National University of Sciences & Technology
School of Electrical Engineering and Computer Science
Department of Computing

EE 353: Computer Networks (3+1): BESE-15AB Fall 2025

Assignment: 01	
CLO-3: Design network solutions to ensure reliability, overcome connectivity issues, and optimize performance.	
Maximum Marks: 50	Due Date: 2nd November 2025 11:55 pm on LMS

Instructions:

This is an individual assignment. Write your name, section and CMSID in your details. Upload the complete document to the course LMS site by the deadline. Please avoid plagiarism; any such case would result in award zero marks both to the "sharer" and the "acquirer". You may note that submitted assignment would be checked through turnitin that would check similarly with not only internet resources but also with other submitted assignments.

Task: You are required to design and implement a chat application in Python using socket programming. The application must support multiple clients simultaneously and allow both broadcast (group chat) and unicast (private message) communication between users.

Your application should consist of a Server Program and a Client Program:

Server Responsibilities

- Maintain a list of all active clients with unique usernames.
- Handle new client connections, user registration, and disconnections gracefully.
- Support broadcast messages, where a message sent by one client is delivered to all other connected clients.
- Support unicast messages, where a message is sent only to a specific recipient using the format '@username message'.
- Notify all users when someone joins or leaves the chat.

Client Responsibilities

- Connect to the server and allow the user to choose a unique username.
- Allow users to send and receive both broadcast and unicast messages.
- Display incoming messages appropriately (e.g., show private messages clearly).
- Allow users to exit the chat gracefully with a specific command (e.g., /quit).

Functional Requirements

- Server handles multiple clients concurrently (e.g., using threads).
- Implements a publish–subscribe mechanism for broadcasting.

- Sends private messages only to the intended recipient.
- Client accepts input from the user and sends to server.
- Displays all received messages clearly.
- Uses a special syntax for private messages (e.g., @username Hello).
- Should handle invalid inputs and disconnections without crashing.
- Use clear code structure with proper comments and function separation.

Grading Criteria

You will be graded based on the following grading criteria:

- Basic Client–Server Communication
- Broadcast Functionality
- Unicast Functionality
- Multiple Client Handling
- Join/Leave Notifications
- Error Handling & Robustness
- Code Quality & Comments
- User Experience