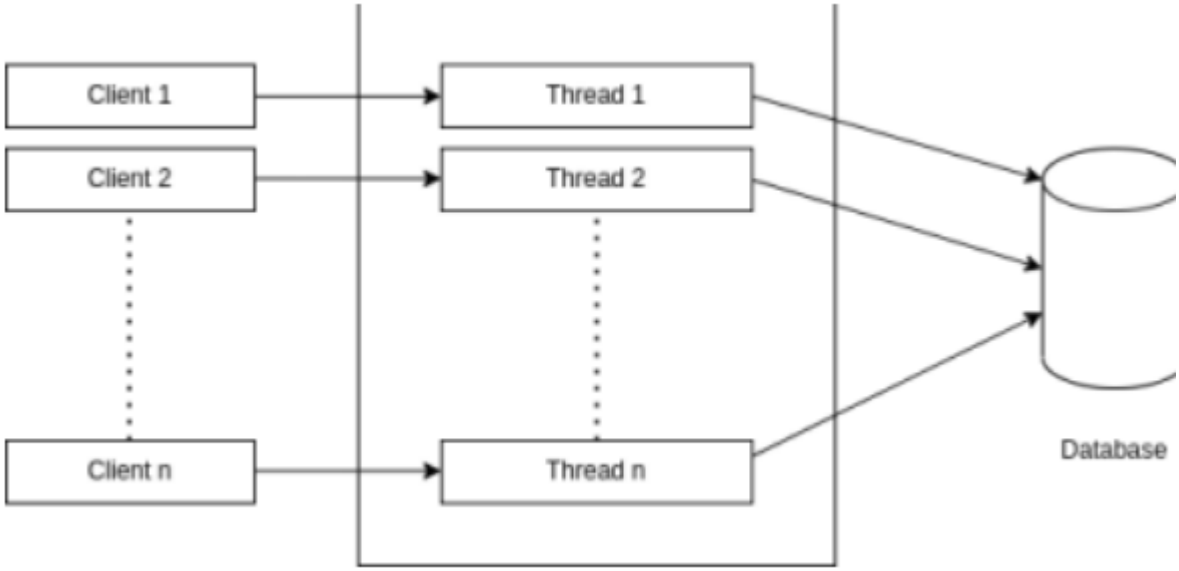
BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to Mumbai University)
Department Of Computer Engineering

Name	Mayur Solankar, Manish Jadhav, Vishesh Savani, Shreyansh Salvi
UID	2023301018, 2023301005, 2022300100, 2022300091
Subject	Distributed Computing
Experiment No.	3
Project title	Social Media System
Problem Statement	Implementation of Multithreading in Distributed System
Objectives	The objective is to use multithreading to increase the real-time interactivity and responsiveness of social media platforms for actions like chatting, liking, commenting, and posting.
Theory	**What is Multithreading?** A distributed system involves many computers or nodes working together to achieve a common goal. Multithreading means that several threads are executing simultaneously in one process. The social media chat feature allows for real time messaging and the most readable messages. Each conversation runs on a separate thread, allowing multiple users to send and receive messages simultaneously. Social media platforms manage various user interactions such as liking, commenting, sharing and responding to posts. Allowing users to see updates in their posts and comments. Notifications must be sent immediately when a user receives likes, comments or messages. It ensures that notifications are delivered without delay. Users upload photos and videos to share with others. **Benefits of Multithreading in Distributed Systems Concurrency:** **Multithreading** allows a distributed system to execute multiple tasks concurrently, improving overall system efficiency and performance. Multithreading enables distributed systems to remain responsive to external requests and events by handling tasks in parallel.



