 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №2**

з дисципліни «ВЕБ-технології та ВЕБ-дизайн»

# на тему: «Клієнт-серверна взаємодія»

**Виконав:**

студент гр. БС-81

Збаровський Д. Д.

**Перевірили:**

ас. Давидько О.Б.

ас. Матвійчук О.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

Створити веб-сервіс та клієнт до нього.

1. Сервіс та клієнт повинні взаємодіяти шляхом передачі даних по TCP сокетах в форматі Прізвище[0] <= “О” ? JSON : XML
2. Кожен з клієнтів після підключення повинен передати серверу структуру з 1 ключем “id” та значенням “Прізвище\_(#клієнта)”
3. Після підключення першого клієнта сервер має запустити глобальний таймер на #Варіант + 10 секунд
4. По завершенню таймера відправити кожному з клієнтів список з усіма підключеними клієнтами з датами їх підключення, а також дату старту таймера
5. Для отримання змін в сокетах скористатись механізмами для роботи із сокетами, що підтримуються обраною вами мовою. наступним системним інструментом Select()
6. \* Екстра варіант [epoll/kqueue/I/O Completion Ports] (в залежності від обраної вами системи)
7. Перевірити роботу системи з (#варіант mod 10) + 10 клієнтами
8. В процесі роботи не використовувати сторонні бібліотеки для роботи в сокетами, з XML та JSON по необхідності

**Виконання:**

**Client.py**import socket

import json

def start\_client(port):

user\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

user\_socket.connect(('localhost', port))

user\_name = input("Input a name:")

user\_id = input("Input an ID:")

user\_info = {"Name": user\_name, "ID": user\_id}

sending = json.dumps(user\_info)

user\_socket.sendall(sending.encode())

data = user\_socket.recv(2048)

print(f"Users: {data.decode()}")

user\_socket.close()

if \_\_name\_\_ == '\_\_main\_\_':

port = 7777

start\_client(port)

**Server.py**import socket

import time

import threading

import datetime

import json

is\_send = False

cycle = False

user\_list = []

timer\_start = ""

def timer():

global timer\_start

global is\_send

timer\_start = str(datetime.datetime.now())

time.sleep(3)

is\_send = True

def start\_server(port):

global cycle

global is\_send

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind(('localhost', port))

server.listen()

print("Server is listening")

while True:

user\_socket, address = server.accept()

data = user\_socket.recv(2048)

users\_data = (json.loads(data.decode()))

user\_accept\_socket = threading.Thread(target=timer)

user\_accept\_socket.start()

users\_data["connection\_time"] = str(datetime.datetime.now())

users\_data["timer\_start"] = timer\_start

user\_list.append(users\_data["Name"])

user\_list.append(users\_data["connection\_time"])

user\_list.append(users\_data["timer\_start"])

send\_json = json.dumps(user\_list)

print(f"User:{send\_json}")

while not cycle:

if is\_send:

user\_socket.send(send\_json.encode())

break

is\_send = False

if \_\_name\_\_ == '\_\_main\_\_':

port = 7777

start\_server(port)