

Proiect socket

Marinescu Mihnea-Ionut Grupa 323

Scopul Proiectului:

Scopul acestui prim proiect este de a crea un sistem simplu de comunicare între un client scris în PHP și un server scris în Python folosind protocolul UDP (User Datagram Protocol).

Proiectul permite utilizatorului să aleagă din mai multe opțiuni disponibile și să comunice cu serverul în funcție de opțiunea aleasă. Opțiunile includ:

- Opțiunea 1:** Serverul furnizează un număr aleatoriu între 1 și 100.
- Opțiunea 2:** Clientul introduce un număr, iar serverul returnează dublul acestuia.
- Opțiunea 3:** Clientul introduce un număr, iar serverul returnează dacă acesta este par sau impar.
- Opțiunea 4:** Clientul introduce un număr, iar serverul returnează dacă acesta este un număr prim sau nu.

Componente ale Proiectului

Serverul UDP (Python)

- Serverul a fost scris în Python și utilizează PyCharm ca mediul de dezvoltare.
- Serverul ascultă pe o adresă IP specificată (127.0.0.1) și pe un port specificat (12345).
- Acesta primește cereri de la client, interpretează opțiunile și returnează răspunsurile corespunzătoare.
- Serverul folosește un socket UDP pentru a primi și trimite date către client.

```
import socket
import random
import math

server_ip = "127.0.0.1" # IP-ul pe care serverul va asculta
server_port = 12345 # Portul pe care serverul va asculta

# Creează un socket UDP
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Leagă socket-ul la o adresă IP și port
sock.bind((server_ip, server_port))

def primulet(num):
    if num <= 1:
        return False
    for i in range(2, int(math.sqrt(num)) + 1):
        if num % i == 0:
            return False
    return True

while True:
    print("Așteptăm pentru a primi date...")
    data, address = sock.recvfrom(1024)
    data = data.decode('utf-8')
```

```

if data == "optiunea1":
    raspuns = str(random.randint(1, 100))
elif data.startswith("optiunea2"):
    numar = int(data.split(":")[1])
    raspuns = str(numar * 2)
elif data.startswith("optiunea3"):
    numar = int(data.split(":")[1])
    if numar % 2 == 0:
        raspuns = "Numărul este par"
    else:
        raspuns = "Numărul este impar"
elif data.startswith("optiunea4"):
    numar = int(data.split(":")[1])
    if primuleet(numar):
        raspuns = "Numărul este prim"
    else:
        raspuns = "Numărul nu este prim"
else:
    raspuns = "Opțiune invalidă"

print(f"Am primit: {data}, Trimit: {raspuns}")
sock.sendto(raspuns.encode('utf-8'), address)

```

Clientul UDP (PHP)

- Clientul a fost scris în PHP și oferă o interfață pentru utilizator pentru a alege opțiunile disponibile.
- Acesta creează un socket UDP pentru a comunica cu serverul.
- Permite utilizatorului să aleagă opțiunea dorită și să trimită cererea către server.
- Clientul așteaptă răspunsul de la server și îl afișează în consolă.

```

<?php
$server_ip = "127.0.0.1";
$server_port = 12345;

$socket = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);

while (true) {
    echo "Alegeți o opțiune (1, 2, 3 sau 4): ";
    $option = trim(fgets(STDIN));

    if ($option == "1") {
        // Opțiunea 1: Cerem un număr random de la server
        $message = "optiunea1";
    } elseif ($option == "2") {
        // Opțiunea 2: Cerem să introducem un număr și serverul să-l
        returneze dublat
        echo "Introduceți un număr: ";
        $numar = trim(fgets(STDIN));
        $message = "optiunea2:" . $numar;
    } elseif ($option == "3") {
        // Opțiunea 3: Cerem să introducem un număr și serverul să
        returneze dacă acesta e par sau impar
        echo "Introduceți un număr: ";
        $numar = trim(fgets(STDIN));
        $message = "optiunea3:" . $numar;
    } elseif ($option == "4") {
        // Opțiunea 4: Cerem să introducem un număr și serverul să

