**Classroom Activity: Implement real-time features in an Express application using WebSocket.**

**Step 1: Set Up Your Project**

mkdir webspckets

cd websockets

npm init -y

npm install socket.io

**Step 2: Create the Application Files**

create app.js and index.html

**Step 3: Set Up the Express Server**

Open app.js and add the following code:

const express = require('express');

const http = require('http');

const socketIo = require('socket.io');

const app = express();

const server = http.createServer(app);

const io = socketIo(server);

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

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

    res.sendFile(\_\_dirname + '/index.html');

});

io.on('connection', (socket) => {

    console.log('a user connected');

    socket.on('disconnect', () => {

        console.log('user disconnected');

    });

    socket.on('chat message', (msg) => {

        io.emit('chat message', msg);

    });

});

server.listen(PORT, () => {

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

});

**Step 4: Create the HTML Form for File Upload**

Open index.html and add the following code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>WebSocket Chat</title>

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

<script>

document.addEventListener('DOMContentLoaded', (event) => {

const socket = io();

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

const messageElement = document.createElement('div');

messageElement.textContent = msg;

document.body.appendChild(messageElement);

});

document.querySelector('form').addEventListener('submit', (e) => {

e.preventDefault();

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

socket.emit('message', input.value);

input.value = '';

});

});

</script>

</head>

<body>

<form>

<input type="text" placeholder="Type your message">

<button type="submit">Send</button>

</form>

</body>

</html>

**Step 5: Run the Application**

node app.js

**Output:**

