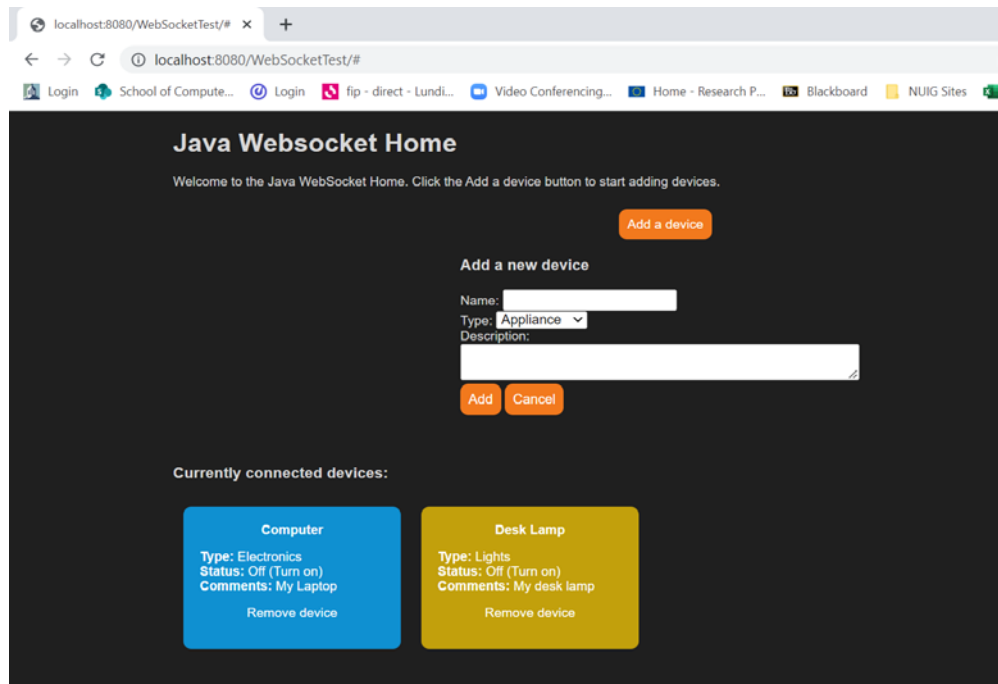


Assignment 6 – WebSocket

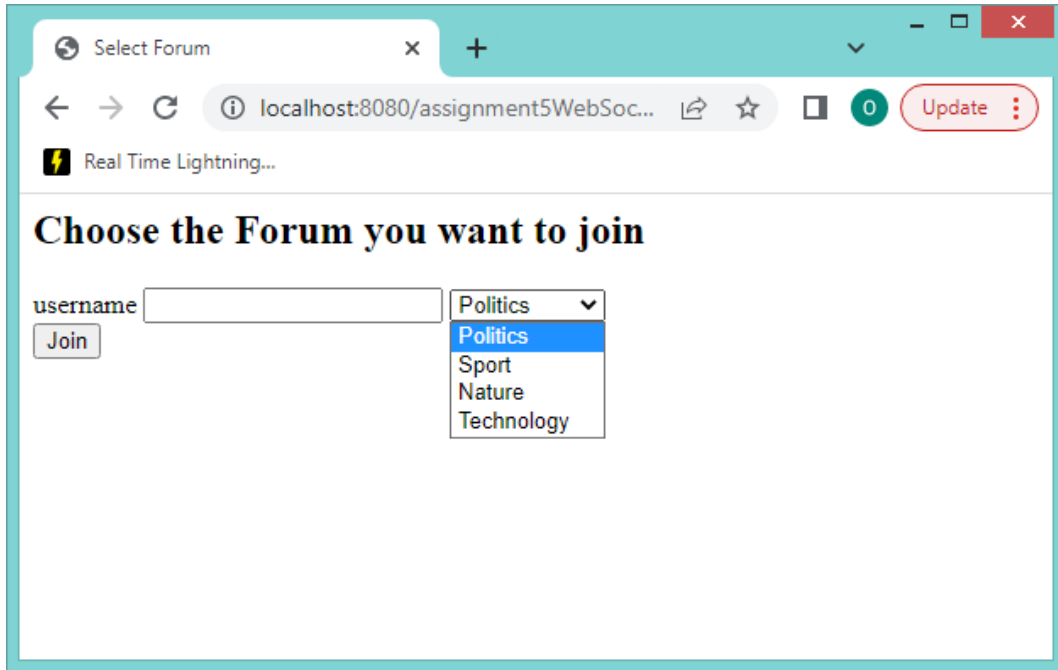
In this assignment, you are asked to create a multi-user chat web application using WebSocket. I started with the WebSocketTest application I showed in the lecture:



