Requirement ID: 2.1
Description: Define database schema for users, messages, and rooms
Artifact: The database schema should contain information about users, messages, and rooms in separate tables. Each entry in the respective tables should be linked such that relevant items are easily accessible. For example messages should be connected to rooms and users in order to track where a message was sent and who sent the message. The database schema should also account for any other potentially useful information about users, messages, etc. As an extra we can add tables for server events such as moderation logs, user logins, etc.

Requirement ID: 2.2
Description: Create backend API structure (FastApi)
Artifact: FastApi will be used to create and manage websocket connections along with the REST endpoints. We need to add endpoints for user creation/deletion/update actions and room creation/deletion actions as well. The websocket connection will handle live message updates for the currently active room. Both of these will be orchestrated by FastApi and will also interact with the database to persist messages and keep track of user and room state.

Requirement ID: 3
Description: Implement WebSocket server for real-time messaging
Artifact: The websocket server in particular will handle live updates to messages, user joins/leaves, and typing indications (extra feature). There will be a client-side implementation which should receive messages and update the UI accordingly. The server side implementation will handle updating global state (database) and filtering and organizing messages.

Requirement ID: 3.3
Description: Implement broadcast of messages to all connected clients
Artifact: This is a part of the websocket implementation. When a user sends a message from a client-side connection the server should receive the message and process it by filtering any profane messages, and then broadcasting to all the clients. Back on the client side the client should only see the message if the broadcasted message is broadcasted to the room they are in.

Requirement ID: 4

Description: Develop frontend chat interface

Artifact: The UI should have at least a main window for the current chat room and a side bar to select the room they are currently in. A user list could also be displayed to the side or in a hidden menu. The main window should have a list of all the messages along with usernames and timestamps, along with an input box for the current user to type in and send messages. The side bar should list all open rooms and each room name should be clickable so that users can select which room they want to be in. The user list would be displayed only and should only list users that are in the currently selected room.