```

```

returneze dacă acesta e prim sau nu
    echo "Introduceți un număr: ";
    $numar = trim(fgets(STDIN));
    $message = "optiunea4:" . $numar;
} else {
    echo "Opțiune invalidă.\n";
    continue;
}

// Trimite mesajul către server
socket_sendto($socket, $message, strlen($message), 0, $server_ip,
$server_port);

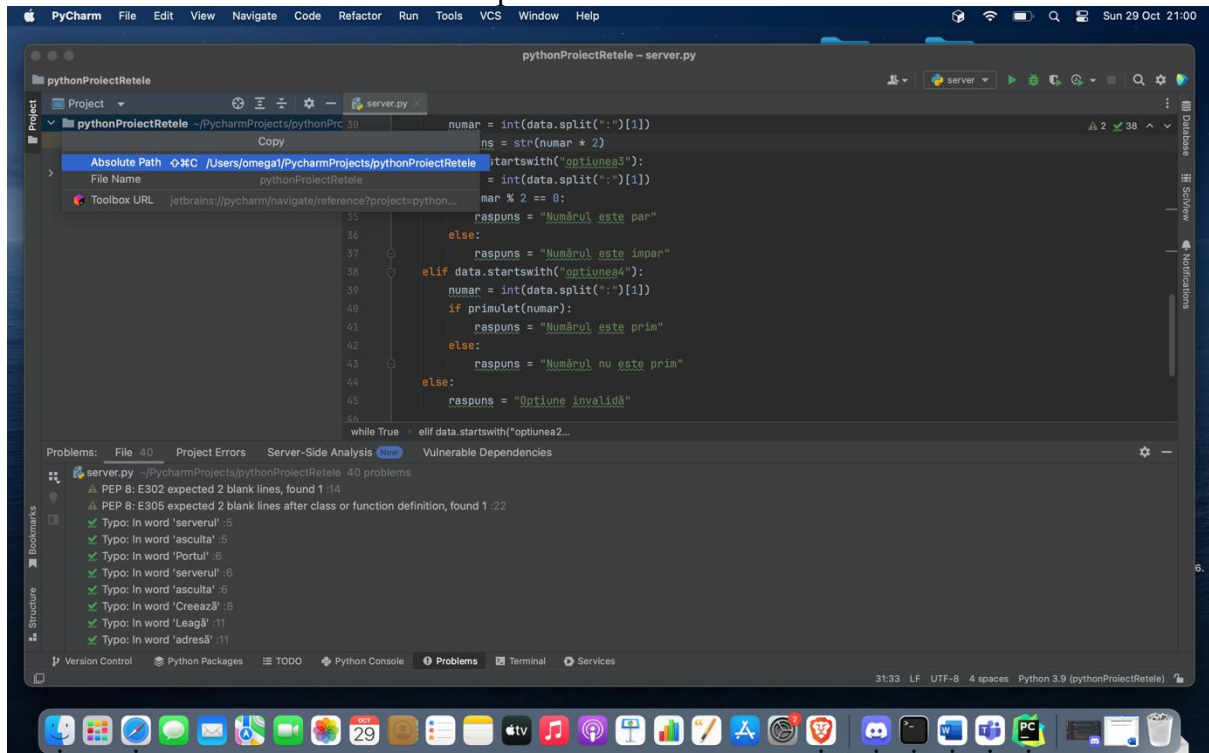
// Așteaptă răspunsul de la server
