const sendBtn = document.getElementById('send-btn');

const userInput = document.getElementById('user-input');

const responseArea = document.getElementById('alien-response');

const dimensionInfo = document.getElementById('current-dimension');

const dimensionBtns = document.querySelectorAll('.dimension-btn');

let currentDimension = "1D"; // Default dimension

// Change dimension when clicking buttons

dimensionBtns.forEach(btn => {

btn.addEventListener('click', (e) => {

currentDimension = e.target.getAttribute('data-dimension');

dimensionInfo.textContent = `Dimension: ${currentDimension}`;

});

});

// Send the message and get a response

sendBtn.addEventListener('click', () => {

const message = userInput.value.trim();

if (message === "") return;

responseArea.textContent = "Processing...";

userInput.value = ""; // Clear the input field

setTimeout(() => {

const alienResponse = getAlienResponse(message, currentDimension);

responseArea.textContent = alienResponse;

}, 1000); // Simulate alien processing time

});

// Function to get a response based on the dimension

function getAlienResponse(message, dimension) {

let response = "";

switch (dimension) {

case "1D":

response = `Binary signal received: ${message.split('').map(c => c.charCodeAt(0).toString(2)).join(' ')}`;

break;

case "2D":

response = `Symbolic echo: ⬛🔳⚪. You speak in shapes.`;

break;

case "3D":

response = `Cause and effect: "I hear you, yet feel your voice’s echo through time."`;

break;

case "4D":

response = `Time-folded response: "You have already asked this in another time loop."`;

break;

case "5D":

response = `Choice: "Will you choose to understand or to divide?"`;

break;

case "?D":

response = `Noise layer: "Transmission corrupted. Disconnection imminent."`;

break;

default:

response = `Unknown dimension. Response unavailable.`;

}

return response;

}