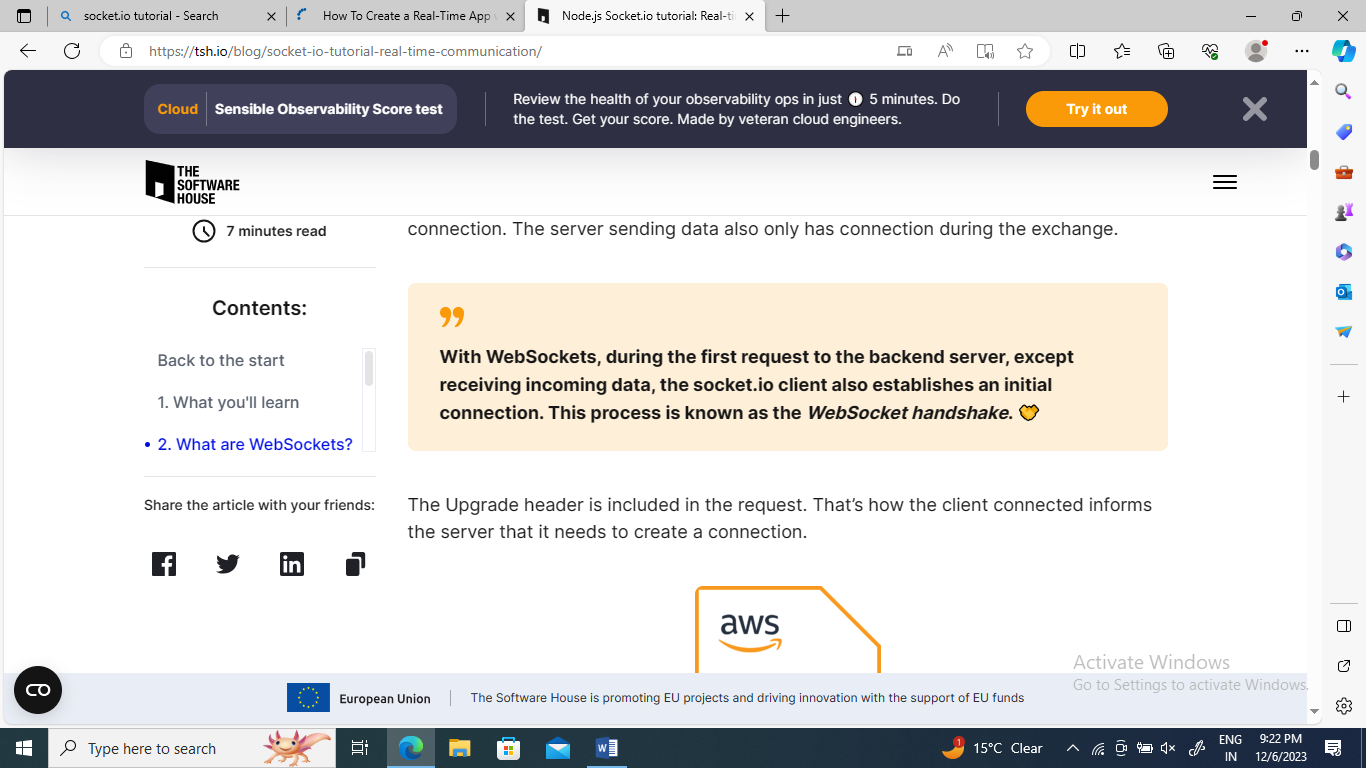
**Lec 31-33 WebSocket :- creating a chatting application**

**WebSocket**

WebSockets API is a technology providing a bidirectional communication channel between a client and a server. That means that the client no longer needs to be an initiator of a transaction while requesting data from both the server and database.