socket_recvfrom($socket, $response, 1024, 0, $server_ip, $server_port);

echo "Răspuns de la server: $response\n";
}

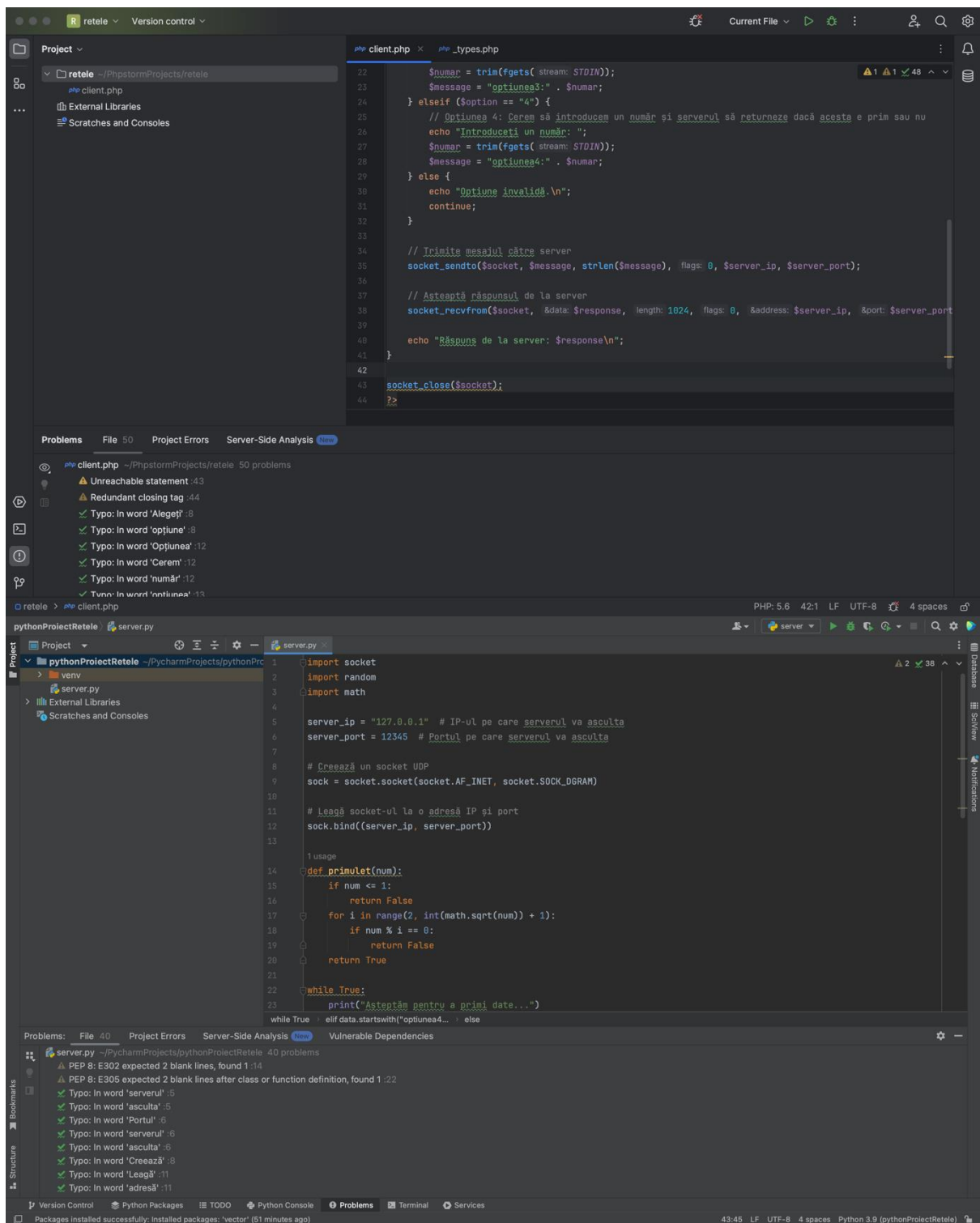
socket_close($socket);
?>

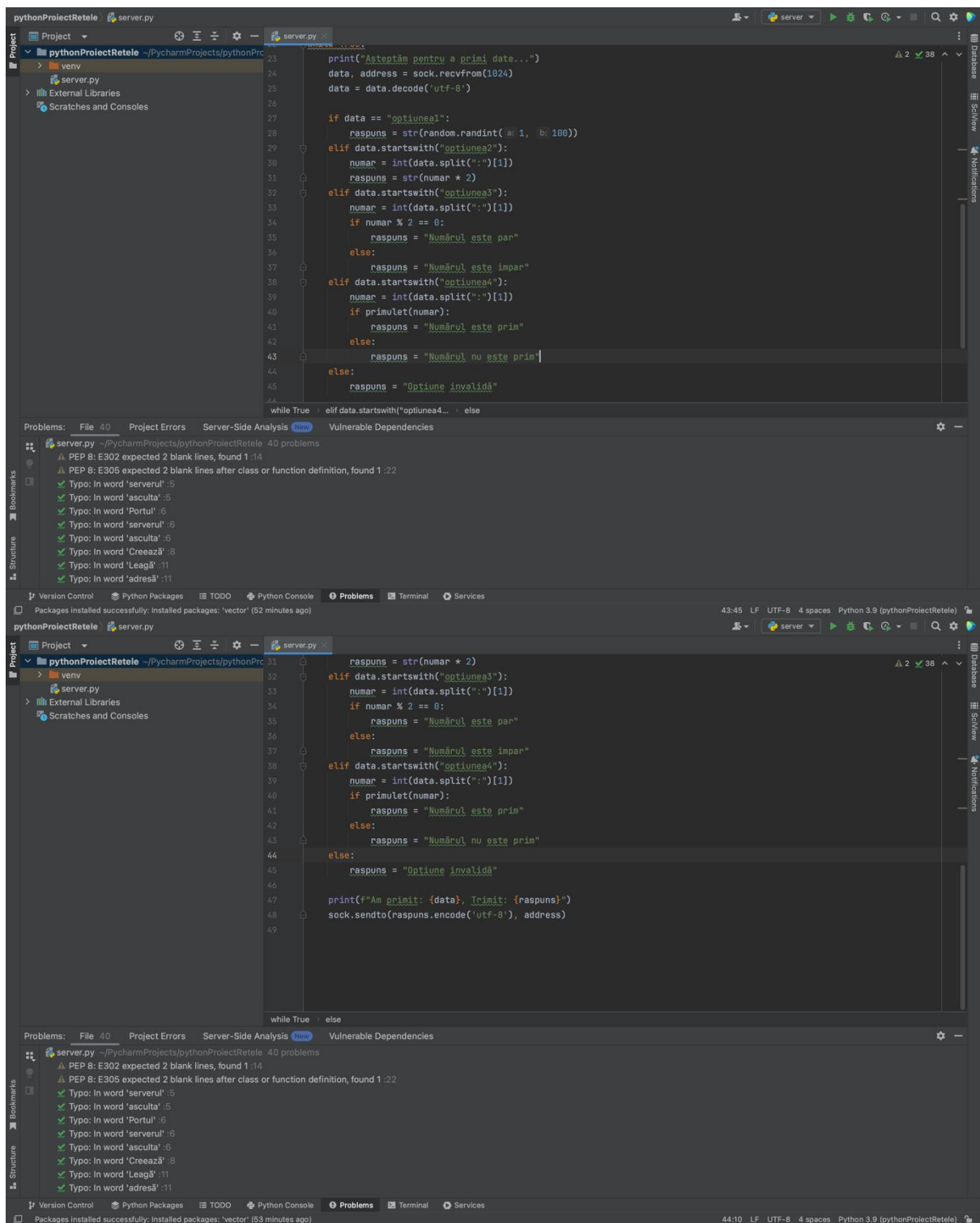
```

