## Chat Application

Write a chat application consisting of server and client with the following functionality -

### **Basic**

* Server has a capacity of handling X (configurable) number of simultaneous clients
* If X+1th client tries to connect to server, it should be rejected
* Any client that sends a msg to the server should be broadcasted to all the connected clients
* Server should keep a history of all the messages received along with the senders and timestamps
* Any new client connecting to the server should receive previous N (configurable) messages on successfully connecting
* Entry and exit message should also be broadcast to all clients
* No 2 clients should be able to simultaneously connect with same username
* Ability for a client to retrieve list of all connected clients
* Ability to send a private message from one client to another
* Ability on server to see list of all clients and complete chat history

### **Advanced**

* Topics
  1. Ability to define/change chatroom topic by clients
  2. Topic should also be broadcast on client connection
  3. Ability for clients to see/retrieve topic
* Admin users
  1. Ability on server to define (or predefine) certain users as admins
  2. Admins should be able to kick any user from chat
  3. Admins should be able to mute any user
     1. Private messages will still work even if muted
     2. Functionality to mute for a certain time period (eg 5 mins) after which user will be auto-unmuted
     3. Muted users will still be able to receive the group messages
     4. Admin can also explicitly unmute any user
     5. On trying to send message, the muted user should be shown till when he is muted
  4. Config to only allow admins to change topic
  5. Ability for any user to get who are the admin users
  6. Ability for admin users to promote any normal user to admin
* Offline messages
  1. In case of private message, if the receiver is not connected, message should be stored on server and sent to client when he connects again
  2. Msg should be sent to sender that receiver is offline and will receive msg later

A sample conversation might look like this -

[Timestamp] \*User1: has entered the room\*  
[Timestamp] User1: Anybody there?  
[Timestamp] \*User2: has entered the room\*  
[Timestamp] User2: Hi, this is User2 here!  
[Timestamp] User2: Nice weather today! Time to learn some erlang!  
[Timestamp] \*User3 has entered the room\*  
[Timestamp] User3: Hi guys, what's up! Let us make a nice chat application!  
[Timestamp] User1: Sure. Let's rock!  
[Timestamp] [Private] User2: This looks like fun! // only sent to User2  
[Timestamp] \*User3 has left the room\*  
[Timestamp] \*User4 has entered the room\* // User4 would have received last 3 messages on connecting  
[Timestamp] User3: Let me know if you need any help

Task 1: Make this application using basic erlang

Task 2: Use mnesia database to store message history, offline messages etc

Task 3: Convert this to use OTP gen\_server