I was able to reuse a lot of the structure there to build it, particularly on the server side, such as the session handler class. And of course the initial code to handle the 4 event types are already there too.

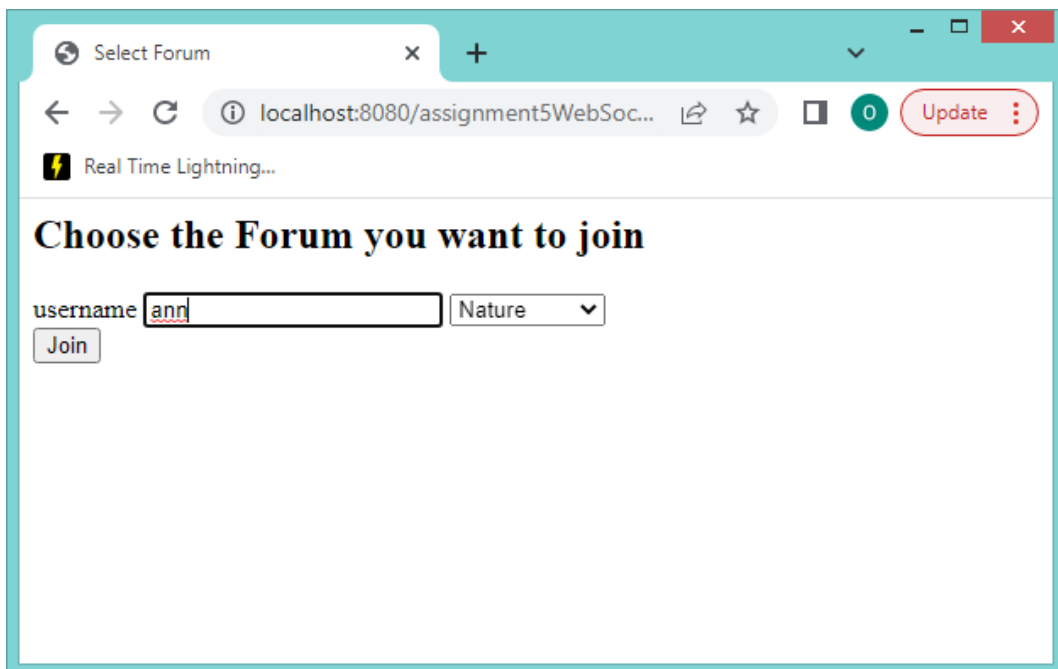
I do not expect you to include any styling / CSS, unless you feel like it.

The application is based around forums. When the user opens the app (index.html) in the browser, they are presented with a choice of forums (these can be fixed / hardcoded).



When the user chooses a forum and clicks on the 'Join' button, the javascript on the page should request to open a websocket with the server application. The server keeps track of the list of connected sessions (e.g. using a Set):

```
// contains the sessions connected  
private final Set<Session> sessions = new HashSet<>();
```



