CODE:

Index.html

```html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>WebSocket Chat</title>
7    <link rel="stylesheet" href="style.css" />
8    <script src="/socket.io/socket.io.js"></script>
9  </head>
10 <body>
11
12   <div class="chat-container">
13     <div id="chat-header"> WebSocket Chat</div>
14     <div id="chat-box"></div>
15     <div id="typing-status"></div>
16     <div class="input-area">
17       <input id="message" type="text" placeholder="Type your message..." />
18       <button id="send">Send</button>
19     </div>
20   </div>
21
22   <div class="overlay">
23     <div class="name-box">
24       <h2>Enter your name</h2>
25       <input id="nameInput" type="text" placeholder="Your name..." />
26       <button id="joinBtn">Join Chat</button>
27     </div>
28   </div>
29
30   <script>
31     const socket = io();
32     const chatBox = document.getElementById('chat-box');
33     const input = document.getElementById('message');
34     const sendBtn = document.getElementById('send');
35     const overlay = document.querySelector('.overlay');
36     const nameInput = document.getElementById('nameInput');
37     const joinBtn = document.getElementById('joinBtn');
38     const typingStatus = document.getElementById('typing-status');
39
40     let username = '';
```

```
42
43      joinBtn.addEventListener('click', () => {
44        username = nameInput.value.trim();
45        if (username) {
46          socket.emit('join', username);
47          overlay.style.display = 'none';
48        }
49      });
50
51      sendBtn.addEventListener('click', () => {
52        const msg = input.value.trim();
53        if (msg) {
54          socket.emit('chatMessage', msg);
55          input.value = '';
56          socket.emit('stopTyping');
57        }
58      });
59
60      input.addEventListener('input', () => {
61        socket.emit('typing');
62        clearTimeout(typingTimeout);
63        typingTimeout = setTimeout(() => {
64          socket.emit('stopTyping');
65        }, 1000);
66      });
67
68      socket.on('message', (data) => {
69        const msgEl = document.createElement('div');
70        msgEl.classList.add(
71          data.user === 'System'
72            ? 'system'
73            : data.user === username
74            ? 'self'
75            : 'other'
76        );
77        msgEl.innerHTML = `<strong>${data.user}:</strong> ${data.text}`;
78        chatBox.appendChild(msgEl);
79        chatBox.scrollTop = chatBox.scrollHeight;
80      });
81
82      socket.on('typing', (user) => {
83        typingStatus.textContent = `${user} is typing...`;
```
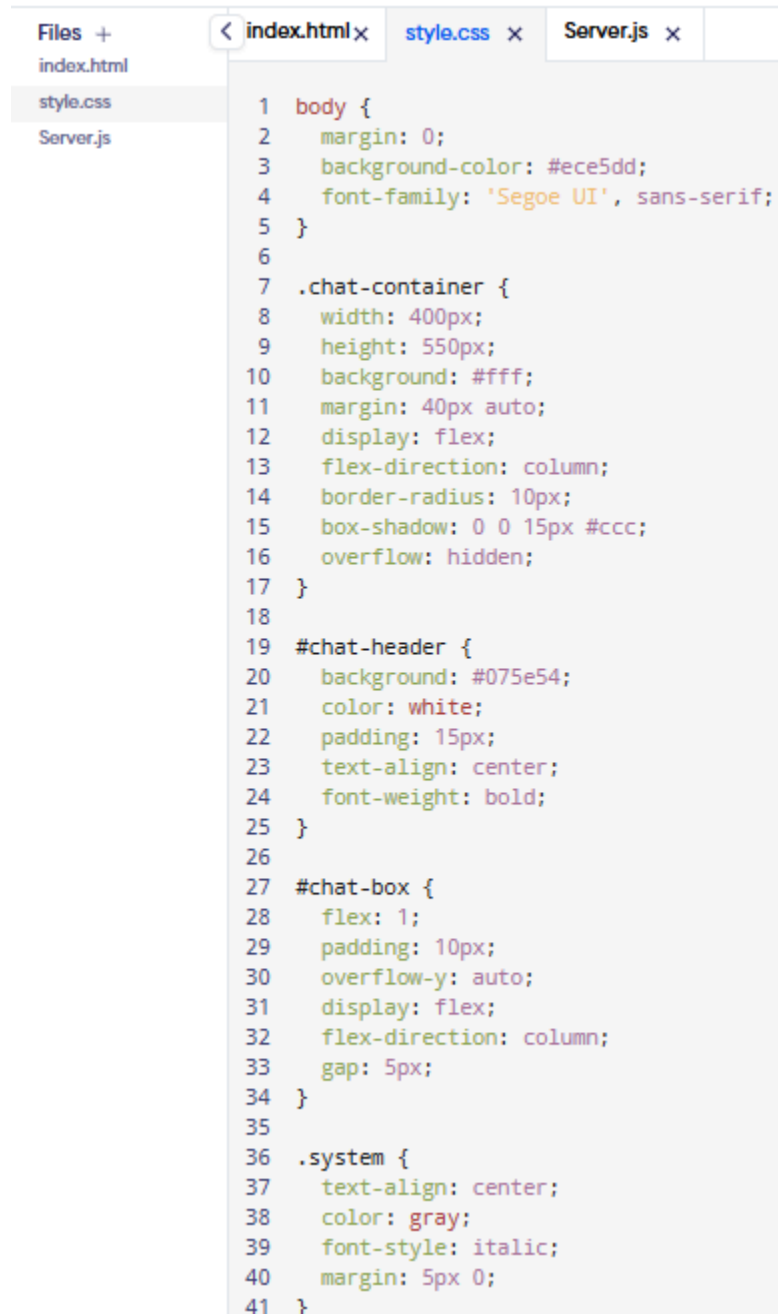
```
82      socket.on('typing', (user) => {
83        typingStatus.textContent = `${user} is typing...`;
84      });
85
86      socket.on('stopTyping', () => {
87        typingStatus.textContent = '';
88      });
89    </script>
90
91  </body>
92  </html>
```

