**MREŽNO PROGRAMIRANJE**

**Vježba 4**

**ZADATAK 1**

Osnovni klijent-server program iz laboratorijske vježbe 2 prilagodite na način da radi preko UDP protokola. Kod i screenshot iz konzole priložite vježbi.

***FILE1: tcp\_server.py***

import socket

UDP\_IP = "127.0.0.1"

UDP\_PORT = 5005

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) # UDP

sock.bind((UDP\_IP, UDP\_PORT))

while True:

data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes

print("received message: %s" % data)

***FILE2: tcp\_client.py***

import socket

UDP\_IP = "127.0.0.1"

UDP\_PORT = 5005

MESSAGE = *b*"Hello, World!"

print("UDP target IP: %s" % UDP\_IP)

print("UDP target port: %s" % UDP\_PORT)

print("message: %s" % MESSAGE)

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) # UD

sock.sendto(MESSAGE, (UDP\_IP, UDP\_PORT))

Text

Description automatically generated

**ZADATAK 2 - *ECHO CLIENT-SERVER***

Osnovni klijent-server program iz laboratorijske vježbe 2 prilagodite na način da radi preko UDP protokola. Kod i screenshot iz konzole priložite vježbi.

***FILE1: echo\_server.py***

# -- coding: utf-8 --

# echo\_server.py

import socket

host = socket.gethostname()

port = 12345

echo\_server = socket.socket() # tcp socket

echo\_server.bind((host, port))

echo\_server.listen(5)

print("Cekam klijenta!")

conn, addr = echo\_server.accept() # prihvaćanje konekcije kada se klijent spoji

print("Spojen: ", addr)

while True:

data = conn.recv(1024) # prihvaćanje podataka od klijenta

if not data: # ako nema podataka, izađi

break

conn.sendall(data) # vrati primljene podatke klijentu

conn.close()

***FILE2: echo\_client.py***

# -- coding: utf-8 --

# echo\_client.py

import socket

host = socket.gethostname()

port = 12345

client\_socket = socket.socket() # tcp socket

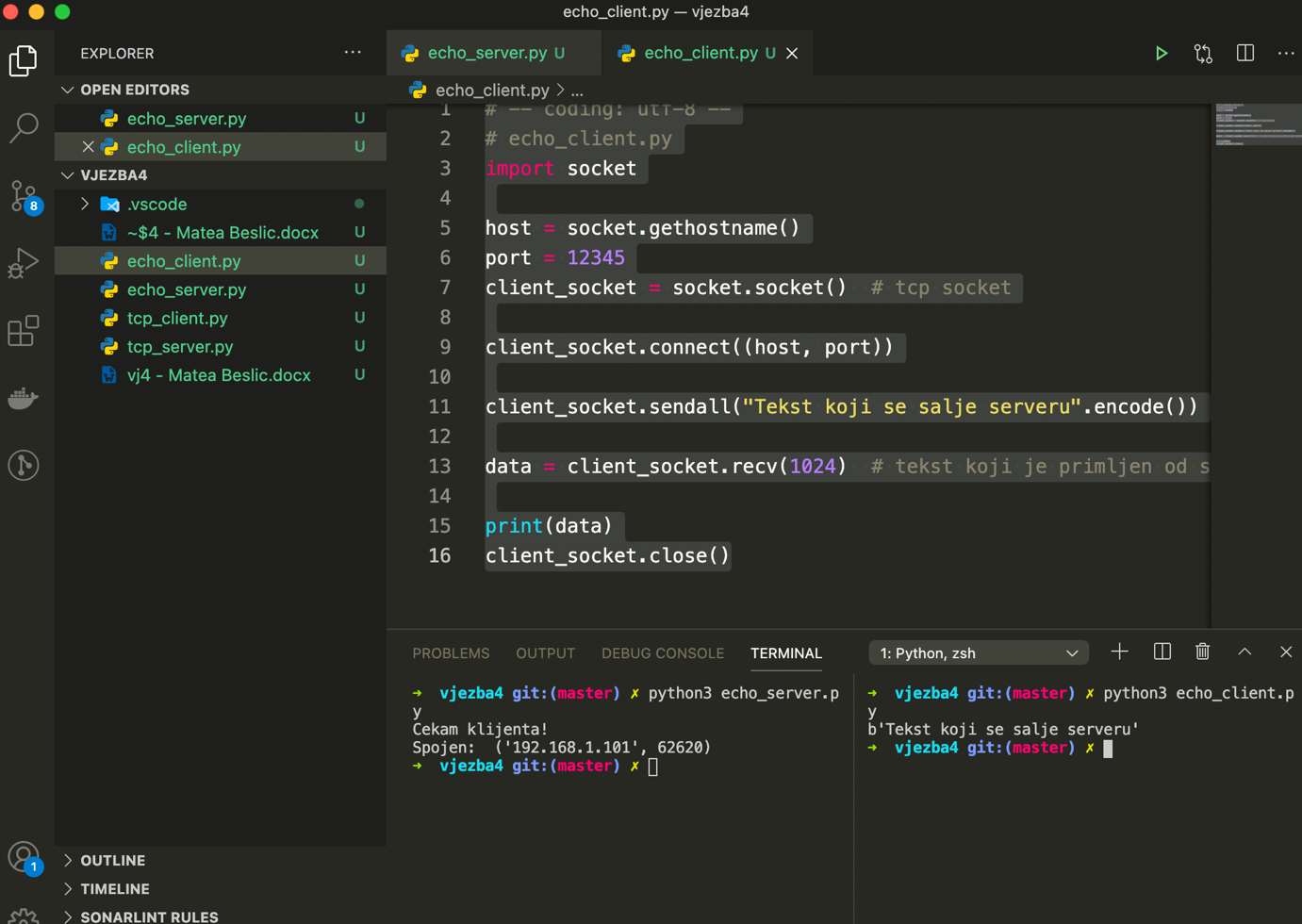
client\_socket.connect((host, port))