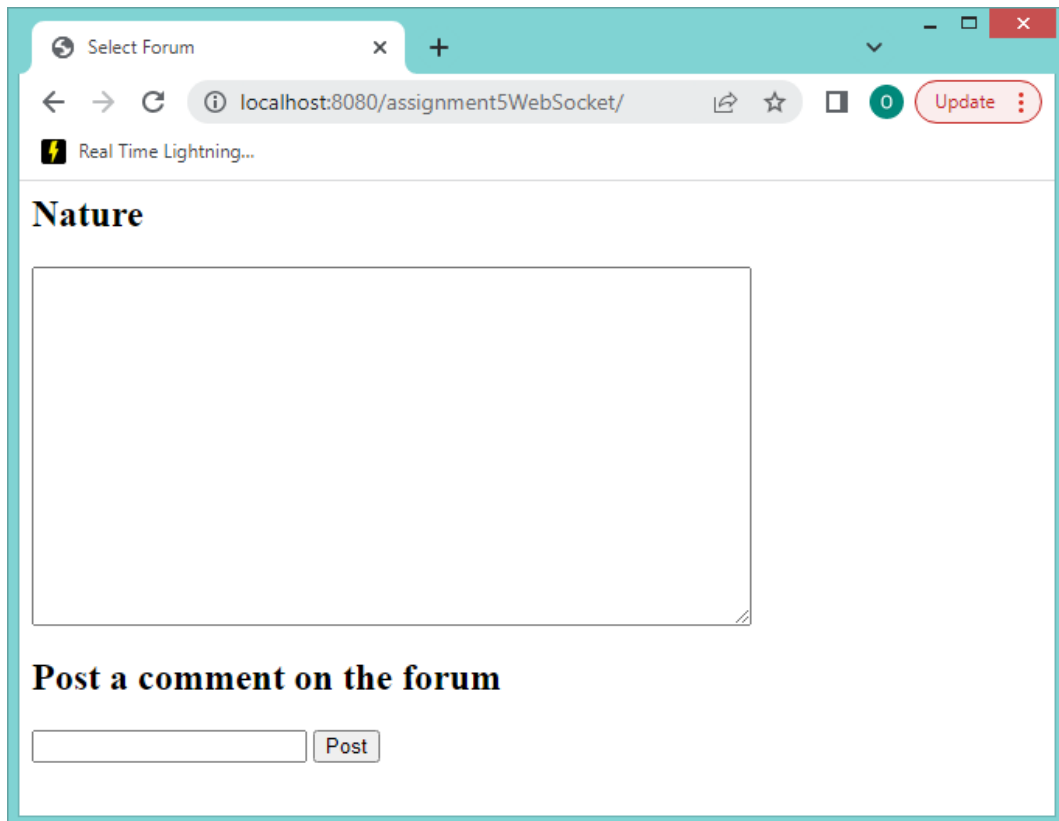
If the websocket opens successfully, the onopen() method on the client side should send a (JSON) message request for that client to join the chosen forum, e.g.:

```
{action = "join", forum : "Politics"}
```

So the server keeps track of which forum a session is associated with (e.g. HashMap of session id (String) and forum name (String)):

```
// key = session id, value = forum name
private final HashMap<String, String> sessionForums = new HashMap<>();
```