### How do WebSockets work?

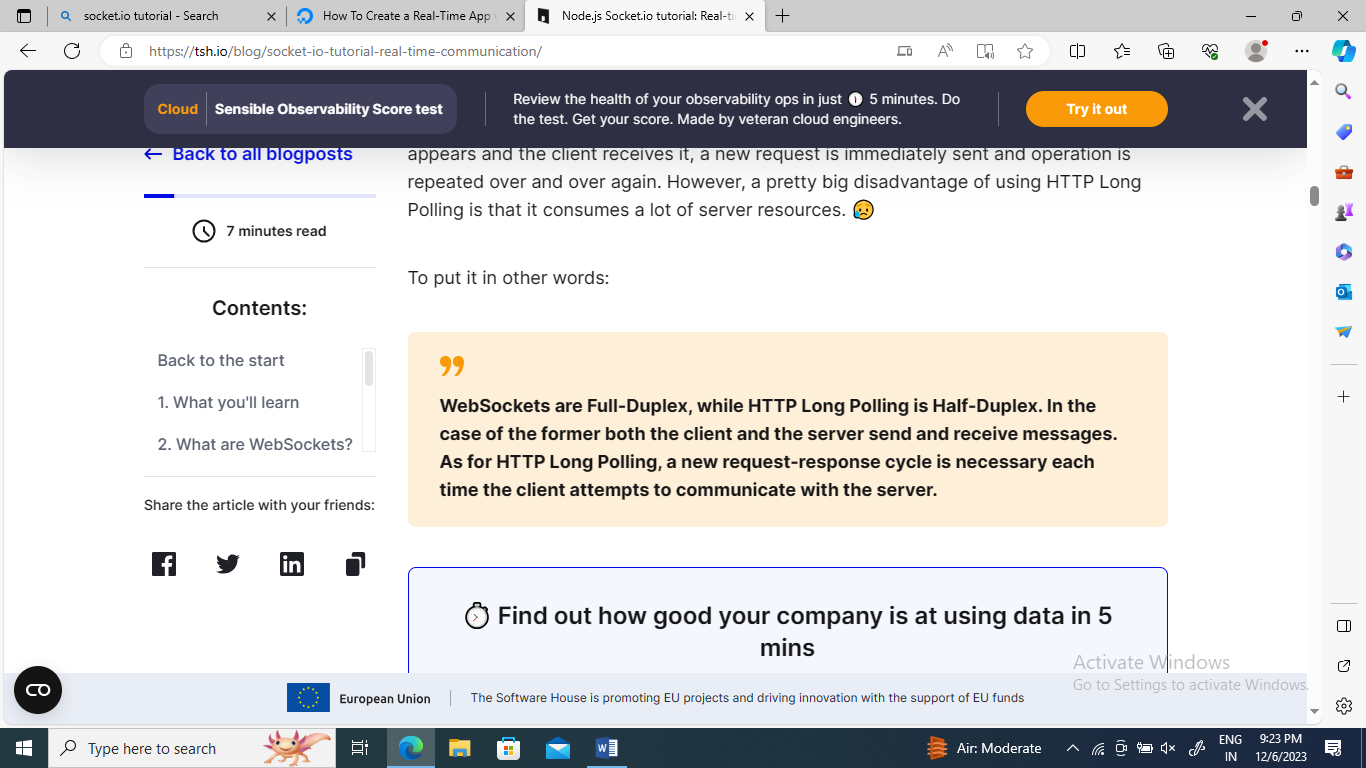
Typically, the client requests data starting a new connection and then the client loses connection. The server sending data also only has connection during the exchange.



## **WebSockets vs HTTP**

**Ques:- Does it mean it’s impossible to create a real-time chat application without WebSockets?**

**Ans:-** Well, there is a technique **called HTTP Long Polling**. Using this, the client sends a request and the server holds that opened until some new data is available. As soon as data appears and the client receives it, a new request is immediately sent and operation is repeated over and over again. However, a pretty big disadvantage of using HTTP Long Polling is that it consumes a lot of server resources**. To put it in other words:**



**The Socket io library is a JavaScript library that enables real-time, bidirectional, event-based communication** between the connected clients (browser) and the server side. It consists of a JavaScript client library and another one dedicated to servers. Both components share most of their API.

**Practical implementation**

**Step 1:** Make a folder named as **project** & open terminal from root directory then type the command

**npm init**

This will create a package.json file.