Style.css

< index.html ✕   style.css ✕   Server.js ✕

```css
1   body {
2     margin: 0;
3     background-color: #ece5dd;
4     font-family: 'Segoe UI', sans-serif;
5   }
6
7   .chat-container {
8     width: 400px;
9     height: 550px;
10    background: #fff;
11    margin: 40px auto;
12    display: flex;
13    flex-direction: column;
14    border-radius: 10px;
15    box-shadow: 0 0 15px #ccc;
16    overflow: hidden;
17  }
18
19  #chat-header {
20    background: #075e54;
21    color: white;
22    padding: 15px;
23    text-align: center;
24    font-weight: bold;
25  }
26
27  #chat-box {
28    flex: 1;
29    padding: 10px;
30    overflow-y: auto;
31    display: flex;
32    flex-direction: column;
33    gap: 5px;
34  }
35
36  .system {
37    text-align: center;
38    color: gray;
39    font-style: italic;
40    margin: 5px 0;
41  }
```

```css
40      margin: 5px 0;
41  }
42
43  .self {
44      align-self: flex-end;
45      background: #dcf8c6;
46      padding: 8px 12px;
47      border-radius: 10px;
48      max-width: 70%;
49  }
50
51  .other {
52      align-self: flex-start;
53      background: #fff;
54      border: 1px solid #ccc;
55      padding: 8px 12px;
56      border-radius: 10px;
57      max-width: 70%;
58  }
59
60  .input-area {
61      display: flex;
62      padding: 10px;
63      background: #f0f0f0;
64      border-top: 1px solid #ccc;
65  }
66
67  #message {
68      flex: 1;
69      padding: 10px;
70      border-radius: 20px;
71      border: 1px solid #ccc;
72      outline: none;
73  }
74
75  #send {
76      background: #128c7e;
77      color: white;
78      border: none;
79      border-radius: 20px;
80      margin-left: 10px;
81      padding: 10px 20px;
```

Files +

index.html
style.css
Server.js

< index.html ×   style.css ×   Server.js ×

```css
81      padding: 10px 20px;
82      cursor: pointer;
83  }
84
85  #send:hover {
86      background: #0e766e;
87  }
88
89  #typing-status {
90      text-align: left;
91      font-size: 12px;
92      color: gray;
93      padding-left: 15px;
94      height: 20px;
95  }
96
97  .overlay {
98      position: fixed;
99      top: 0;
100     left: 0;
101     right: 0;
102     bottom: 0;
103     background: rgba(0,0,0,0.6);
104     display: flex;
105     justify-content: center;
106     align-items: center;
107 }
108
109 .name-box {
110     background: #fff;
111     padding: 20px 30px;
112     border-radius: 10px;
113     box-shadow: 0 0 10px #333;
114     text-align: center;
115 }
116
117 .name-box h2 {
118     margin-bottom: 15px;
119 }
120
```

```css
120
121 .name-box input {
122     padding: 8px;
123     width: 80%;
124     margin-bottom: 10px;
125     border: 1px solid #ccc;
126     border-radius: 5px;
127 }
128
129 .name-box button {
130     background: #075e54;
131     color: white;
132     border: none;
133     border-radius: 5px;
134     padding: 8px 15px;
135     cursor: pointer;
136 }
137
138 .name-box button:hover {
139     background: #128c7e;
140 }
141
```