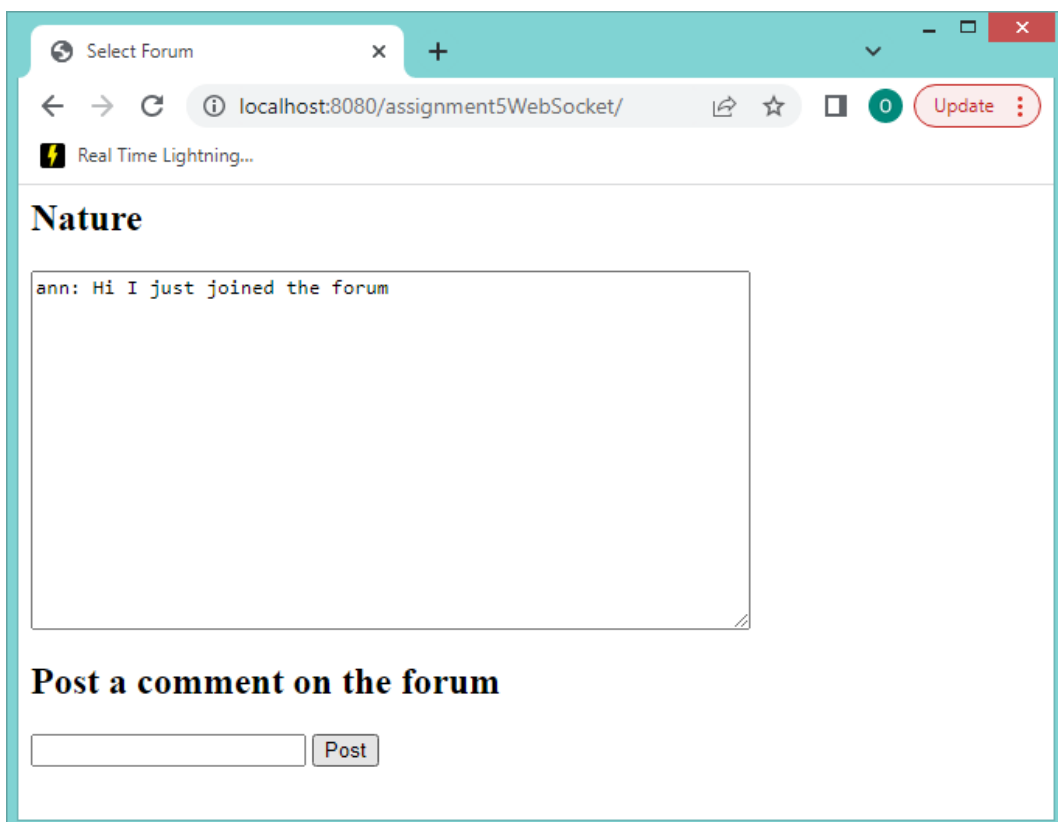
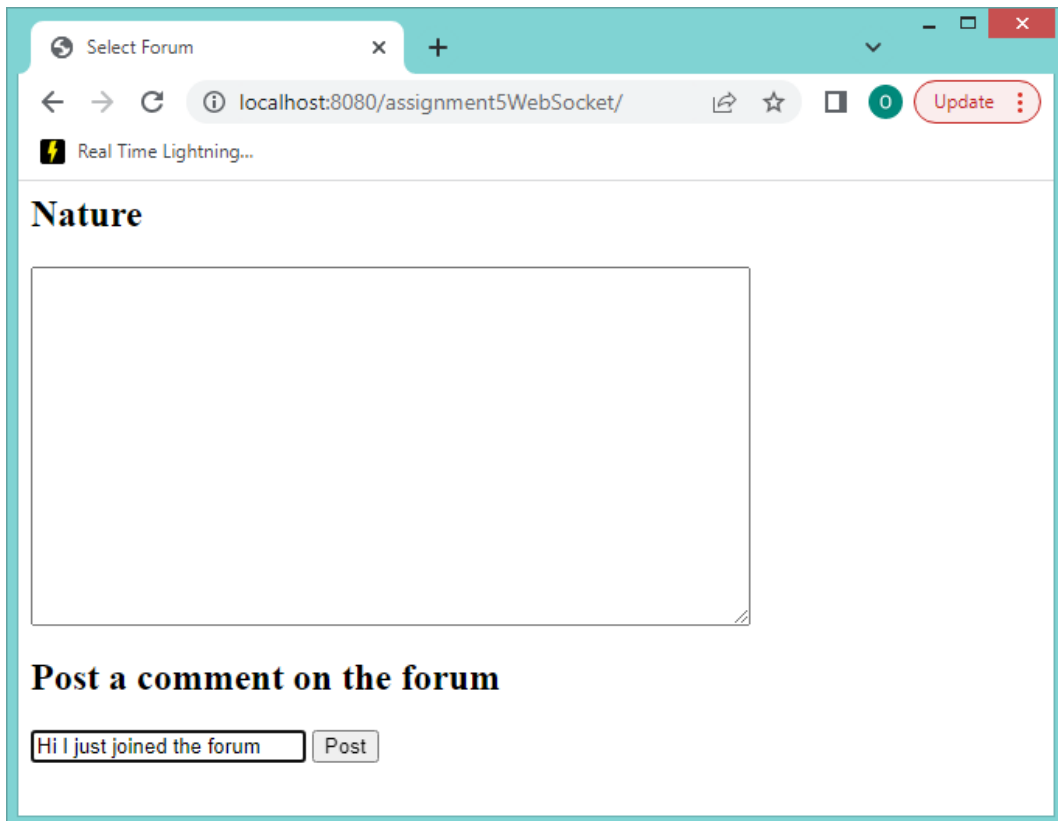
The javascript should then hide the first form, and present the user with a text field containing posts from the forum, and a text input where they can create new posts.