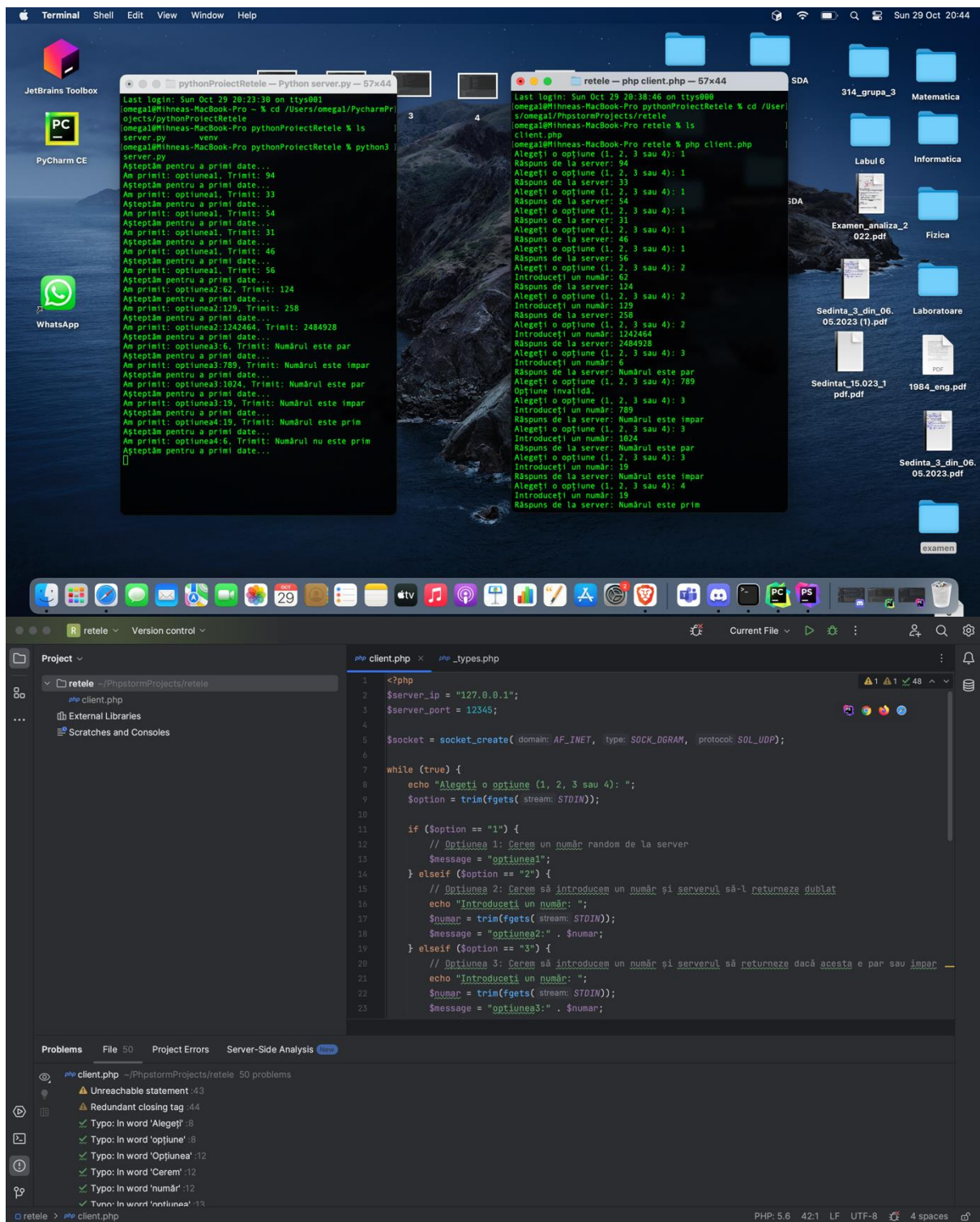
Mai jos voi atasa cateva capturi de ecran cu codul utilizat in aplicatii, precum si diverse rezultate obtinute in urma diverselor optiuni si date introduse de client.