Code:

1. **Client Side**:

```
import socket
import threading

class User:
    def _init_(self, username):
        self.username = username
        self.friends = []

    def add_friend(self, friend_username):
        self.friends.append(friend_username)

class SocialMediaPlatform:
    def _init_(self):
        self.posts = []
        self.users = {}
        self.lock = threading.Lock()

    def add_user(self, username):
        if username not in self.users:
            self.users[username] = User(username)
            return f"User '{username}' created successfully."
        return f"User '{username}' already exists."
```

```
    def add_friend(self, username, friend_username):
        user = self.users.get(username)
        friend = self.users.get(friend_username)
        if user and friend:
            user.add_friend(friend_username)
            friend.add_friend(username)
            return f"Friend '{friend_username}' added to '{username}'."
        return "User or friend not found."

    def post(self, username, message):
        with self.lock:
            post_index = len(self.posts)
            post = f"{username}: {message}"
            self.posts.append(post)
            return f"Post created successfully. Index: {post_index}\nPost Content:
{post}"

    def comment(self, username, post_index, comment):
        with self.lock:
            if 0 <= post_index < len(self.posts):
                original_post = self.posts[post_index]
                updated_post = f"{original_post}\n- Comment by {username}:
{comment}"
                self.posts[post_index] = updated_post
                return f"Comment added successfully.\nComment Content: - Comment by
{username}: {comment}"
            return "Invalid post index."

    def get_posts(self):
        return self.posts

def handle_client(client_socket, platform):
    while True:
        data = client_socket.recv(1024).decode()
        if not data:
            break

        # Parse the client's request
        parts = data.split()
        response = ""
```

```
            if parts[0] == '1':
                # Create User
                username = parts[1]
                response = platform.add_user(username)
            elif parts[0] == '2':
                # Add Friend
                username = parts[1]
                friend_username = parts[2]
                response = platform.add_friend(username, friend_username)
            elif parts[0] == '3':
                # Make a Post
                username = parts[1]
                message = ' '.join(parts[2:])
                response = platform.post(username, message)
            elif parts[0] == '4':
                # Comment on a Post
                username = parts[1]
                post_index = int(parts[2])
                comment = ' '.join(parts[3:])
                response = platform.comment(username, post_index, comment)
            elif parts[0] == '5':
                # Exit
                # Send all posts and summary
                posts = platform.get_posts()
                num_posts = len(posts)
                num_comments = sum(post.count('\n- Comment by ') for post in posts)
                response = f"Posts Summary:\nNumber of posts: {num_posts}\nNumber of
comments: {num_comments}\n"
                for post in posts:
                    response += f"{post}\n"
                client_socket.send(response.encode())
                break
            else:
                response = "Invalid choice."

            client_socket.send(response.encode())

        client_socket.close()

    def main():
```

```
    server_host = '192.168.0.103'  # Use the IP address of 'enp3s0' interface
    server_port = 12345
    platform = SocialMediaPlatform()

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((server_host, server_port))
    server_socket.listen(5)  # Listen for up to 5 incoming connections

    print(f"Server is listening on {server_host}:{server_port}")

    while True:
        client_socket, addr = server_socket.accept()
        client_handler = threading.Thread(target=handle_client, args=(client_socket,
platform))
        client_handler.start()

main()
```

