1. Can we use a third folder like shared/ for common code?

Yes, you may use an additional folder like shared/ to store common code used by both client and server. Just make sure your Makefile handles the build correctly and produces the required executables: chatserver and chatclient.

2. What happens if a room does not exist when using /join <room_name>?

When /join <room_name> is used, if the room does not exist, the server should automatically create it. You do not need to define a default room list.

3. Should /whisper and /broadcast messages be visible on the client side?

Yes. Messages received via /whisper or /broadcast must be displayed in the receiving client's terminal. Users must be able to see incoming messages, not just the ones they send.

4. Should the server create a dedicated thread for each client? When should it terminate?

Yes, a dedicated thread should be created for each client on the server side. The thread should terminate when the client disconnects or sends /exit. Using pthread_detach() is the recommended approach.

5. is it allowed to use pthread_detach()?

Yes, pthread_detach() is allowed and recommended in this project. It is preferred over pthread_join() for better resource management.

6. shows "possibly lost" warnings when using

If the "possibly lost" warnings are caused by pthread library internals (not your own malloc calls) and you're using pthread_detach(), this is acceptable. Just document it clearly in your report.

7. How can the client both wait for user input and receive messages at the same time?

The client must use two separate threads: one for reading user input from the terminal, and another for listening to server messages. You do not need a second socket connection; the same one can be shared.

8. When a file is sent via /sendfile, should the receiver see the file's content?

It is not required to display the file contents in the client's terminal. Simply notifying the client that the file has been received is enough (e.g., File received from user X).

9. Should clients be blocked if the file upload queue is full?

Yes. If the file upload queue (e.g., max 5 items) is full, the client can either wait and retry later (e.g., with a short delay), or receive a message like "Queue full, please try again later." Either approach is acceptable as long as it is clearly explained in your report.

10. Can file transfer be simulated via messages only?

Yes, you can simulate file transfer using message-based protocols (e.g., send file name, size, and a confirmation). It is not mandatory to transfer the actual binary contents unless you choose to implement it.