Copiem locatia Path-ului pentru a o folosi ulterior in Terminal.

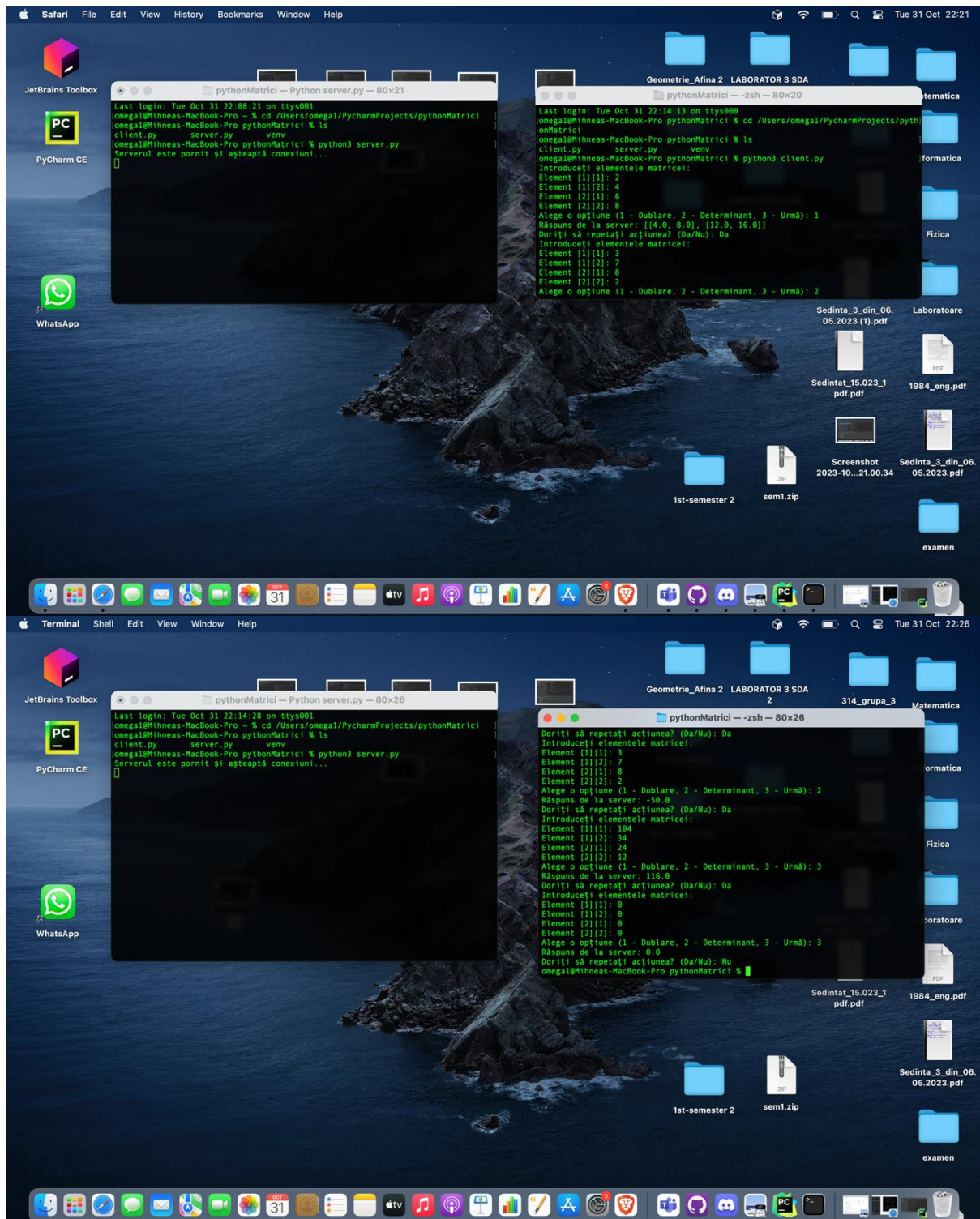