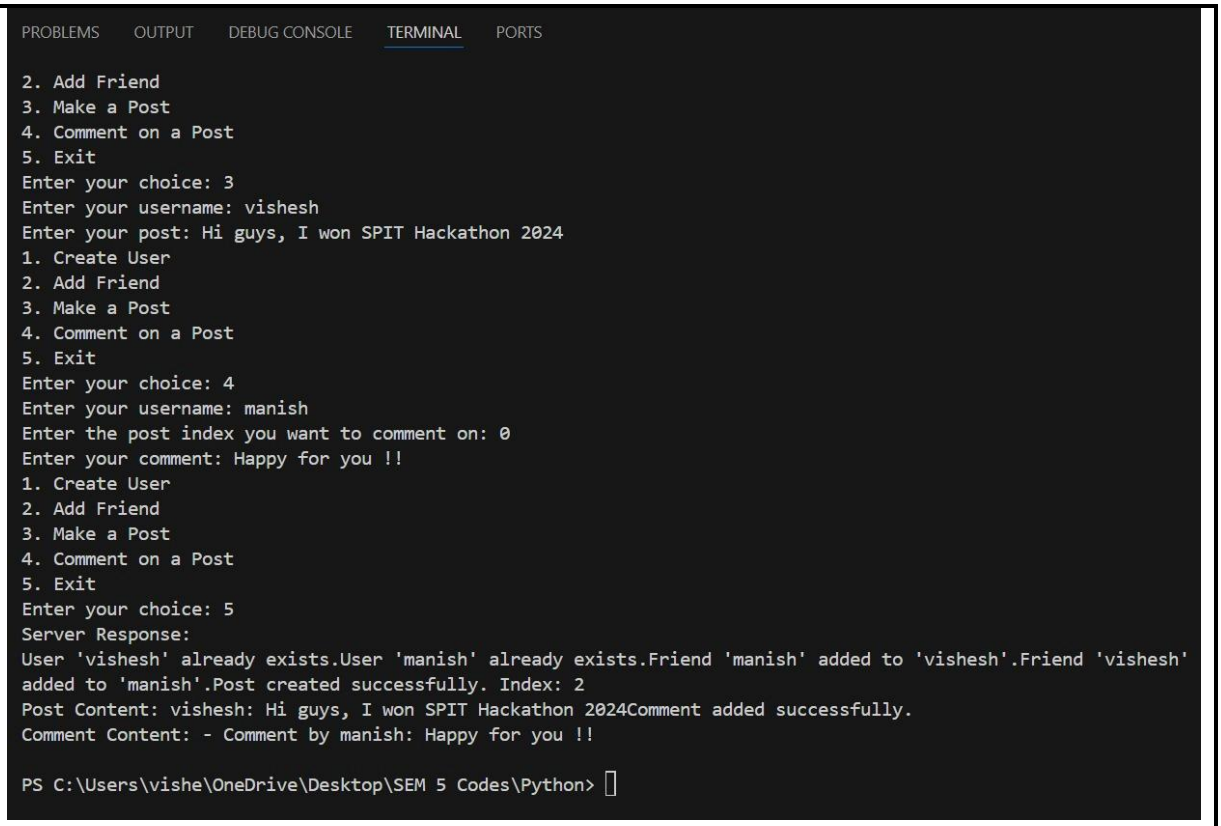
2. Server Side:

```
import socket

def main():
    server_host = '192.168.0.103'  # Use the IP address of 'enp3s0' interface
    server_port = 12345

    # Set up a socket to communicate with the server
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((server_host, server_port))

    while True:
        print("1. Create User")
        print("2. Add Friend")
        print("3. Make a Post")
        print("4. Comment on a Post")
        print("5. Exit")

        choice = int(input("Enter your choice: "))

        if choice == 1:
            username = input("Enter your username: ")
            client_socket.send(f"1 {username}".encode())
```

```
        elif choice == 2:
            username = input("Enter your username: ")
            friend_username = input("Enter your friend's username: ")
            client_socket.send(f"2 {username} {friend_username}".encode())

        elif choice == 3:
            username = input("Enter your username: ")
            message = input("Enter your post: ")
            client_socket.send(f"3 {username} {message}".encode())

        elif choice == 4:
            username = input("Enter your username: ")
            post_index = int(input("Enter the post index you want to comment on: "))
            comment = input("Enter your comment: ")
            client_socket.send(f"4 {username} {post_index} {comment}".encode())

        elif choice == 5:
            client_socket.send("5".encode())
            # Receive and print all posts and summary
            data = client_socket.recv(4096).decode()
            print("Server Response:")
            print(data)
            break

        else:
            print("Invalid choice. Please try again.")

    # Close the client socket
    client_socket.close()

main()
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\vishe\OneDrive\Desktop\SEM 5 Codes\Python> Python client.py
1. Create User
2. Add Friend
3. Make a Post
4. Comment on a Post
5. Exit
Enter your choice: 1
Enter your username: vishesh
1. Create User
2. Add Friend
3. Make a Post
4. Comment on a Post
5. Exit
Enter your choice: 1
Enter your username: manish
1. Create User
2. Add Friend
3. Make a Post
4. Comment on a Post
5. Exit
Enter your choice: 2
Enter your username: vishesh
Enter your friend's username: manish
1. Create User
2. Add Friend
3. Make a Post
4. Comment on a Post
5. Exit
Enter your choice: 2
Enter your username: manish
Enter your friend's username: vishesh
1. Create User
2. Add Friend
```



Conclusion:	Hence by completing we came to about implementation of Multithreading in Distributed Systems.