Title: Socket Programming and Wireshark Verification
Due: 10/27 by Midnight

Group Organization and Rules:

- Each team will consist of 5 members.
- A group leader will be selected after the first meeting. The elected group leader must report the first group meeting via Blackboard (BB) within the week of 10/7. All group communication will be coordinated by the group leader.
- **Group Leader Responsibilities**:
    - Assign tasks needed to complete the assignment in a fair and reasonable manner. Each member must understand their task and agree to the assignment.
    - Ensure a consistent group standard so integration is seamless. The group leader has the authority to reject substandard work.
- Group meetings should be held on a weekly basis, preferably in person, though Zoom meetings are acceptable.
- If a member misses more than two meetings without a legitimate reason, they will no longer be part of the team. The removed member will be required to complete the entire assignment or project individually.
- **NO FREE RIDES**: All members must contribute equally.
- There will be **no individual grading**. All members of the group will share the assigned grade.
- The minimum number of active group members is 3. If there are 3 or more members, the assignment will not be affected by missing team members.

Homework Requirements:

1. **Implementation of one of two socket types**:
    - Either a traditional socket (Unix-based) or Windows socket (Winsock), with a **Graphical User Interface (GUI)**.
    - **GUI Specifications**:
        - **Client Side (Windows)**:
            1. Server IP address (editable).
            2. Client IP address (display only).
            3. Message received from the server.
            4. Message to send to the server (host).
            5. Error message (e.g., “Connection closed!”).
        - **Client Buttons**:
            1. Clear
            2. Send
            3. Quit
        - **Server Side (Windows)**:
            1. Server IP address (display only).
            2. Client IP address (displayed when a connection is made).
            3. Message received from the client.
            4. Message to send to the client.

        5. Error message.
    - **Server Buttons**:
        1. Clear
        2. Send
        3. Quit

2. **Explanation of the API Used**:
    - Reference figures 3.7 and 3.8 for a summary of the socket API and the sequence of socket function calls by both the client and server.
    - Map your socket API functions to the sequence shown in the figures. Explain what each API function does and the sequence of function calls.
3. **Wireshark Usage**:
    - Prepare a communication scenario.
    - Start Wireshark on server side and client side (need two machines)
    - Run the client and server according to the user manual prepared.
    - Capture packets from both machines
    - Filter the packets by IP address so only the packets used by the communication are displayed.
    - Take the screenshots of the above screen from both machines.

Deliverables (What to Submit):

4. **Description of the Socket Implementation**:
    - Type and function of the socket your group is implementing.
    - How it works (include a user manual with screenshots).
    - GUI screenshots.
    - Source code for both client and server.
5. **Explanation of the API**:
    - Provide an explanation of the socket API used, following the description in Item 2 above.
6. **Wireshark Screenshots and Analysis**:
    - Include screenshots from Wireshark and provide an explanation of each screen – which part is showing the socket communication.
7. Task assignment by group member (optional but recommended)