EXPERIMENT NO. - 08

Roll No.	24
Name	Chaitanya Dinesh Dhayarkar
Class	D15B
Subject	Full Stack Development
Lab Outcome	Enable **real-time communication** via WebSockets
Date of Performance/ Submission	
Signature & Grades	

Experiment No. 8

Aim: Enable **real-time communication** via WebSockets

Directory Structure:

GROUPCHAT

|

├── backend → [node_modules, package-lock.json, package.json, server.js]

└── frontend →

    ├── css → [style.css]

    ├── js → [chat.js]

    └── index.html

Code:

server.js

```
const express = require("express");
const http = require("http");
const { Server } = require("socket.io");
const path = require("path");

const app = express();
const server = http.createServer(app);
const io = new Server(server);

app.use(express.static(path.join(__dirname, "../frontend")));

io.on("connection", (socket) => {
  console.log("🟢 New user connected");

  socket.on("new_user", (username) => {
    socket.username = username;
    io.emit("receive_message", { username: "System", message: `${username} joined the chat` });
  });
```

```
  socket.on("send_message", (data) => {
    socket.broadcast.emit("receive_message", data);
  });

  socket.on("disconnect", () => {
    if (socket.username) {
      io.emit("receive_message", { username: "System", message:
`${socket.username} left the chat` });
    }
  });
});

const PORT = 5000;
server.listen(PORT, () => console.log(`🚀 Server running on
http://localhost:${PORT}`));
```

chat.js

```
const socket = io();

let username = prompt("Enter your name:");
if (!username || username.trim() === "") username = "Anonymous";

document.getElementById("userDisplay").textContent = `You: ${username}`;

socket.emit("new_user", username);

const form = document.getElementById("chatForm");
const input = document.getElementById("messageInput");
const chatBox = document.getElementById("chatBox");

form.addEventListener("submit", (e) => {
  e.preventDefault();
  const message = input.value.trim();
  if (message) {
    socket.emit("send_message", { username, message });
    appendMessage("You", message, "right");
    input.value = "";
  }
});
```

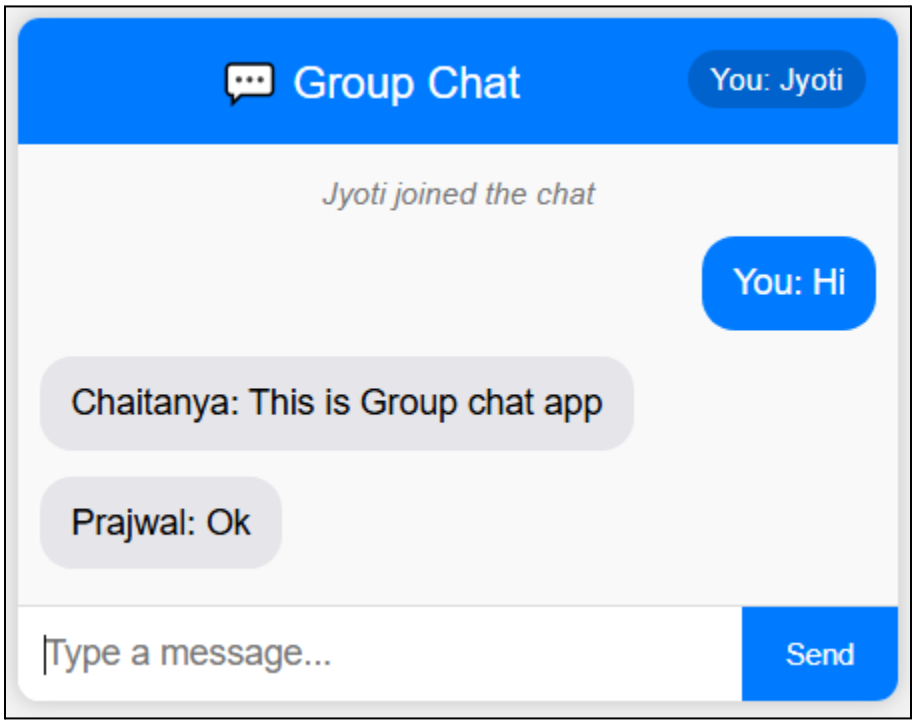
```
socket.on("receive_message", (data) => {
  if (data.username === "System") {
    const msg = document.createElement("p");
    msg.className = "system";
    msg.textContent = data.message;
    chatBox.appendChild(msg);
  } else if (data.username !== username) {
    appendMessage(data.username, data.message, "left");
  }
  chatBox.scrollTop = chatBox.scrollHeight;
});

function appendMessage(name, message, position) {
  const msg = document.createElement("div");
  msg.classList.add("message", position);
  msg.textContent = `${name}: ${message}`;
  chatBox.appendChild(msg);
}
```