**Step 2:** Now run the below command to install express Js

**npm install express**

**Step 3:** Check in package.json whether all the dependences has been installed or not.

**Step 4:** Now make an **index.js** file in the **project** folder.

**index.js**

**const http=require("http");**

**const express=require("express");**

**const path=require("path");**

**const app=express();**

**const server=http.createServer(app);**

**app.use(express.static(path.resolve("./public")));**

**app.get("/",(req,res)=>{**

**return res.sendFile("./public/index.html");**

**});**

**server.listen(9000,()=>console.log(`server started at port :9000`));**

**Step 5:** create a **public folder** inside project folder & make **index.html** file inside it.

**index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

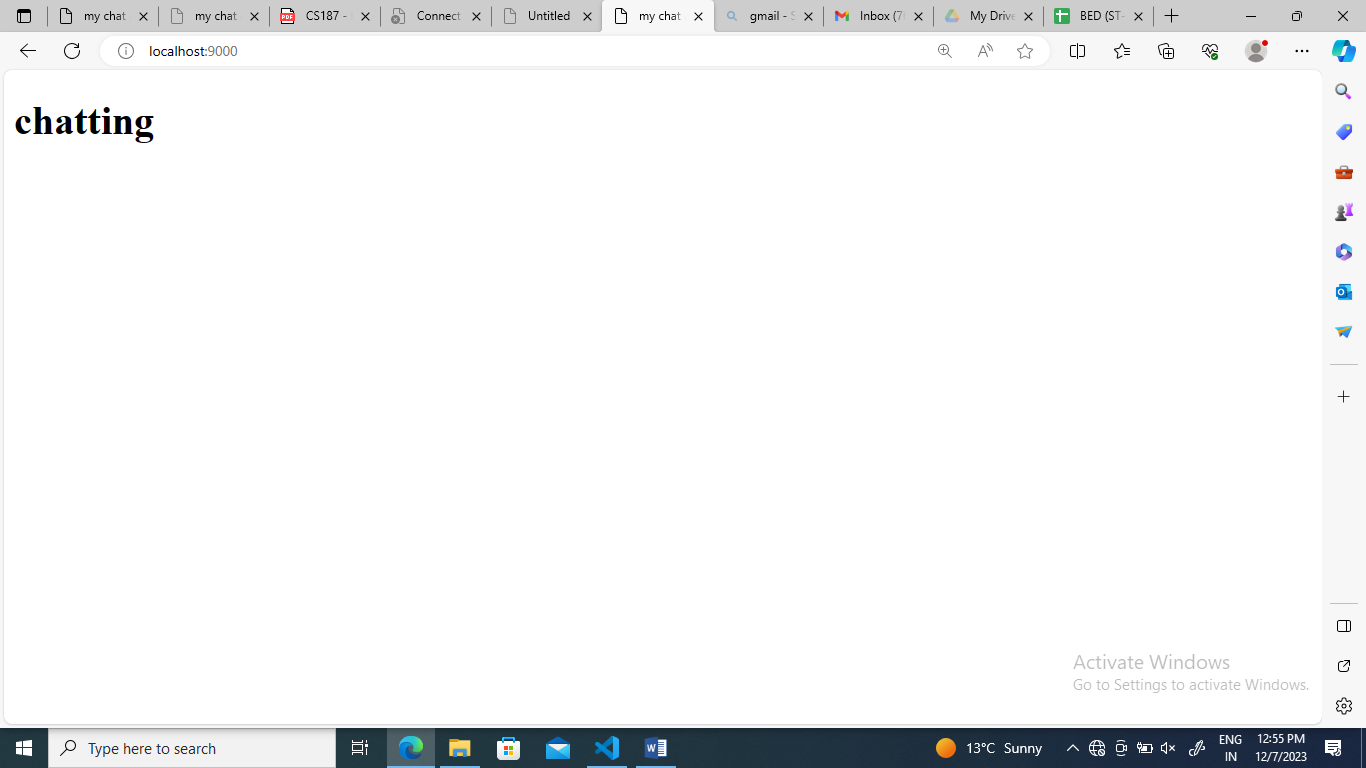
**<h1>chatting</h1>**

**</body>**

**Step 6**: Now run the files

**Run:** node index.js

Output:



**Step 7:** Now install the **socket.io** package so make socket connections by using the command:

**npm install socket.io**

Now updated version of index.js is

**const http=require("http");**

**const express=require("express");**

**const path=require("path");**

**const { Server } = require("socket.io");**

**const app=express();**

**const server=http.createServer(app);**

**const io= new Server(server);**

**//socket.io**

**io.on("connection", (socket) =>{**

**console.log(" a new user has connected",socket.id);**

**});**

**app.use(express.static(path.resolve("./public")));**

**app.get("/",(req,res)=>{**

**return res.sendFile("./public/index.html");**

**});**

**server.listen(9000,()=>console.log(`server started at port :9000`));**

**updated index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

**<h1>chatting</h1>**

**<script src="/socket.io/socket.io.js"></script>**

**</body>**

Now run the file again

**Run: node index.js**

**Output: Now inspect the output page & goto network tab & check for the url , you will be able to see the long script .**

Now update index.js

**Now update index.html**

Let’s create a button so that after clicking the button a **ws** must be created which will make http request & will have an **upgrade header.**

