


Activity Title: “Live Emoji Wall”

A collaborative app where students can post emojis to a shared screen in real time.

Key Concepts Reinforced:

Concept	How It's Used
WebSockets	Students see emoji posts from others instantly
View Engine (EJS)	Renders the homepage and emoji board
Serverless-style Logic	A function logs each emoji submission

Overview

 “Students type their name, pick an emoji, and click ‘Send’. Their emoji + name shows up instantly on a shared board — just like in a live stream.”

Project Structure

emoji-wall/

```
|— views/
|  |— index.ejs
|— public/
|  |— style.css
|— server.js
```

Setup Instructions

1. Create the folder & install dependencies:

```
mkdir emoji-wall
cd emoji-wall
npm init -y
npm install express socket.io ejs
```

 **server.js (core app)**

```
const express = require("express");
const app = express();
const http = require("http").createServer(app);
const io = require("socket.io")(http);

app.set("view engine", "ejs");
app.use(express.static("public"));

let emojis = [];

function logEmojiPost(user, emoji) {
  console.log(`[Serverless Log] ${user} posted: ${emoji}`);
}

app.get("/", (req, res) => {
  res.render("index", { emojis });
});

io.on("connection", (socket) => {
  socket.on("new-emoji", (data) => {
    emojis.push(data);
    logEmojiPost(data.user, data.emoji);
    io.emit("update-wall", emojis);
  });
});

http.listen(3000, () => console.log("http://localhost:3000"));
```

🚀 index.ejs (fun interactive UI)

```
<!DOCTYPE html>

<html>

<head>

  <title>Live Emoji Wall</title>

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

  <link rel="stylesheet" href="/style.css" />

</head>

<body>

  <h2>🚀 Live Emoji Wall</h2>

  <input id="user" placeholder="Your name" />

  <input id="emoji" placeholder="Your emoji (e.g. 😊)" />

  <button onclick="sendEmoji()">Send</button>


  <div id="wall">

    <% emojis.forEach(e => { %>

      <div class="emoji-post"><%= e.user %>: <%= e.emoji %></div>

    <% }) %>

  </div>


<script>

  const socket = io();

  function sendEmoji() {

    const user = document.getElementById("user").value;

    const emoji = document.getElementById("emoji").value;

    socket.emit("new-emoji", { user, emoji });
```

```

    }

    socket.on("update-wall", (emojis) => {
      const wall = document.getElementById("wall");
      wall.innerHTML = "";
      emojis.forEach(e => {
        const div = document.createElement("div");
        div.className = "emoji-post";
        div.textContent = `${e.user}: ${e.emoji}`;
        wall.appendChild(div);
      });
    });
  </script>
</body>
</html>

```

style.css

```

body {
  font-family: sans-serif;
  text-align: center;
  background: #f7f7fb;
}


.emoji-post {
  font-size: 2rem;
  margin: 0.5rem;
}


input {

```

```
padding: 0.5rem;  
margin: 0.5rem;  
font-size: 1rem;  
}  
button {  
padding: 0.5rem 1rem;  
background: #4caf50;  
color: white;  
border: none;  
cursor: pointer;  
}
```

Suggested Variants / Extensions:

 Add a “Clear Wall” button (teacher control)

 Count how many times each emoji is used