01.12.2020 23:55.

3.3.Grading

You will be graded over 100 points.

4.WHAT DID I DO?

- ✓ Web Server
 - ✓ Lab1
 - ✓ OptionalExercise1
 - ✓ OptionalExercise2
- ✓ UDP Pinger
 - ✓ Lab2
 - ✓ OptionalExercise1
 - ✓ OptionalExercise2
- ✓ Mail Client
 - ✓ Lab3
 - ✓ Optional Exercise1
 - ✓ OptionalExercise2

5.WHAT DID NOT I DO?

- ✓ I did all part.
- ✓ Only in “Lab3” first case, you change with your e-mail address and with your encoded base 64 password auth from <https://www.base64encode.org/>

END OF THE REPORT

UPDATED : 30 November 2020, 23:55

STUDENT

Şeyda Nur DEMİR
12 10 44 042

LECTURER

Hasari ÇELEBİ

TEACHING ASSISTANT

Abdullah Salih BAYRAKTAR

KOCAELİ, 2020