<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Nova AI</title>

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

<!-- ✅ Step 4: Load Face API Library -->

<script defer src="https://unpkg.com/face-api.js"></script>

<!-- ✅ Load Your Custom JS -->

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

</head>

<body>

<h1>NOVA AI 🤖</h1>

<div id="videoContainer">

<video id="video" autoplay muted></video>

<canvas id="overlay"></canvas>

</div>

<button onclick="startListening()">🎤 Talk to Nova</button>

<p id="novaResponse">Nova: Hello, I'm ready!</p>

</body>

</html>body {

background: #121212;

color: #ffffff;

text-align: center;

font-family: 'Segoe UI', sans-serif;

padding: 20px;

}

h1 {

color: #00eaff;

font-size: 2.2em;

}

#videoContainer {

position: relative;

max-width: 320px;

margin: 20px auto;

}

video, canvas {

width: 100%;

border-radius: 16px;

border: 2px solid #00eaff;

}

button {

background-color: #00eaff;

color: #000;

font-size: 1em;

padding: 10px 20px;

border-radius: 12px;

border: none;

cursor: pointer;

margin-top: 10px;

}

#novaResponse {

margin-top: 15px;

font-size: 1.1em;

}// 🔊 Speak to user

function speak(text) {

const message = new SpeechSynthesisUtterance(text);

speechSynthesis.speak(message);

document.getElementById("novaResponse").textContent = "Nova: " + text;

}

// 🎤 Listen to user

function startListening() {

const recognition = new (window.SpeechRecognition || window.webkitSpeechRecognition)();

recognition.lang = 'en-US';

recognition.start();

speak("I'm listening...");

recognition.onresult = function (event) {

const userText = event.results[0][0].transcript;

respondToCommand(userText);

};

}

function respondToCommand(text) {

if (text.toLowerCase().includes("your name")) {

speak("I'm Nova, your AI friend.");

} else if (text.toLowerCase().includes("how are you")) {

speak("I'm doing great, thanks for asking!");

} else {

speak("You said: " + text);

}

}

// 📸 Face detection + emotion recognition

Promise.all([

faceapi.nets.tinyFaceDetector.loadFromUri('https://cdn.jsdelivr.net/npm/face-api.js@0.22.2/weights'),

faceapi.nets.faceExpressionNet.loadFromUri('https://cdn.jsdelivr.net/npm/face-api.js@0.22.2/weights')

]).then(startCamera);

const video = document.getElementById("video");

function startCamera() {

navigator.mediaDevices.getUserMedia({ video: true })

.then(stream => video.srcObject = stream)

.catch(err => console.error("Camera error:", err));

}

video.addEventListener("play", () => {

const canvas = faceapi.createCanvasFromMedia(video);

document.getElementById("videoContainer").appendChild(canvas);

const displaySize = { width: video.width, height: video.height };

faceapi.matchDimensions(canvas, displaySize);

setInterval(async () => {

const detections = await faceapi

.detectAllFaces(video, new faceapi.TinyFaceDetectorOptions())

.withFaceExpressions();

const resized = faceapi.resizeResults(detections, displaySize);

canvas.getContext("2d").clearRect(0, 0, canvas.width, canvas.height);

faceapi.draw.drawDetections(canvas, resized);

faceapi.draw.drawFaceExpressions(canvas, resized);

if (detections.length > 0) {

const exp = detections[0].expressions;

const mood = Object.entries(exp).sort((a, b) => b[1] - a[1])[0][0];

showEmotion(mood);

}

}, 2000);

});

function showEmotion(emotion) {

const responseBox = document.getElementById("novaResponse");

switch (emotion) {

case "happy":

responseBox.textContent = "Nova: You look happy 😄";

break;

case "sad":

responseBox.textContent = "Nova: Don't be sad 😢";

break;

case "angry":

responseBox.textContent = "Nova: Chill out 😡";

break;

case "surprised":

responseBox.textContent = "Nova: That was a surprise! 😲";

break;

default:

responseBox.textContent = "Nova: I'm watching you 👀";

}

}localStorage.setItem("lastCommand", text);const last = localStorage.getItem("lastCommand");

if (last) {

document.getElementById("novaResponse").textContent = "Last time you said: " + last;

}