When the user inputs a comment and clicks on 'Post' button, this sends a message to the server to add this post to the posts for that forum. So the server keeps track of the posts for each forum (I used a HashMap<String, ArrayList<String>>, where the key is the forum name, and the value is the ArrayList of posts:

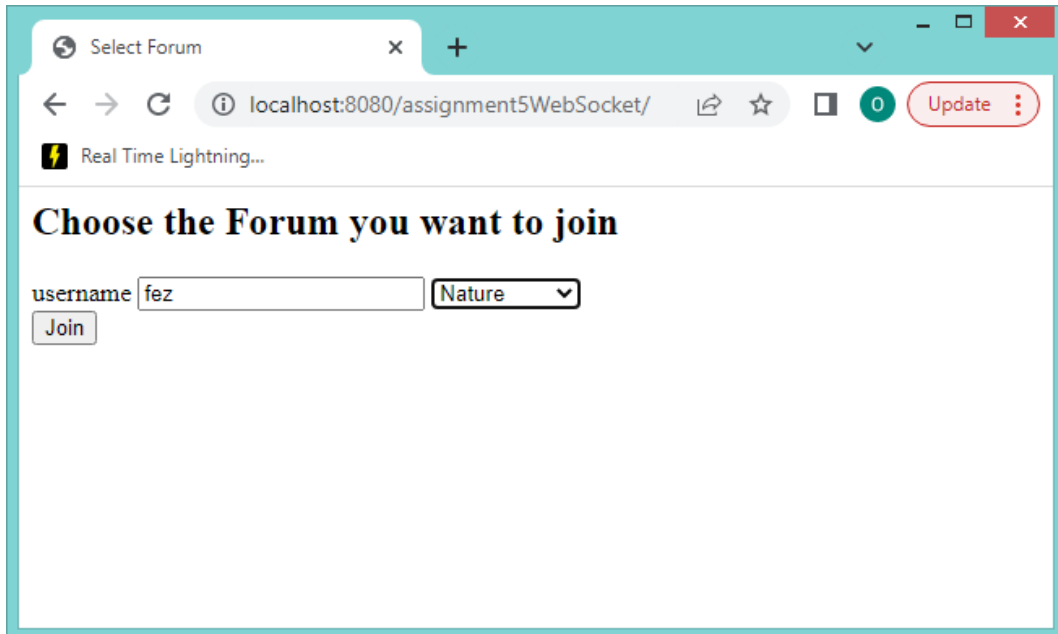
```
// key = forum name, value = posts on that forum
private final HashMap<String, ArrayList<String>> forumPosts = new
HashMap<>();
```

Every time a new post comes in for a forum, I retrieve the ArrayList of posts for that forum from the HashMap, add the new post to it, and then put it back in the HashMap using the put() method.

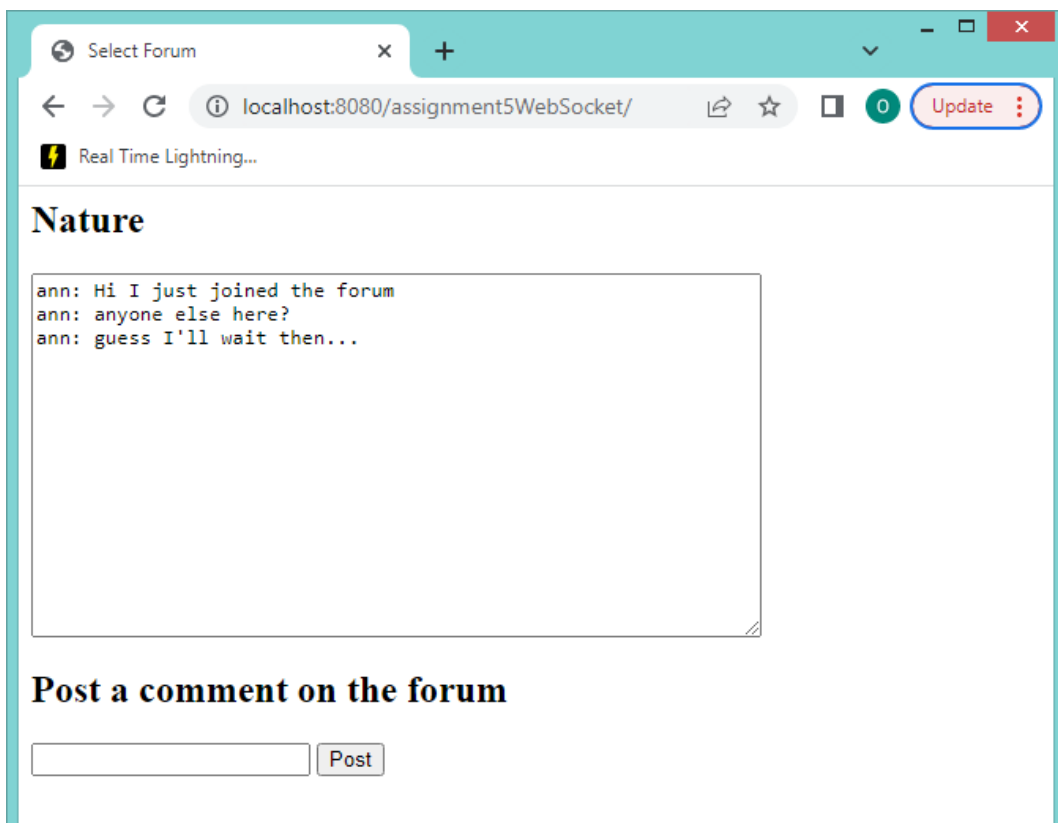


Every time a user posts on their forum, the message is sent to the server and added to the messages for that forum. The server then sends a message back to the client, which it simply adds to the text area.

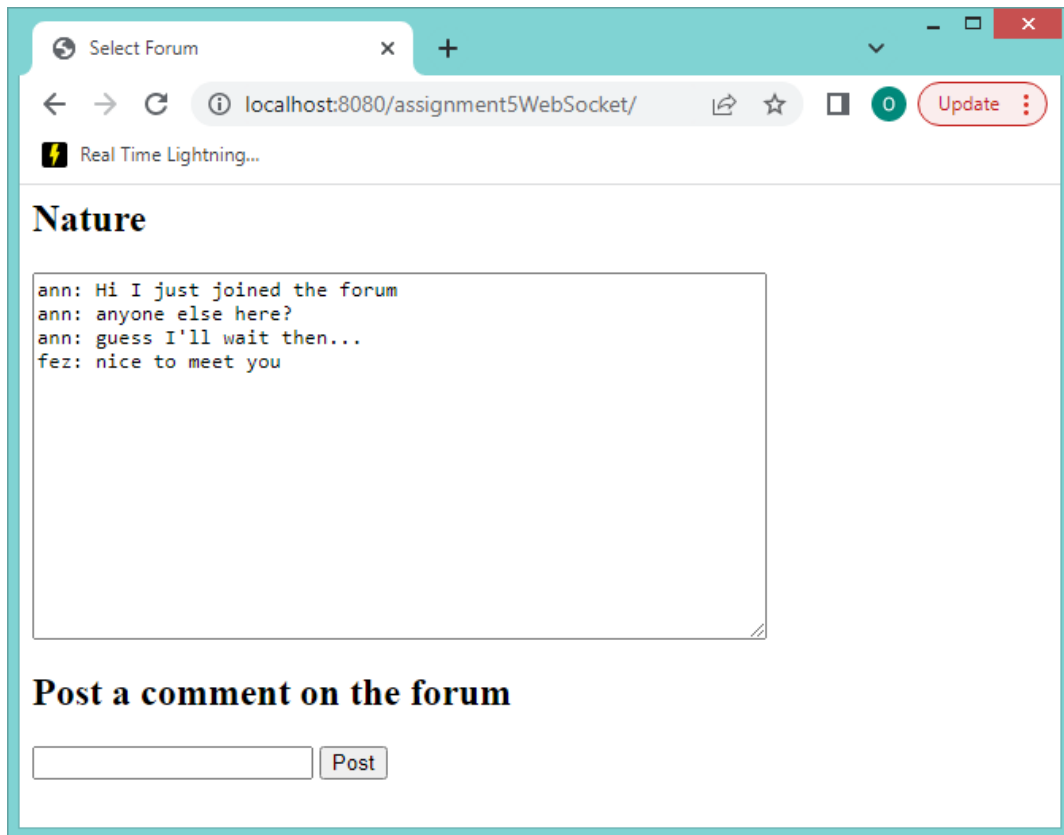
When a new user joins a forum



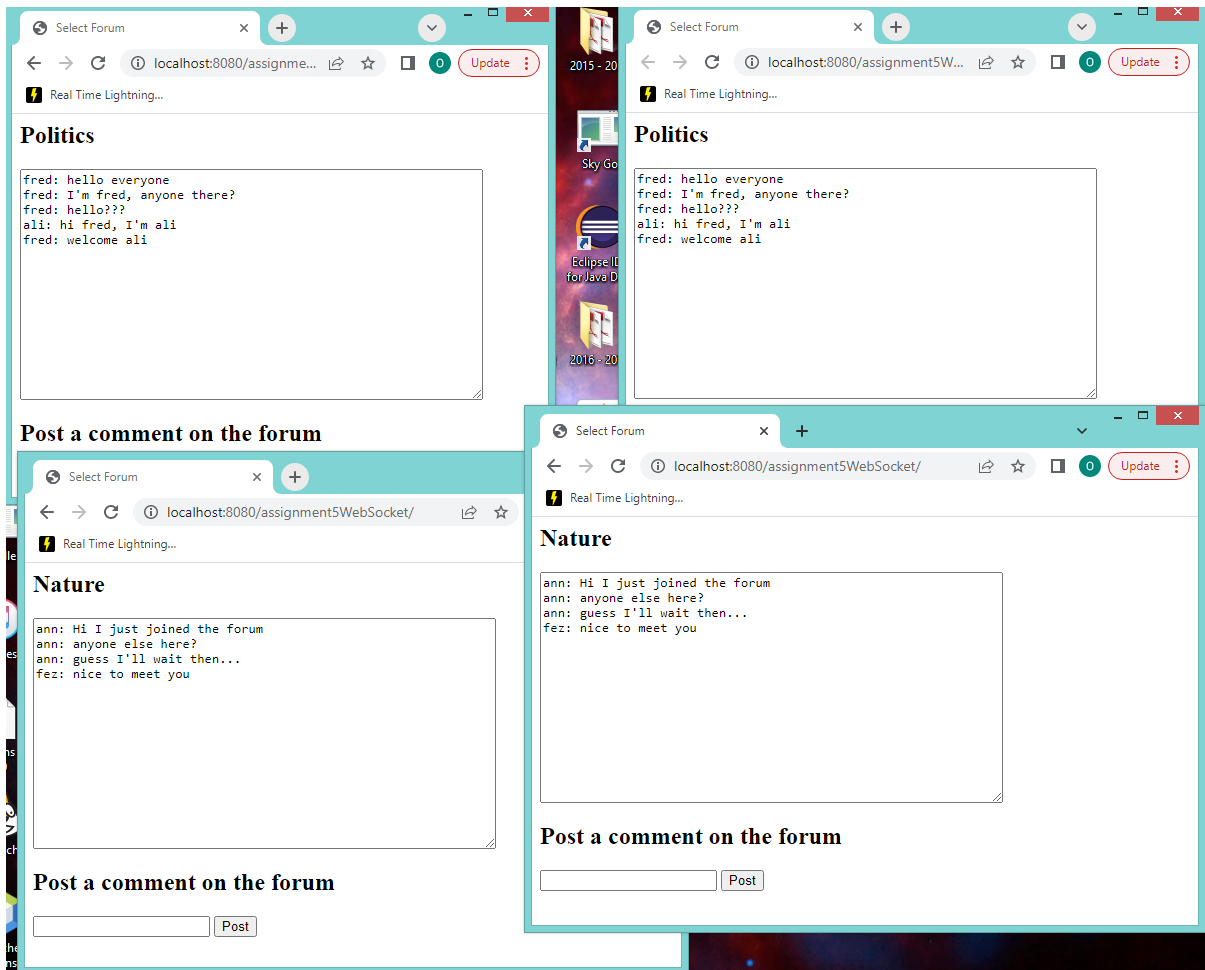
The server should send them all the messages already posted on that forum.



To which of course they can then add:



So we should have a multi-user, multi-forum chat application using WebSocket:



To submit your assignment:

- Perform a 'Clean' on the project
- Zip the project folder
- Submit the Zip file on Blackboard for the assignment