**Index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

**<h1>chatting</h1>**

**<button onclick="createConnection()">create WS connection</button>**

**<script src="/socket.io/socket.io.js"></script>**

**<script>**

**function createConnection(){**

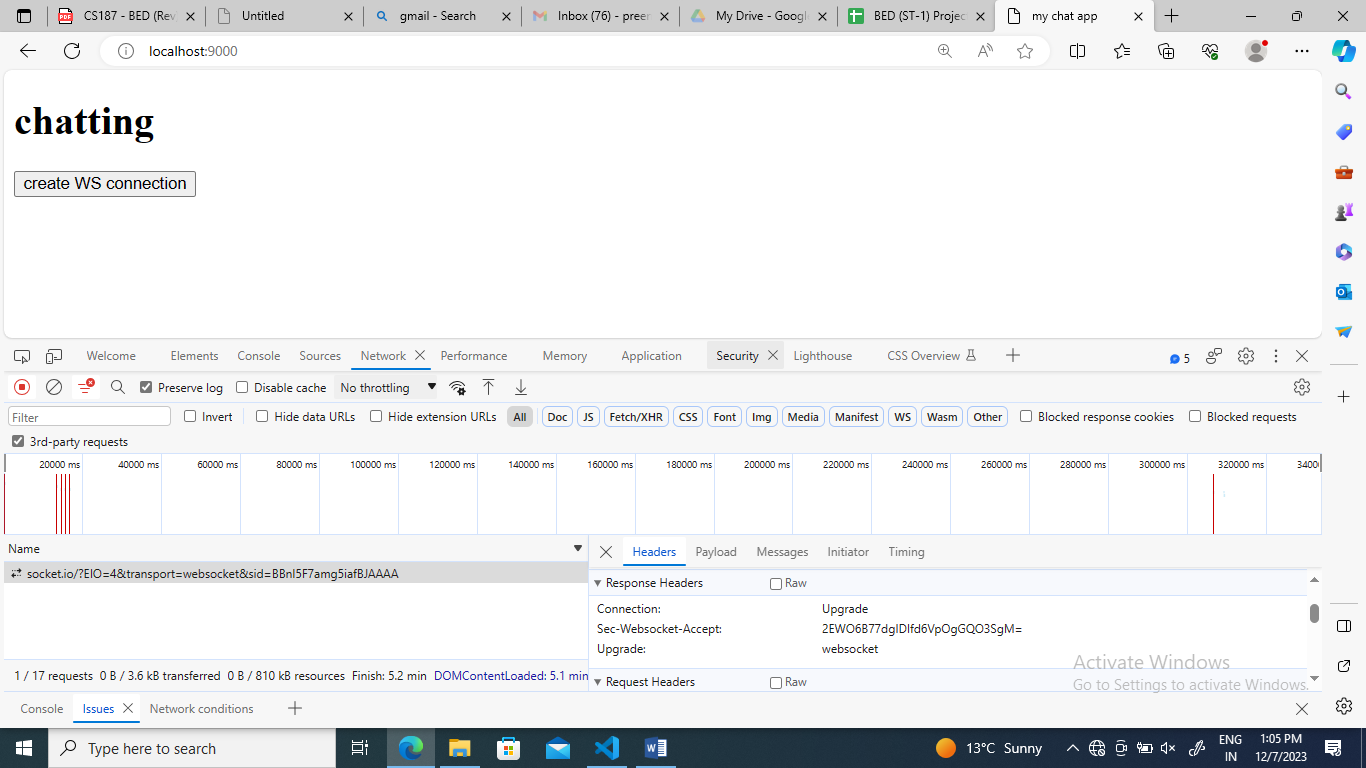
**const socket=io();**

**}**

**</script>**

**</body>**

**Run the file again: node index.js**



**Updated