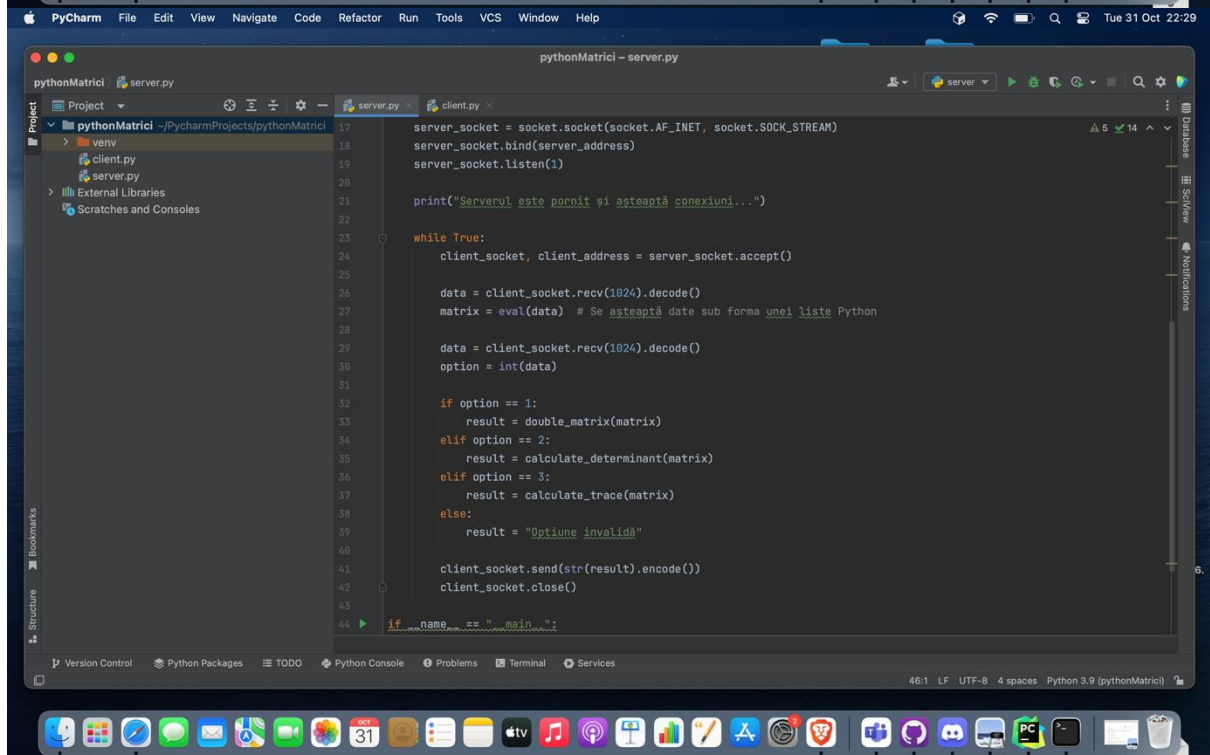
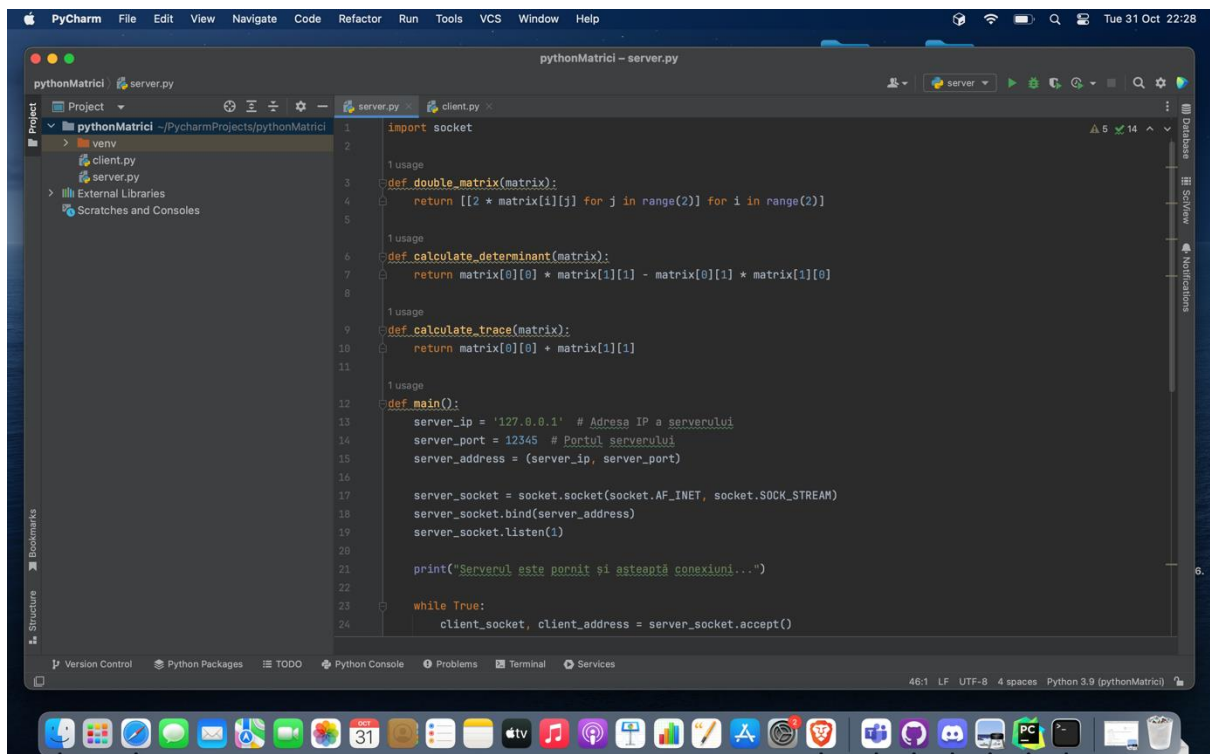