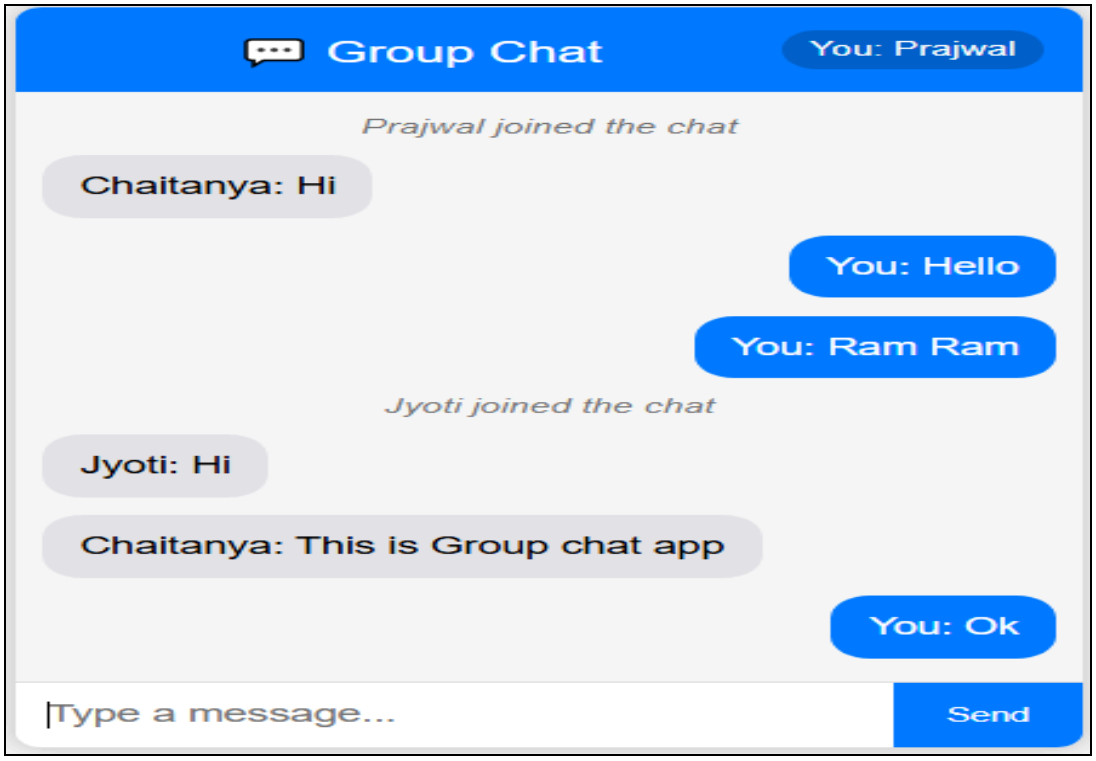
Index.html

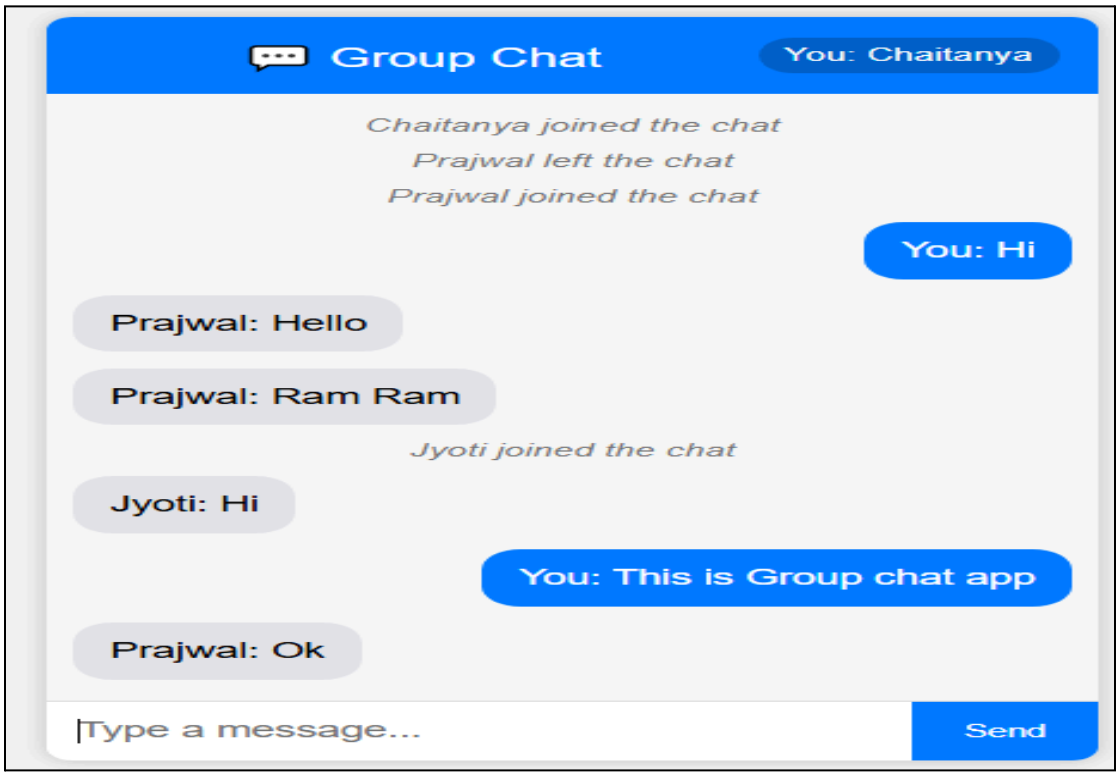
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Realtime Group Chat</title>
  <link rel="stylesheet" href="css/style.css" />
  <script src="https://cdn.socket.io/4.7.2/socket.io.min.js"></script>
</head>
<body>
  <div class="container">
    <div class="header">
      💬 Group Chat
      <span id="userDisplay" class="username-tag"></span>
    </div>

    <div class="chat-box" id="chatBox"></div>
```

```
    <form id="chatForm">
      <input id="messageInput" placeholder="Type a message..."
autocomplete="off" />
      <button type="submit">Send</button>
    </form>
  </div>
  <script src="js/chat.js"></script>
</body>
</html>
```

Output:




Group Chat
You: Prajwal
Prajwal joined the chat
Chaitanya: Hi
You: Hello
You: Ram Ram
Jyoti joined the chat
Jyoti: Hi
Chaitanya: This is Group chat app
You: Ok
Type a message...
Send


Group Chat
You: Chaitanya
Chaitanya joined the chat
Prajwal left the chat
Prajwal joined the chat
You: Hi
Prajwal: Hello
Prajwal: Ram Ram
Jyoti joined the chat
Jyoti: Hi
You: This is Group chat app
Prajwal: Ok
Type a message...
Send