index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

**<h1>chatting</h1>**

**<input type="text" id="message" placeholder="enter message"/>**

**<button id="sendBtn">send</button>**

**<script src="/socket.io/socket.io.js"></script>**

**<script>**

**const socket=io();**

**const sendBtn=document.getElementById("sendBtn");**

**const messageInput=document.getElementById("message");**

**sendBtn.addEventListener('click' ,e => {**

**const message = messageInput.value;**

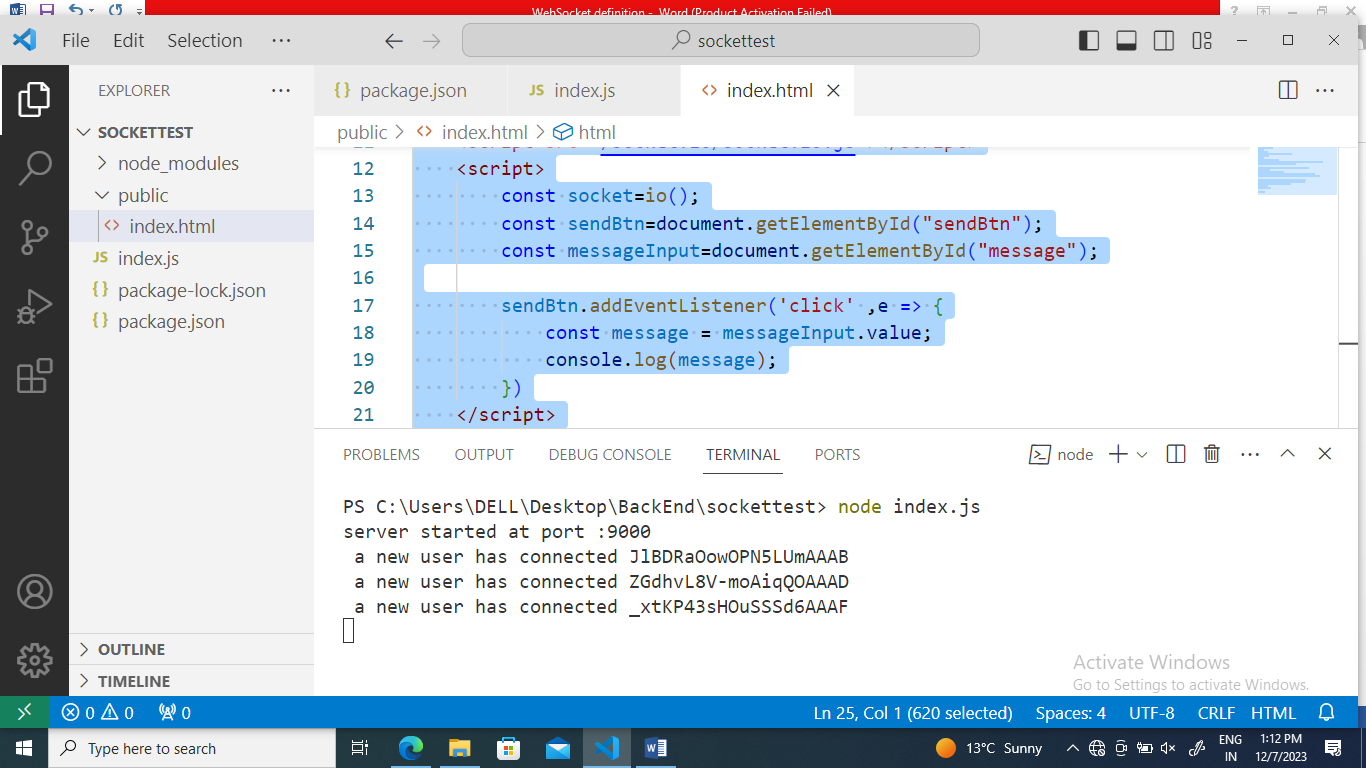
**console.log(message);**

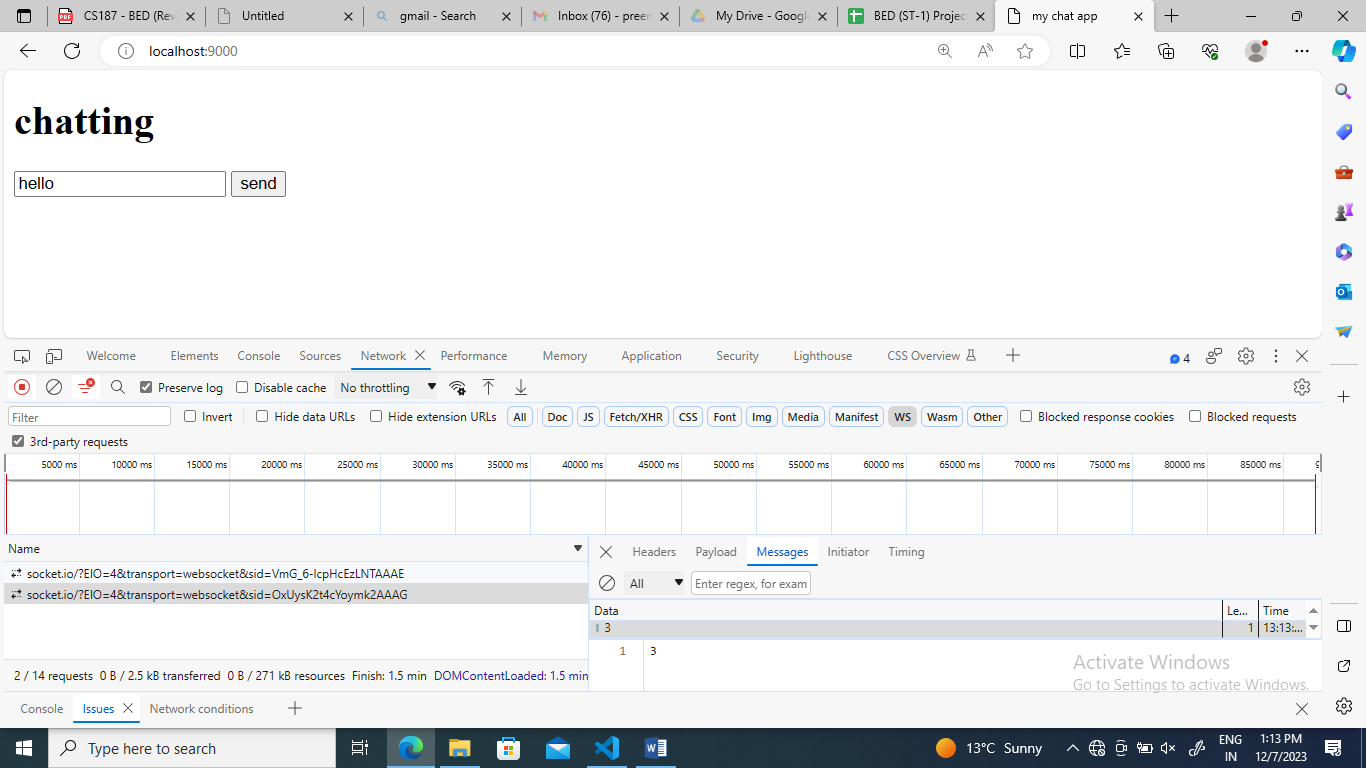
**})**

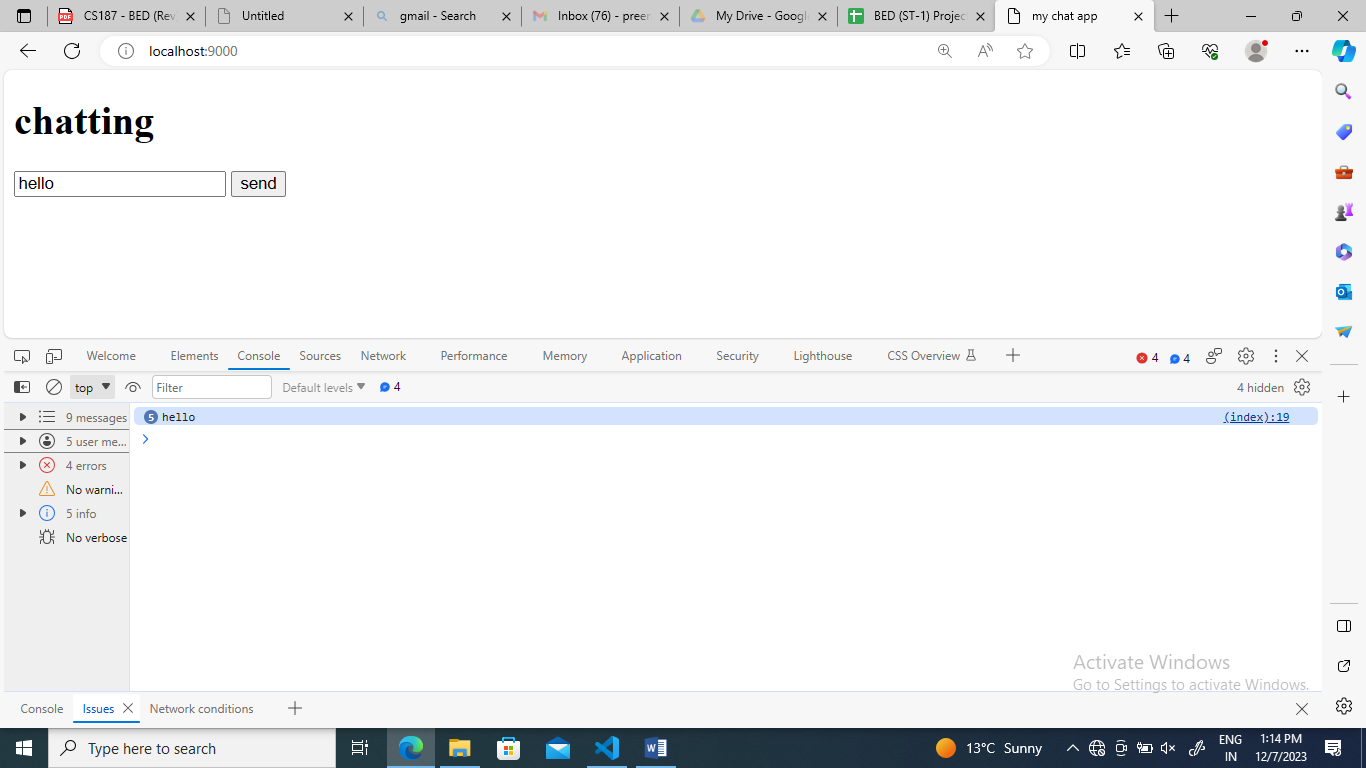
**</script>**

**</body>**

**Run again**







**Updated index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

**<h1>chatting</h1>**

**<input type="text" id="message" placeholder="enter message"/>**

**<button