Scopul celui de al doilea proiect este de a crea un sistem simplu de comunicare între un client un server scris în Python folosind protocolul TCP. Acest proiect se bazează pe trimiterea de către client a unei matrici 2x2, iar serverul poate întoarce funcție de dorință

clientului dublul sau, determinantul sau urma matricei. Atasez mai jos capturile de ecran





PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help Tue 31 Oct 22:32

pythonMatrici - client.py

```
1 client_socket.connect(server_address)
2
3 while True:
4     matrix = input_matrix()
5     option = int(input("Alege o optiune (1 - Dublare, 2 - Determinant, 3 - Urmă): "))
6
7     matrix_data = str(matrix)
8     client_socket.send(matrix_data.encode())
9
10    option_data = str(option)
11    client_socket.send(option_data.encode())
12
13    response = client_socket.recv(1024).decode()
14    print("Răspuns de la server:", response)
15
16    repeat = input("Doriti să repetați acțiunea? (Da/Nu): ")
17    if repeat.lower() != "da":
18        break
19
20 client_socket.close()
21
22 if __name__ == "__main__":
23     main()
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
```

main() while True

Version Control Python Packages TODO Python Console Problems Terminal Services 24:90 LF UTF-8 4 spaces Python 3.9 (pythonMatrici)

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help Tue 31 Oct 22:30

pythonMatrici - client.py

```
1 import socket
2
3 1 usage
4 def input_matrix():
5     matrix = []
6     print("Introduceți elementele matricei:")
7     for i in range(2):
8         row = []
9         for j in range(2):
10             element = float(input(f"Element [{i + 1}][{j + 1}]: "))
11             row.append(element)
12         matrix.append(row)
13     return matrix
14
15 1 usage
16 def main():
17     server_ip = '127.0.0.1' # Adresa IP a serverului
18     server_port = 12345 # Portul serverului
19     server_address = (server_ip, server_port)
20
21     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
22     client_socket.connect(server_address)
23
24     while True:
25         matrix = input_matrix()
26         option = int(input("Alege o optiune (1 - Dublare, 2 - Determinant, 3 - Urmă): "))
27
28         matrix_data = str(matrix)
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
```

main() while True

Version Control Python Packages TODO Python Console Problems Terminal Services 24:90 LF UTF-8 4 spaces Python 3.9 (pythonMatrici)