

```

#Client
import socket
import json
import time

SERVER_HOST = 'localhost'
SERVER_PORT = 5000
BUFSIZE = 1024
#Defined Functions Used
def register_user(sock):
    username = input("Enter a new username: ")
    password = input("Enter a new password: ")
    request = {"action": "register", "username": username, "password": password}
    sock.send(json.dumps(request).encode())
    response = json.loads(sock.recv(BUFSIZE).decode())
    print(response.get("message", "For some reason an error has occurred while attempting to register!"))

def login_user(sock):
    username = input("Enter username: ")
    password = input("Enter password: ")
    request = {"action": "login", "username": username, "password": password}
    sock.send(json.dumps(request).encode())
    response = json.loads(sock.recv(BUFSIZE).decode())
    if response["status"] == "success":
        print("Login successful!")
        return username
    else:
        print("Login unsuccessful!")
        return None

def cast_vote(sock, username):
    candidate = input("Who or what are you voting for?: ")
    request = {"action": "vote", "username": username, "candidate": candidate}
    sock.send(json.dumps(request).encode())
    response = json.loads(sock.recv(BUFSIZE).decode())
    print(response.get("message", "For some reason an error has occurred while casting your vote!"))

def view_tally(sock):
    try:
        request = {"action": "tally"}
        sock.send(json.dumps(request).encode())
        response = json.loads(sock.recv(BUFSIZE).decode())
        print("\nVote Tally:")
        for candidate, votes in response.get("tally", {}).items():
            print(f"{candidate}: {votes} votes")
    except ConnectionAbortedError:
        print("Connection to the server was lost. Please try again later.")
    except Exception as e:
        print(f"Error viewing tally: {str(e)}")

RETRY_INTERVAL = 10

def check_server_connection():
    max_retries = 3
    attempts = 0

    while attempts < max_retries:

```

```

    try:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.connect((SERVER_HOST, SERVER_PORT))
        print("Successfully connected to the voting server.")
        return sock
    except ConnectionRefusedError:
        attempts += 1
        print(f"Attempt {attempts}/{max_retries}. Server not responsive.
Retrying in {RETRY_INTERVAL} seconds...")
        time.sleep(RETRY_INTERVAL)

    print("Sorry, the server is unresponsive at the moment. Please try again
later...")
    return None

def main():
    sock = check_server_connection()
    if not sock:
        print("A successful connection to the server could not be established.
Please try again later.")
        return

    username = None
    while True:
        try:
            print("\nMenu:\n1. Register\n2. Login\n3. Vote\n4. View Tally\n5.
Exit")
            choice = input("Enter a corresponding number to choose action: ")

            if choice == "1":
                register_user(sock)
            elif choice == "2":
                username = login_user(sock)
            elif choice == "3":
                if username:
                    cast_vote(sock, username)
                else:
                    print("Please login before attempting to vote.")
            elif choice == "4":
                view_tally(sock)
            elif choice == "5":
                print("Exiting...")
                break
            else:
                print("Invalid choice! Please try again.")
        except (ConnectionResetError, ConnectionAbortedError, OSError):
            print("Sorry, the connection to the server has been lost. Please try
reconnecting.")
            sock = check_server_connection()
            if not sock:
                print("Sorry, a connection to the server could not be
reestablished. Please try again later.")
                break

if __name__ == "__main__":
    main()

```