<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>D.I.M BOT CONTROLLER</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

background: linear-gradient(to bottom right, #e9d5ff, #fbcfe8, #fecaca);

}

.container {

text-align: center;

padding: 2rem;

}

h1 {

font-size: 2.25rem;

font-weight: bold;

margin-bottom: 0.5rem;

animation: color-change 5s infinite;

}

.names {

color: white;

font-size: 1.125rem;

}

.controls {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 0.75rem;

width: 14rem;

}

.button {

width: 100%;

padding: 1.25rem;

border-radius: 0.5rem;

font-size: 1.5rem;

border: none;

cursor: pointer;

transition: background-color 0.2s;

}

.arrow {

background-color: #6366f1;

color: white;

}

.arrow:hover {

background-color: #4f46e5;

}

.arrow.active {

background-color: #fbbf24;

color: #1f2937;

}

.stop {

grid-column: 2;

grid-row: 2;

padding: 0.5rem;

background-color: #ef4444;

}

.stop:hover {

background-color: #dc2626;

}

@keyframes color-change {

0% { color: #f472b6; }

25% { color: #60a5fa; }

50% { color: #34d399; }

75% { color: #fbbf24; }

100% { color: #f472b6; }

}

</style>

</head>

<body>

<div class="container">

<h1>D.I.M BOT CONTROLLER</h1>

<p class="names">Ishanvi Lakhani, Megha Esturi, Deepika Agrawal</p>

</div>

<div class="controls">

<button class="button arrow" id="up">▲</button>

<button class="button arrow" id="left">◄</button>

<button class="button stop" id="stop">STOP</button>

<button class="button arrow" id="right">►</button>

<button class="button arrow" id="down">▼</button>

</div>

<script>

let ws;

// Function to connect to the WebSocket

function connectWebSocket() {

// ws = new WebSocket('ws://192.168.88.165/ws'); // DEV BOARD

ws = new WebSocket('ws://192.168.88.12/ws'); // ESP CHIP 1

ws.onopen = () => {

console.log('Connected to WebSocket');

};

ws.onmessage = (event) => {

console.log("Received message from ESP32: " + event.data);

};

ws.onerror = (error) => {

console.error("WebSocket error:", error);

};

ws.onclose = (event) => {

console.log("WebSocket closed, attempting to reconnect...");

setTimeout(connectWebSocket, 1000); // Retry connection after 1 second

};

}

// Initialize WebSocket connection

connectWebSocket();

// Function to send commands via WebSocket

function sendCommand(command) {

if (ws.readyState === WebSocket.OPEN) {

ws.send(command);

} else {

console.log("WebSocket not connected. Command not sent:", command);

}

}

// Button click handlers

const buttons = document.querySelectorAll('.button');

buttons.forEach(button => {

button.addEventListener('mousedown', () => {

button.classList.add('active');

const commandMap = {

up: 'U',

down: 'D',

left: 'L',

right: 'R',

stop: 'S'

};

sendCommand(commandMap[button.id]);

});

button.addEventListener('mouseup', () => button.classList.remove('active'));

button.addEventListener('mouseleave', () => button.classList.remove('active'));

});

// Keyboard event listeners in the desired format

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

console.log('Key pressed:', event.key);

if (event.key === 'ArrowUp') {

sendCommand('U');

} else if (event.key === 'ArrowDown') {

sendCommand('D');

} else if (event.key === 'ArrowLeft') {

sendCommand('L');

} else if (event.key === 'ArrowRight') {

sendCommand('R');

} else if (event.code === 'KeyS') {

sendCommand('S');

}

});

document.addEventListener('keyup', () => {

buttons.forEach(button => button.classList.remove('active'));

});

</script>

</body>

</html>