

COMPUTER NETWORKS (UE22CS252B) MINI PROJECT

FILE TRANSFER SYSTEM

AAKANKSH SEELIN(PES2UG22CS003)

AAKRITI MISHRA(PES2UG22CS004)

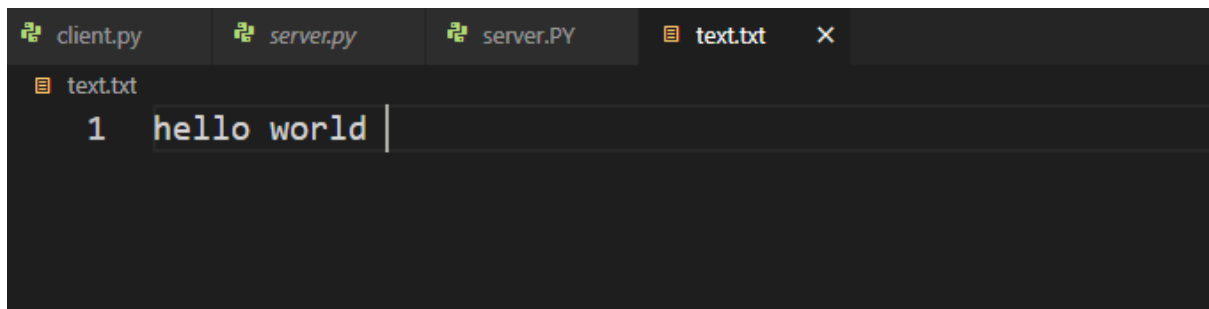
Client.py:

```
client.py x Win64OpenSSL_Light-3_2_1.exe text.txt
client.py > ...
1 import socket
2 import ssl
3 import os
4
5 def send_file_to_server(filename, server_ip, server_port):
6     context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
7     context.check_hostname = False
8     context.verify_mode = ssl.CERT_NONE
9
10    with socket.create_connection((server_ip, server_port)) as sock:
11        with context.wrap_socket(sock, server_hostname=server_ip) as ssl_sock:
12            with open(filename, 'rb') as f:
13                file_data = f.read()
14                ssl_sock.sendall(file_data)
15
16 if __name__ == '__main__':
17     server_ip = '10.1.19.138' # Server IP address (same as server)
18     server_port = 8080 # Port number (same as server)
19
20     while True:
21         filename = input('Enter the filename to send (or type "exit" to quit): ')
22         if filename.lower() == 'exit':
23             break
24         try:
25             send_file_to_server(filename, server_ip, server_port)
26             print(f'Successfully sent {filename} to the server.')
27         except FileNotFoundError:
28             print(f'File {filename} not found. Please try again.')
29
30     print('Client connection terminated.')
31
```

Server.py:

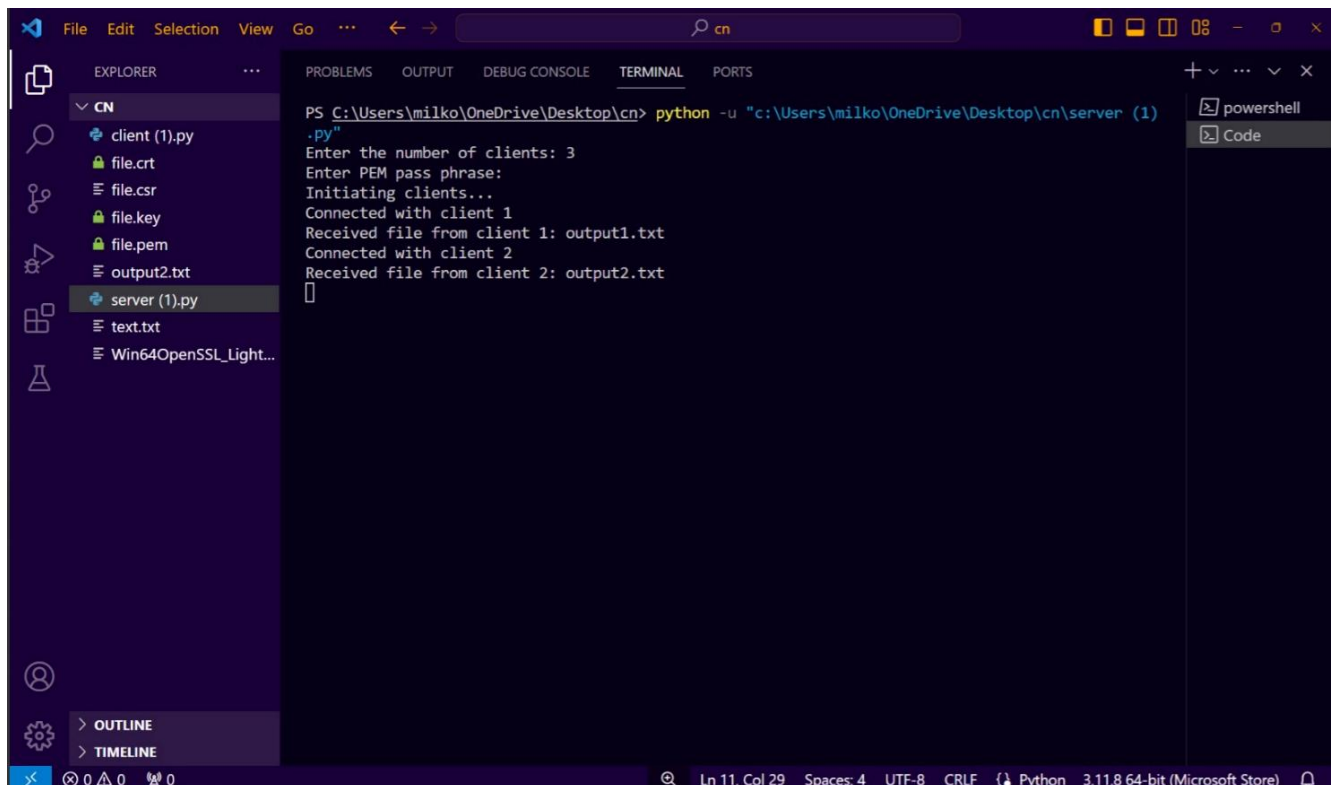
```
1 import socket
2 import ssl
3 import threading
4
5 def handle_client(conn, fileno):
6     try:
7         filename = f'output{fileno}.txt'
8         with open(filename, "wb") as fo:
9             while True:
10                 data = conn.recv(1024)
11                 if not data:
12                     break
13                 fo.write(data)
14         print(f'Received file from client {fileno}: {filename}')
15     finally:
16         conn.close()
17
18 if __name__ == '__main__':
19     host = '10.1.19.138' # Server IP address (localhost for testing)
20     port = 8080         # Port number
21
22     total_clients = int(input('Enter the number of clients: '))
23     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
24
25     context = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
26     context.load_cert_chain('file.crt', 'file.key')
27
28     ssl_sock = context.wrap_socket(sock, server_side=True)
29     ssl_sock.bind((host, port))
30     ssl_sock.listen(total_clients)
31
32     connections = []
33     print('Initiating clients...')
34     try:
35         for i in range(total_clients):
36             conn, _ = ssl_sock.accept()
37             connections.append(conn)
38             print(f'Connected with client {i + 1}')
39             threading.Thread(target=handle_client, args=(conn, i + 1)).start()
40     finally:
41         ssl_sock.close()
42
43     print('All files received successfully!')
```

Text.txt on client side:



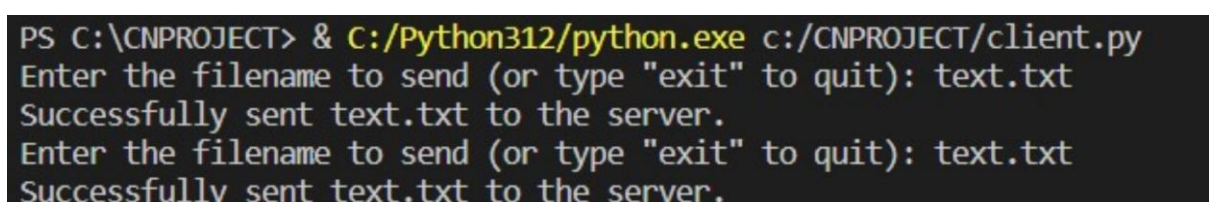
A screenshot of a code editor window. The top bar shows four tabs: 'client.py', 'server.py', 'server.PY', and 'text.txt'. The 'text.txt' tab is active, showing a single line of text: '1 hello world'.

Server running:



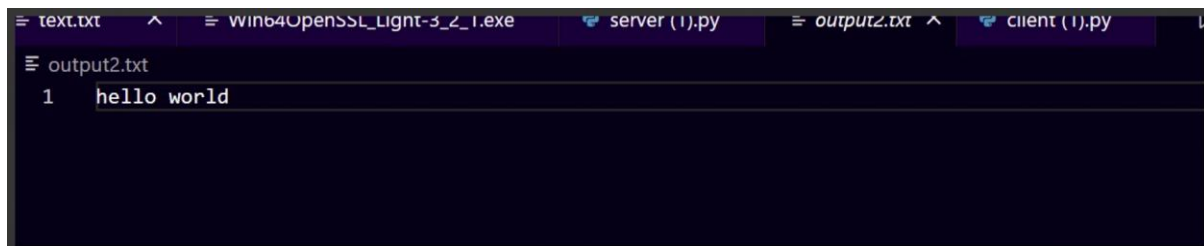
A screenshot of a Visual Studio Code interface. The left sidebar shows the 'EXPLORER' view with a folder named 'CN' containing files: 'client (1).py', 'file.crt', 'file.csr', 'file.key', 'file.pem', 'output2.txt', 'server (1).py', 'text.txt', and 'Win64OpenSSL_Light...'. The main editor area shows the 'TERMINAL' view with the following output:
PS C:\Users\milko\OneDrive\Desktop\cn> python -u "c:\Users\milko\OneDrive\Desktop\cn\server (1).py"
Enter the number of clients: 3
Enter PEM pass phrase:
Initiating clients...
Connected with client 1
Received file from client 1: output1.txt
Connected with client 2
Received file from client 2: output2.txt
[