client\_socket.sendall("Tekst koji se salje serveru".encode())

data = client\_socket.recv(1024) # tekst koji je primljen od servera

print(data)

client\_socket.close()

******

***Ove programe prilagodite na sljedeći način:***

* ***U svaki kod treba napraviti import datetime modula, te napraviti sljedeću liniju koda:***
  + ***print datetime.datetime.now()***
* ***local\_machine\_info.py program iz prve vježbe treba importirati i pozvati funkciju koja će ispisati hostname i ip adresu.***
  + ***from local\_machine\_info import print\_machine\_info***
* ***U klijent programu, umjesto slanja teksta, napravite da se korisnik programa pita tekst za unos, te se onda taj isti tekst pošalje serveru koji ga vrati natrag i klijent ga ispiše. Koristite raw\_input funkciju koja prima bilo kakav unos od strane korisnika i sprema ga u string.***
* ***Ukoliko korisnik unese string 'vaše\_ime\_prezime' server treba vratiti klijentu obavijest da taj unos nije podržan.***
* ***Također, program prilagodite na način da server cijelo vrijeme sluša na dolazne konekcije, a ne da izađe nakon što primi i zatvori konekciju.***

***FILE1: echo\_server2.py***

# -- coding: utf-8 --

# echo\_server.py

import socket

import datetime

from local\_machine\_info import print\_machine\_info

print\_machine\_info()

print(datetime.datetime.now())

host = socket.gethostname()

port = 12345

echo\_server = socket.socket() # tcp socket

echo\_server.bind((host, port))

echo\_server.listen(5)

while True:

conn, addr = echo\_server.accept()

print("Spojen: ", addr)

data = conn.recv(1024) # prihvaćanje podataka od klijenta

if data == *b*"matea\_beslic":

data2 = "Unos nije podrzan"

conn.sendall(data2.encode())

break

else:

conn.sendall(data) # vrati primljene podatke klijentu

conn.close()

***FILE2: echo\_client2.py***

# -- coding: utf-8 --

# echo\_client.py

import socket

import datetime

from local\_machine\_info import print\_machine\_info

print\_machine\_info()

print(datetime.datetime.now())

host = socket.gethostname()

port = 12345

client\_socket = socket.socket() # tcp socket

client\_socket.connect((host, port))

print("Unesite tekst koji zelite poslati: ")

data = input()

client\_socket.sendall(data.encode())

data = client\_socket.recv(1024) # tekst koji je primljen od servera

print(data)

client\_socket.close()

***Graphical user interface, text

Description automatically generated***