<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Code Translator</title>

<style>

body { font-family: Arial, sans-serif; text-align: center; margin: 50px; }

textarea { width: 45%; height: 300px; margin: 10px; }

#translateBtn { margin: 20px; }

.camera-icon { position: absolute; top: 10px; right: 10px; cursor: pointer; }

</style>

</head>

<body>

<h1>Code Translator</h1>

<div>

<textarea id="inputCode" placeholder="Paste your code here"></textarea>

<textarea id="outputCode" placeholder="Translated code will appear here"></textarea>

</div>

<button id="translateBtn">Translate</button>

<img src="camera-icon.png" alt="Camera" class="camera-icon" id="cameraIcon">

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

</body>

</html>

document.getElementById('translateBtn').addEventListener('click', function() {

let inputCode = document.getElementById('inputCode').value;

let outputCode = translateCode(inputCode);

document.getElementById('outputCode').value = outputCode;

});

document.getElementById('cameraIcon').addEventListener('click', function() {

// Logic to take a picture and translate the code

alert('Camera functionality is not implemented yet.');

});

function translateCode(code) {

// Determine the type of code and translate accordingly

// This is a simple detection based on the content, which can be enhanced

if (isAvoskCode(code)) {

return translateAvosk(code);

} else if (isMorseCode(code)) {

return translateMorse(code);

} else if (isTicTacToeCode(code)) {

return translateTicTacToe(code);

} else if (isTelephoneCode(code)) {

return translateTelephone(code);

} else if (isInverseCode(code)) {

return translateInverse(code);

} else {

return "Unrecognized code format.";

}

}

function isAvoskCode(code) {

return /^[a-z\s]+$/i.test(code); // Simple check for letters and spaces

}

function translateAvosk(code) {

const avosk = { a: 'k', b: 'l', c: 'm', d: 'n', e: 'o', f: 'p', g: 'q', h: 'r', i: 's', j: 't', k: 'u', l: 'v', m: 'w', n: 'x', o: 'y', p: 'z', q: 'a', r: 'b', s: 'c', t: 'd', u: 'e', v: 'f', w: 'g', x: 'h', y: 'i', z: 'j' };

return code.toLowerCase().split('').map(char => avosk[char] || char).join('');

}

function isMorseCode(code) {

return /^[\.\-\/\s]+$/.test(code); // Check for dots, dashes, slashes, and spaces

}

function translateMorse(code) {

const morse = {

'.-': 'A', '-...': 'B', '-.-.': 'C', '-..': 'D', '.': 'E', '..-.': 'F', '--.': 'G', '....': 'H', '..': 'I', '.---': 'J',

'-.-': 'K', '.-..': 'L', '--': 'M', '-.': 'N', '---': 'O', '.--.': 'P', '--.-': 'Q', '.-.': 'R', '...': 'S', '-': 'T',

'..-': 'U', '...-': 'V', '.--': 'W', '-..-': 'X', '-.--': 'Y', '--..': 'Z', '-----': '0', '.----': '1', '..---': '2',

'...--': '3', '....-': '4', '.....': '5', '-....': '6', '--...': '7', '---..': '8', '----.': '9'

};

return code.split(' ').map(symbol => morse[symbol] || ' ').join('');

}

function isTicTacToeCode(code) {

return /^[ox\s]+$/i.test(code); // Simple check for 'o', 'x', and spaces

}

function translateTicTacToe(code) {

// Placeholder translation for Tic Tac Toe code

return code; // Implement the actual translation logic here

}

function isTelephoneCode(code) {

return /^[2-9\s]+$/.test(code); // Check for digits 2-9 and spaces

}

function translateTelephone(code) {

const telephone = {

'2': 'ABC', '3': 'DEF', '4': 'GHI', '5': 'JKL', '6': 'MNO', '7': 'PQRS', '8': 'TUV', '9': 'WXYZ'

};

return code.split('').map(digit => telephone[digit] || digit).join('');

}

function isInverseCode(code) {

return /^[a-zA-Z\s]+$/.test(code); // Simple check for letters and spaces

}

function translateInverse(code) {

const inverse = { a: 'z', b: 'y', c: 'x', d: 'w', e: 'v', f: 'u', g: 't', h: 's', i: 'r', j: 'q', k: 'p', l: 'o', m: 'n', n: 'm', o: 'l', p: 'k', q: 'j', r: 'i', s: 'h', t: 'g', u: 'f', v: 'e', w: 'd', x: 'c', y: 'b', z: 'a' };

return code.toLowerCase().split('').map(char => inverse[char] || char).join('');

}