Client side:



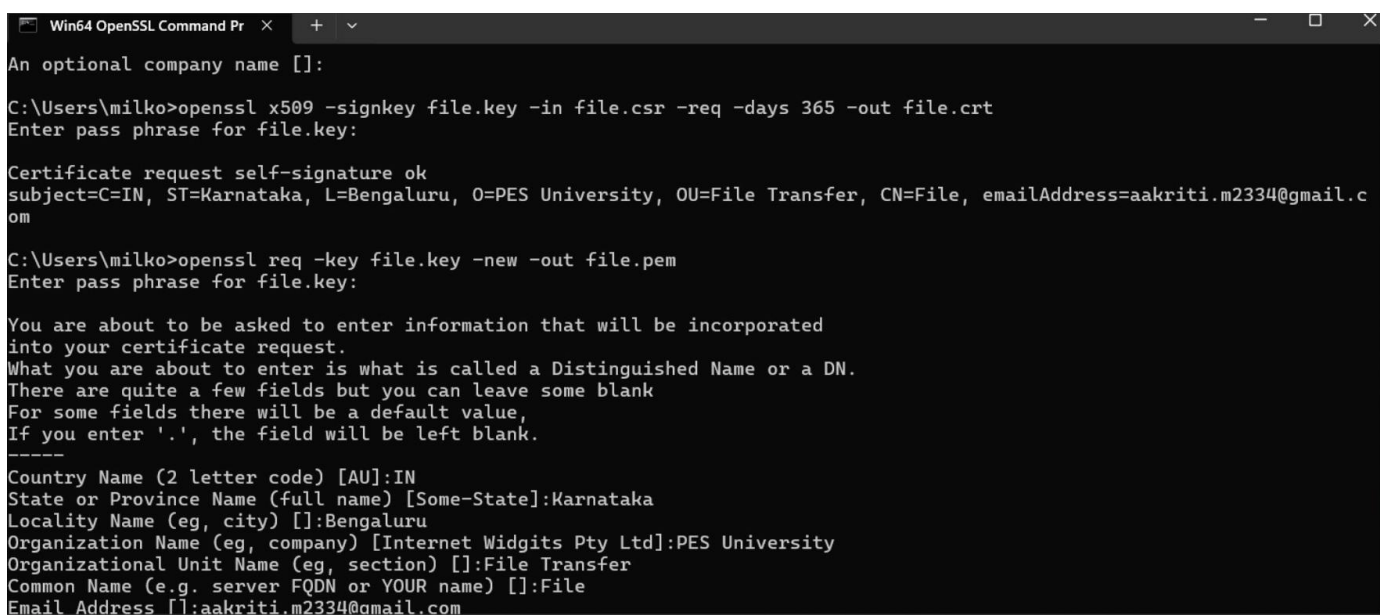
A screenshot of a terminal window showing the following output:
PS C:\CNPROJECT> & C:/Python312/python.exe c:/CNPROJECT/client.py
Enter the filename to send (or type "exit" to quit): text.txt
Successfully sent text.txt to the server.
Enter the filename to send (or type "exit" to quit): text.txt
Successfully sent text.txt to the server.

Text.txt received on server side:



```
text.txt  win64OpenSSL_Light-3_2_1.exe  server (1).py  output2.txt  client (1).py
output2.txt
1  hello world
```

SSL certificate:



```
Win64 OpenSSL Command Pr  +  v
An optional company name []:

C:\Users\milko>openssl x509 -signkey file.key -in file.csr -req -days 365 -out file.crt
Enter pass phrase for file.key:

Certificate request self-signature ok
subject=C=IN, ST=Karnataka, L=Bengaluru, O=PES University, OU=File Transfer, CN=File, emailAddress=aakriti.m2334@gmail.com

C:\Users\milko>openssl req -key file.key -new -out file.pem
Enter pass phrase for file.key:

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Karnataka
Locality Name (eg, city) []:Bengaluru
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PES University
Organizational Unit Name (eg, section) []:File Transfer
Common Name (e.g. server FQDN or YOUR name) []:File
Email Address []:aakriti.m2334@gmail.com
```