id="sendBtn">send</button>**

**<script src="/socket.io/socket.io.js"></script>**

**<script>**

**const socket=io();**

**const sendBtn=document.getElementById("sendBtn");**

**const messageInput=document.getElementById("message");**

**sendBtn.addEventListener('click' ,e => {**

**const message = messageInput.value;**

**console.log(message);**

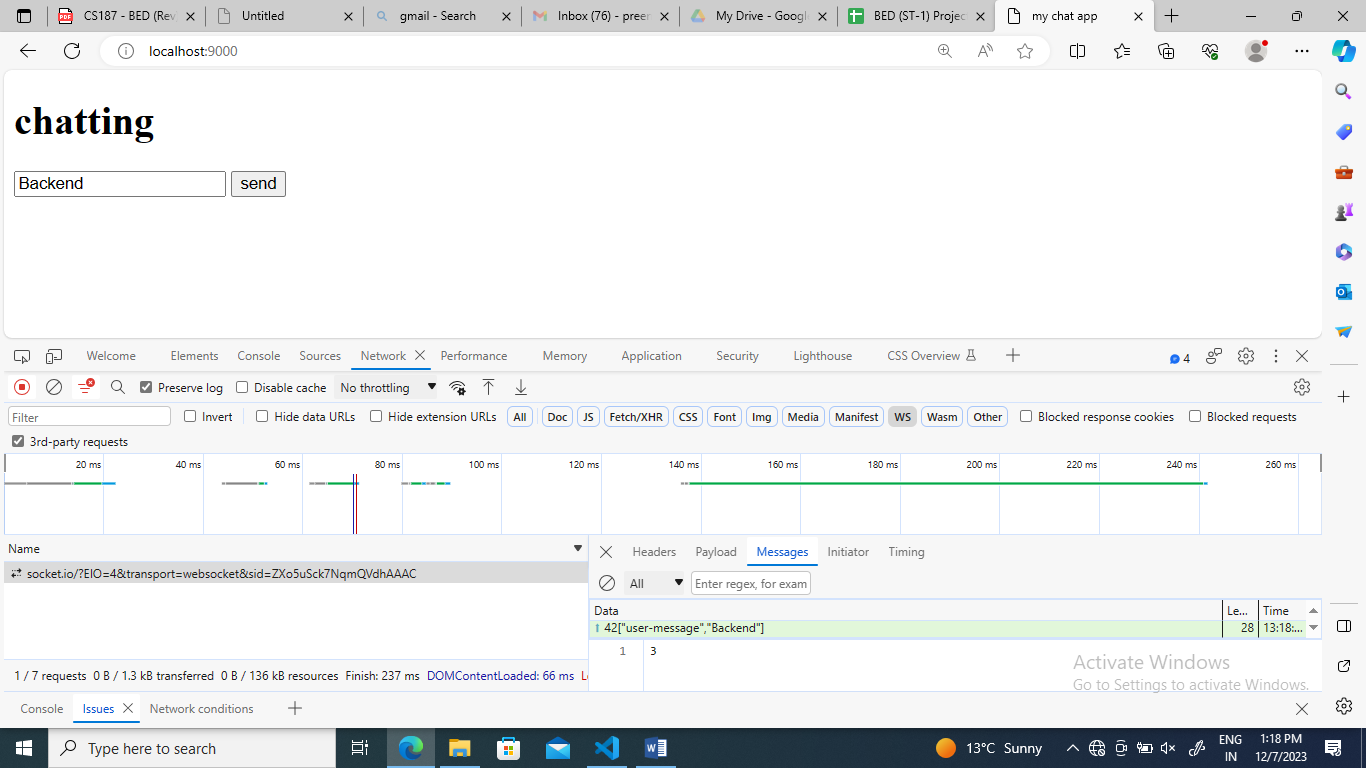
**socket.emit('user-message',message);**

