Supriya Patil Page no.:____/ JAVA AND PYTHON LAB 14 Date:__/_/__ Aim: Build a concurrent multithreaded file transfer server using threads LOGIC: function () (Socket binds Start-server Host (host, post) SERVER HOST slistening Socket sistening Request Client 1 -- Listening Request aien+2 Tistening i-i-Request server creates Clinet 2 various sockets Client 3 Socket Server socket Clients 30 continued START_SERVER FUNCTION CAMPUS

	Page no.:/
	Date:
	Fanction() handle_client
	SERVER (Client_socket)
	Instance
	Client-Sockettk (Client)
	Server (Client-Sectet2) (Client2)
	fle name to download (3)
	File protent on server (2)
	No Yes C1 Chunkes Sending Gends "Error message" Send data
	Gends "Error message" Send data to Client in chunks!
	HANDLE-CLIENT FUNCTION
CAMPUS	

JAVA AND PYTHON LAB 14

Build a concurrent multithreaded file transfer server using threads

SOURCE CODE:

```
import socket
import threading
import os
def handle client(client socket):
  # Receive the file name from the client
  file name = client socket.recv(1024).decode()
  print(f"Received request for file: {file name}")
  # Check if the file exists
  if os.path.isfile(file name):
    # Open the file in binary mode
    with open(file name, 'rb') as file:
       # Read the file data in chunks and send it to the client
       chunk = file.read(1024)
       while chunk:
         client socket.send(chunk)
         chunk = file.read(1024)
  else:
    # File does not exist
    error message = f"File '{file name}' does not exist"
    client socket.send(error message.encode())
  # Close the client socket
  client socket.close()
def start server(host, port):
  # Create a socket object
  server socket = socket.socket(socket.AF INET, socket.SOCK STREAM)
```

```
# Bind the socket to a specific address and port
  server socket.bind((host, port))
  # Listen for incoming connections
  server socket.listen(5)
  print(f"Server listening on {host}:{port}")
  while True:
     # Accept a client connection
     client socket, client address = server socket.accept()
     print(f"Accepted connection from {client_address[0]}:{client_address[1]}")
     # Start a new thread to handle the client
     client thread = threading.Thread(target=handle client, args=(client socket,))
     client thread.start()
if name__ == '__main__':
  # Set the server host and port
  HOST = '127.0.0.1' # localhost
  PORT = 12345
  # Start the server
  start server(HOST, PORT)
```

OUTPUT

Server listening on 127.0.0.1:12345