Name	Manish Shashikant Jadhav
UID	2023301005
Subject	Computer Communication and Networks (CCN)
Experiment No.	2
Aim	Network Socket Programming

Code for Part 1:	**Problem Statement:** Implement the following rudimentary string processing application using connection oriented client-server programming. Some guidelines for the implementation are as follows. The client will send a textual paragraph terminated by '\n' to the server (assume that in the paragraph, '.' appears only at the end of sentences and nowhere else). The server will compute the number of characters, number of words, and number of sentences in the paragraph, and send these numbers back to the client. The client will print these numbers on the screen.
	
	 1. **Server:**
	
	 ``python
	import socket
	
	HOST = '127.0.0.1' # Standard loopback interface address (localhost)
	PORT = 65432 # Port to listen on (non-privileged ports are > 1023)
	
	with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
	    s.bind((HOST, PORT))
	    s.listen()
	    conn, addr = s.accept()
	    with conn:
	        print('Connected by', addr)
	        while True:
	            data = conn.recv(1024)
	            if not data:
	                break
	
	            paragraph = data.decode('utf-8')
	``

```python
            num_chars = len(paragraph)
            num_words = len(paragraph.split())
            num_sentences = paragraph.count('.')

            results = f"Characters: {num_chars}\nWords:
{num_words}\nSentences: {num_sentences}"
            conn.sendall(results.encode('utf-8'))
```

2. **Client Side:**

```python
import socket

HOST = '127.0.0.1'   # The server's hostname or IP address
PORT = 65432         # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    paragraph = input("Enter a paragraph: ") + '\n'
    s.sendall(paragraph.encode('utf-8'))
    results = s.recv(1024).decode('utf-8')
    print(results)
```

Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python server1.py
Connected by ('127.0.0.1', 50803)
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code>
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python client1.py
Enter a paragraph: Hello Myelf Manish Jadhav
Characters: 26
Words: 4
Sentences: 0
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> |
```

Code for Part 2:	**Problem Statement:** Make it concurrent so that it can serve multiple clients at a time. (Multiple clients on multiple terminals and single server terminals). **Server(Updated):**

```python
import socket
import threading


HOST = '127.0.0.1'
PORT = 65432


def handle_client(conn, addr):
    print(f"Connected by {addr}")
    while True:
        data = conn.recv(1024).decode('utf-8')
        if not data:
            break

        paragraph = data
        num_chars = len(paragraph)
        num_words = len(paragraph.split())
        num_sentences = paragraph.count('.')

        results = f"Characters: {num_chars}\nWords:
{num_words}\nSentences: {num_sentences}"
        conn.sendall(results.encode('utf-8'))
    conn.close()


def start_server():
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.bind((HOST, PORT))
        s.listen()
        print(f"Server listening on port {PORT}")
        while True:
            conn, addr = s.accept()
```

```python
            thread = threading.Thread(target=handle_client,
args=(conn, addr))
            thread.start()


if __name__ == '__main__':
    start_server()
```
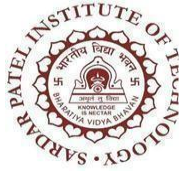
Output:	**Server (connected to 3 Clients):**
| | ```
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python server2.py
Server listening on port 65432
Connected by ('127.0.0.1', 51163)
Connected by ('127.0.0.1', 51516)
Connected by ('127.0.0.1', 51519)
``` |
| | **Client 1:** |
| | ```
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python client1.py
Enter a paragraph: Manish Jadhav from Raigad
Characters: 26
Words: 4
Sentences: 0
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code>
``` |

Client 2:

```
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python client1.py
Enter a paragraph: Mayur Solankar from Vashind
Characters: 28
Words: 4
Sentences: 0
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code>
```

Client 3:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code> python client1.py
Enter a paragraph: Nishant Satere from andheri
Characters: 28
Words: 4
Sentences: 0
PS D:\Manish\SPIT\4th SEM\CCN\LAB\Exp2\Code>
```

Code for Part 3:	1. **Server Side:**

```python
import socket
import threading

HOST = '127.0.0.1'  # Standard loopback interface address
(localhost)
PORT = 65432        # Port to listen on

clients = []
client_names = {}

def handle_client(conn, addr, name):
    global clients, client_names
    print(f'{name} connected from {addr}')
    broadcast(f'{name} has joined the chat!', name)
    while True:
        try:
            data = conn.recv(1024).decode()
```

```python
            if not data:
                break
            message = f'{name}: {data}'
            broadcast(message, name)
        except ConnectionError:
            print(f'{name} disconnected unexpectedly')
            broadcast(f'{name} has left the chat!', name)
            clients.remove(conn)
            del client_names[conn]
            break
    conn.close()

def broadcast(message, sender):
    for client in clients:
        if client != sender:
            try:
                client.sendall(message.encode())
            except ConnectionError:
                print(f'Error sending message to
{client_names[client]}')

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    print(f'Server listening on {HOST}:{PORT}')
    while True:
        conn, addr = s.accept()
        name = conn.recv(1024).decode()
        client_names[conn] = name
        clients.append(conn)
        client_thread = threading.Thread(target=handle_client,
args=(conn, addr, name))
        client_thread.start()
```
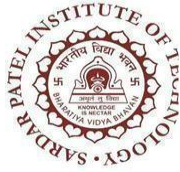
2. Client Side:

```python
import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    name = input("Enter your name: ")
    s.sendall(name.encode())

    while True:
        message = input()
        if message == '/quit':
            break
        s.sendall(message.encode())
        data = s.recv(1024).decode()
        print(data)
```
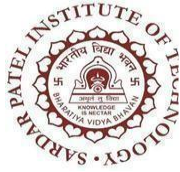
Output:

1. Server:

```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>cd desktop

C:\Users\USER\Desktop>cd chat app

C:\Users\USER\Desktop\chat app>python 3server.py
Server listening on 127.0.0.1:65432
ManishJ connected from ('127.0.0.1', 54464)
AnishB connected from ('127.0.0.1', 54486)
```

2. **Client 2:**

```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>cd desktop

C:\Users\USER\Desktop>cd chat app

C:\Users\USER\Desktop\chat app>python 3server.py
Server listening on 127.0.0.1:65432
ManishJ connected from ('127.0.0.1', 54464)
AnishB connected from ('127.0.0.1', 54486)
```

3. **Client 3:**

```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>cd desktop

C:\Users\USER\Desktop>cd chat app

C:\Users\USER\Desktop\chat app>python 3server.py
Server listening on 127.0.0.1:65432
ManishJ connected from ('127.0.0.1', 54464)
AnishB connected from ('127.0.0.1', 54486)
```

Conclusion	Hence by completing this experiment I came to know about Network Socket Programming