Server.js

```javascript
Files  +        < index.html ×   style.css ×   Server.js ×

index.html
style.css        1  const express = require('express');
Server.js        2  const http = require('http');
                 3  const { Server } = require('socket.io');
                 4  const app = express();
                 5  const server = http.createServer(app);
                 6  const io = new Server(server);
                 7
                 8  app.use(express.static('public'));
                 9
                10  io.on('connection', (socket) => {
                11    console.log('A user connected');
                12
                13    socket.on('join', (username) => {
                14      socket.username = username;
                15      console.log(`${username} joined the chat`);
                16      io.emit('message', { user: 'System', text: `${username} joined the chat` });
                17    });
                18
                19    socket.on('chatMessage', (msg) => {
                20      io.emit('message', { user: socket.username, text: msg });
                21    });
                22
                23    socket.on('typing', () => {
                24      socket.broadcast.emit('typing', socket.username);
                25    });
                26
                27    socket.on('stopTyping', () => {
                28      socket.broadcast.emit('stopTyping');
                29    });
                30
                31    socket.on('disconnect', () => {
                32      if (socket.username) {
                33        io.emit('message', { user: 'System', text: `${socket.username} left the chat` });
                34      }
                35      console.log('User disconnected');
                36    });
                37  });
                38
                39  const PORT = 3000;
                40  server.listen(PORT, () => console.log(`✅ Server running on http://localhost:${PORT}`));
                41
```
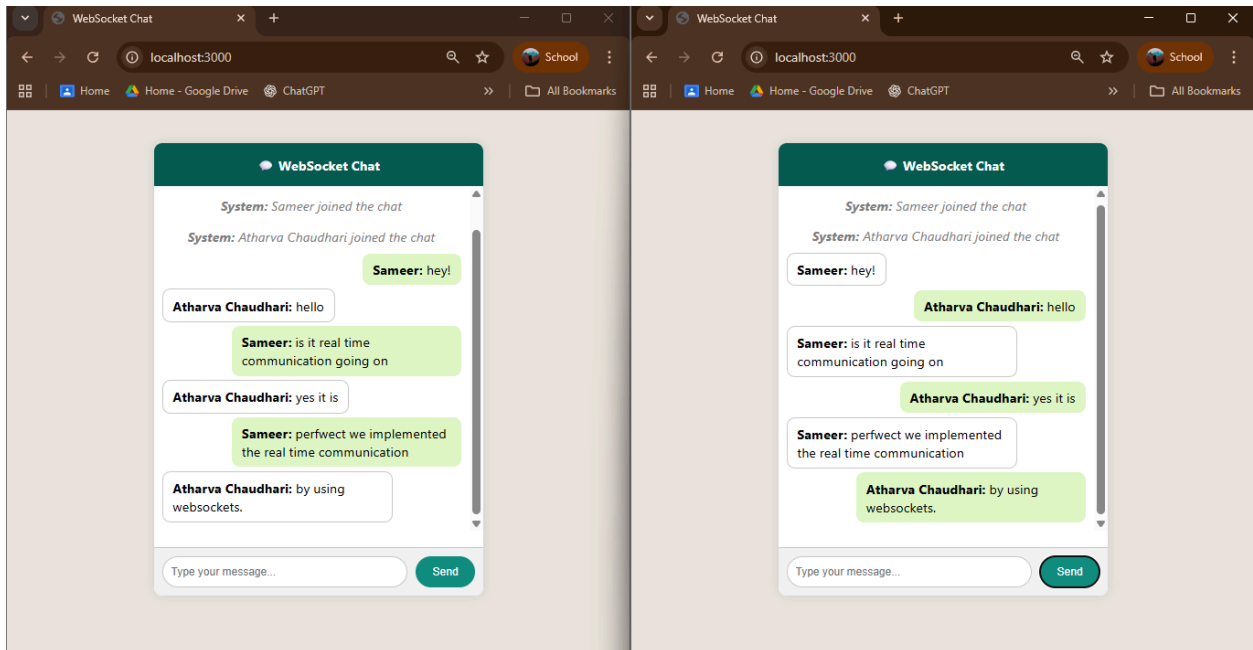
OUTPUT: