JavaScript chat over network

Assigment

This assignment consists of creating a server that allows multiple clients to chat over the network.

Users will connect to the server in a specified port (that will be easy to configure, just changing a variable) and will talk to the server using a simple protocol defined below:

1. When the socket is opened, the first thing a user has to do is to configure their username typing:

   ```
   USERNAME teixe
   ```

   > The username won't accept spaces in between.

2. To send a message, the user has to type:

   ```
   SEND Hello, world!
   ```

   > A user won't be allowed to send a message if they didn't specify an username first.

3. The user will receive the peers messages with the following format:

   ```
   RECEIVE teixe Hello, world!
   ```

   > The user won't receive their own messages. The receive command receives first the username that sent the message followed by the message itself.

To try the application use the program `netcat` and, once completed, you should be able to reproduce the following session:

```
 root@ubuntu~: nc localhost 8080
USERNAME teixe                              (->)
SEND Hello, world!                          (->)
USERNAME not valid                          (->)
ERROR Username contains empty spaces.       (<-)
NOT_A_COMMAND test                          (->)
ERROR Unrecognised command.                 (<-)
```

> (->) Indicates that this message was sent by the user. (<-) Indicates that this message was sent from the server.

Another example, this time using two connections at the same time:

```
 root@ubuntu~: nc localhost 8080
USERNAME teixe                              (1)
SEND hi                                     (3)
RECEIVE jeff bye                            (-)
```

```
 root@ubuntu~: nc localhost 8080
USERNAME jeff                               (2)
RECEIVE teixe hi                            (-)
SEND bye                                    (4)
```

> The numbers on the right indicate order. (-) indicates that it was automaticly received by the server.

Tips

1. You can try a live working example with `netcat` on *learn.jtei.io* port 8080.

2. Start with the following code:

```
const net = require('net');

const server = net.createServer((socket) => {
  socket.end('goodbye\n');
});

server.listen(8080, () => {
  console.log('opened server on', server.address());
});
```

3. Find more information about NodeJS sockets in: https://nodejs.org/docs/latest-v16.x/api/net.html.