3b_CHAT_USING_TCP_SOCKETS

Public

forked from NaliniG007/3b_CHAT_USING_TCP_SOCKETS

1 Branch 0 Tags Go to file t Go to file Add file Code

This branch is 3 commits ahead of NaliniG007/3b_CHAT_USING_TCP_SOCKETS:main .

Contribute Sync fork

ganeshprabhu2005 Update README.md

dc60f5b · now

README.md	Update README.md	now

README

3b.CREATION FOR CHAT USING TCP SOCKETS

NAME:GANESH PRABHU J

REG NO:212223220023

AIM

To write a python program for creating Chat using TCP Sockets Links.

ALGORITHM:

1. Import the necessary modules in python
2. Create a socket connection to using the socket module.
3. Send message to the client and receive the message from the client using the Socket module in server

About

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

4. Send and receive the message using the send function in socket.

PROGRAM

```
import socket

# Create a socket object
server_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# Get local machine name (can also use 'localhost' or
an IP address)
host = socket.gethostname()  # or '127.0.0.1' for
localhost
port = 12345  # Choose any unused port

# Bind the socket to the address and port
server_socket.bind((host, port))

# Start listening for incoming connections (can queue
up to 5)
server_socket.listen(5)
print(f"Server listening on {host}:{port}")

# Accept the connection from the client
client_socket, client_address = server_socket.accept()
print(f"Connection from {client_address} has been
established!")

# Chat loop to send and receive messages
while True:
    # Receive message from client
    message_from_client =
client_socket.recv(1024).decode('ascii')
    print(f"Client: {message_from_client}")

    # End chat if the client sends 'bye'
    if message_from_client.lower() == 'bye':
        print("Client has left the chat.")
        break

    # Send a message to the client
    message_to_client = input("Server: ")

client_socket.send(message_to_client.encode('ascii'))

    # End chat if the server sends 'bye'
    if message_to_client.lower() == 'bye':
        print("Closing connection...")
        break

# Close the connection
```

```
client_socket.close()
server_socket.close()
```

```
import socket

# Create a socket object
client_socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# Get the local machine name
host = socket.gethostname()  # or '127.0.0.1' for
localhost
port = 12345  # Port should be the same as the
server's

# Connect to the server
client_socket.connect((host, port))
print(f"Connected to server at {host}:{port}")

# Chat loop to send and receive messages
while True:
    # Send a message to the server
    message_to_server = input("Client: ")

client_socket.send(message_to_server.encode('ascii'))

    # End chat if the client sends 'bye'
    if message_to_server.lower() == 'bye':
        print("Exiting chat...")
        break

    # Receive a message from the server
    message_from_server =
client_socket.recv(1024).decode('ascii')
    print(f"Server: {message_from_server}")

    # End chat if the server sends 'bye'
    if message_from_server.lower() == 'bye':
        print("Server has closed the chat.")
        break

# Close the connection
client_socket.close()
```

OUPUT

```
Connection from ('192.168.1.2', 52405) has been established!
Client: hi , server!
Server: hello client.
Client: how are you doing?
Server: i am well pretty good , how abt you?
Client: fine here , thanks
Server: welcome
Server: i am well pretty good , how abt you?
Client: fine here , thanks
Server: welcome
Client: bye
```

```
Client: hi , server!
Server: hello client.
Client: how are you doing?
Server: i am well pretty good , how abt you?
Client: fine here , thanks
Server: welcome
```