mkdir websocket-chat

cd websocket-chat

npm init -y

npm install ws express

const express = require('express');

const WebSocket = require('ws');

const http = require('http');

const app = express();

const server = http.createServer(app);

const wss = new WebSocket.Server({ server });

// Serve static files

app.use(express.static('public'));

// Store connected clients

const clients = new Set();

// Handle WebSocket connections

wss.on('connection', (ws) => {

clients.add(ws);

console.log('Client connected');

ws.on('message', (message) => {

console.log(`Received: ${message}`);

// Broadcast message to all connected clients

clients.forEach(client => {

if (client !== ws && client.readyState === WebSocket.OPEN) {

client.send(message);

}

});

});

ws.on('close', () => {

clients.delete(ws);

console.log('Client disconnected');

});

});

// Start the server

const PORT = process.env.PORT || 3000;

server.listen(PORT, () => {

console.log(`Server is listening on port ${PORT}`);

});

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>WebSocket Chat</title>

<style>

body { font-family: Arial, sans-serif; }

#messages { list-style-type: none; padding: 0; }

#messages li { padding: 5px; }

#input { width: 300px; }

</style>

</head>

<body>

<h1>WebSocket Chat</h1>

<ul id="messages"></ul>

<input id="input" type="text" placeholder="Type a message..." />

<button id="send">Send</button>

<script>

const ws = new WebSocket('ws://localhost:3000');

// Handle incoming messages

ws.onmessage = (event) => {

const messagesList = document.getElementById('messages');

const newMessage = document.createElement('li');

newMessage.textContent = event.data;

messagesList.appendChild(newMessage);

};

// Send messages

document.getElementById('send').onclick = () => {

const input = document.getElementById('input');

const message = input.value;

ws.send(message);

input.value = '';

};

// Optional: Handle connection open/close events

ws.onopen = () => console.log('Connected to server');

ws.onclose = () => console.log('Disconnected from server');

</script>

</body>

</html>

// Handle incoming messages with notifications

ws.onmessage = (event) => {

const messagesList = document.getElementById('messages');

const newMessage = document.createElement('li');

newMessage.textContent = event.data;

messagesList.appendChild(newMessage);

// Show a browser notification

if (Notification.permission === 'granted') {

new Notification('New message', { body: event.data });

}

};

// Request notification permission

if (Notification.permission !== 'granted') {

Notification.requestPermission();

}