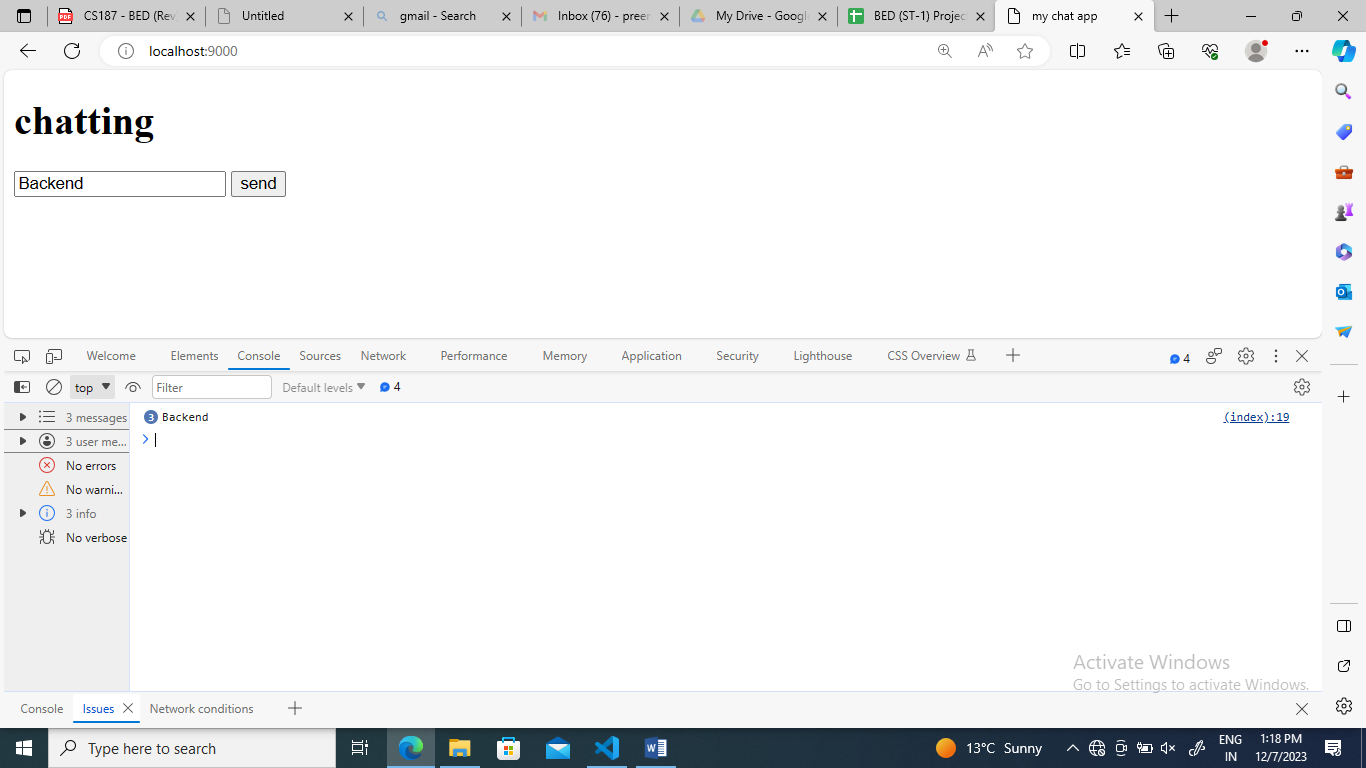
**})**

**</script>**

**</body>**

**Run again: Node index.js**





**Index.js**

**const http=require("http");**

**const express=require("express");**

**const path=require("path");**

**const { Server }=require("socket.io");**

**const app=express();**

**const server=http.createServer(app);**

**const io=new Server(server);**

**//socket.io**

**io.on("connection", (socket) => {**

**socket.on("user-message", (message) =>{**

**//console.log("new user message",message);**

**io.emit("message",message);**

**});**

**});**

**app.use(express.static(path.resolve("./public")));**

**app.get("/",(req,res)=>{**

**return res.sendFile("./public/index.html");**

**});**

**server.listen(9000,()=>console.log(`server started at port :9000`));**

**index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>my chat app</title>**

**</head>**

**<body>**

**<h1>chatting</h1>**

**<input type="text" id="message" placeholder="enter message"/>**

**<button id="sendBtn">send</button>**

**<div id="messages"></div>**

**<script src="/socket.io/socket.io.js"></script>**

**<script>**

**const socket=io();**

**const sendBtn=document.getElementById("sendBtn");**

**const messageInput=document.getElementById("message");**

**const allMessages=document.getElementById("messages");**

**socket.on('message', (message) => {**

**const p=document.createElement("p");**

**p.innerText=message;**

**allMessages.appendChild(p);**

**});**

**sendBtn.addEventListener("click",(e) => {**

**const message=messageInput.value;**

**console.log(message);**

**socket.emit("user-message",message);**

**});**

**</script>**

**</body>**

**RUN: node index.js**

