1.HTML – CODE

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Weather Finder</title>

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

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">

</head>

<body>

    <div class="container">

        <header>

            <h1><i class="fas fa-cloud-sun"></i> Weather Finder</h1>

            <p>Get current weather information for any city worldwide</p>

        </header>

        <div class="search-container">

            <div class="search-box">

                <input type="text" id="cityInput" placeholder="Enter city name..." autocomplete="off">

                <button id="searchBtn"><i class="fas fa-search"></i></button>

            </div>

        </div>

        <div class="weather-container" id="weatherContainer">

            <div class="loading" id="loading">

                <i class="fas fa-spinner fa-spin"></i>

                <p>Loading weather data...</p>

            </div>

            <div class="weather-card" id="weatherCard">

                <div class="weather-header">

                    <h2 id="cityName"></h2>

                    <p id="currentDate"></p>

                </div>

                <div class="weather-main">

                    <div class="temperature">

                        <span id="temp"></span>

                        <div class="temp-unit">

                            <button id="celsiusBtn" class="active">°C</button>

                            <button id="fahrenheitBtn">°F</button>

                        </div>

                    </div>

                    <div class="weather-icon">

                        <img id="weatherIcon" src="" alt="Weather Icon">

                        <p id="weatherDescription"></p>

                    </div>

                </div>

                <div class="weather-details">

                    <div class="detail-item">

                        <i class="fas fa-eye"></i>

                        <span>Feels like</span>

                        <span id="feelsLike"></span>

                    </div>

                    <div class="detail-item">

                        <i class="fas fa-tint"></i>

                        <span>Humidity</span>

                        <span id="humidity"></span>

                    </div>

                    <div class="detail-item">

                        <i class="fas fa-wind"></i>

                        <span>Wind Speed</span>

                        <span id="windSpeed"></span>

                    </div>

                    <div class="detail-item">

                        <i class="fas fa-thermometer-half"></i>

                        <span>Pressure</span>

                        <span id="pressure"></span>

                    </div>

                    <div class="detail-item">

                        <i class="fas fa-low-vision"></i>

                        <span>Visibility</span>

                        <span id="visibility"></span>

                    </div>

                    <div class="detail-item">

                        <i class="fas fa-sun"></i>

                        <span>UV Index</span>

                        <span id="uvIndex"></span>

                    </div>

                </div>

            </div>

            <div class="error-message" id="errorMessage">

                <i class="fas fa-exclamation-triangle"></i>

                <h3>Oops! Something went wrong</h3>

                <p id="errorText"></p>

                <button id="retryBtn">Try Again</button>

            </div>

        </div>

        <div class="api-info">

            <p><strong>Note:</strong> To use this weather app, you need to:</p>

            <ol>

                <li>Get a free API key from <a href="https://openweathermap.org/api" target="\_blank">OpenWeatherMap</a></li>

                <li>Replace 'YOUR\_API\_KEY' in the script.js file with your actual API key</li>

            </ol>

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

2.CSS – CODE

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    background: linear-gradient(135deg, #74b9ff 0%, #0984e3 100%);

    min-height: 100vh;

    color: #333;

}

.container {

    max-width: 800px;

    margin: 0 auto;

    padding: 20px;

}

header {

    text-align: center;

    margin-bottom: 30px;

    color: white;

}

header h1 {

    font-size: 2.5rem;

    margin-bottom: 10px;

    text-shadow: 2px 2px 4px rgba(0,0,0,0.3);

}

header p {

    font-size: 1.1rem;

    opacity: 0.9;

}

.search-container {

    margin-bottom: 30px;

}

.search-box {

    display: flex;

    max-width: 400px;

    margin: 0 auto;

    background: white;

    border-radius: 25px;

    overflow: hidden;

    box-shadow: 0 4px 15px rgba(0,0,0,0.2);

}

#cityInput {

    flex: 1;

    padding: 15px 20px;

    border: none;

    outline: none;

    font-size: 1rem;

    background: transparent;

}

#cityInput::placeholder {

    color: #999;

}

#searchBtn {

    padding: 15px 20px;

    background: #0984e3;

    color: white;

    border: none;

    cursor: pointer;

    transition: background 0.3s ease;

}

#searchBtn:hover {

    background: #0770c4;

}

.weather-container {

    position: relative;

}

.loading {

    text-align: center;

    color: white;

    display: none;

}

.loading i {

    font-size: 2rem;

    margin-bottom: 10px;

}

.weather-card {

    background: rgba(255, 255, 255, 0.95);

    border-radius: 20px;

    padding: 30px;

    box-shadow: 0 8px 32px rgba(0,0,0,0.2);

    backdrop-filter: blur(10px);

    display: none;

    animation: slideUp 0.5s ease;

}

@keyframes slideUp {

    from {

        opacity: 0;

        transform: translateY(30px);

    }

    to {

        opacity: 1;

        transform: translateY(0);

    }

}

.weather-header {

    text-align: center;

    margin-bottom: 30px;

}

.weather-header h2 {

    font-size: 2rem;

    color: #333;

    margin-bottom: 5px;

}

.weather-header p {

    color: #666;

    font-size: 1rem;

}

.weather-main {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin-bottom: 30px;

    flex-wrap: wrap;

    gap: 20px;

}

.temperature {

    display: flex;

    align-items: center;

    gap: 15px;

}

#temp {

    font-size: 4rem;

    font-weight: bold;

    color: #0984e3;

}

.temp-unit {

    display: flex;

    flex-direction: column;

    gap: 5px;

}

.temp-unit button {

    padding: 5px 10px;

    border: 2px solid #0984e3;

    background: transparent;

    color: #0984e3;

    border-radius: 15px;

    cursor: pointer;

    transition: all 0.3s ease;

    font-size: 0.9rem;

}

.temp-unit button.active,

.temp-unit button:hover {

    background: #0984e3;

    color: white;

}

.weather-icon {

    text-align: center;

}

.weather-icon img {

    width: 100px;

    height: 100px;

}

.weather-icon p {

    margin-top: 10px;

    font-size: 1.1rem;

    color: #666;

    text-transform: capitalize;

}

.weather-details {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

    gap: 15px;

}

.detail-item {

    display: flex;

    align-items: center;

    gap: 10px;

    padding: 15px;

    background: rgba(116, 185, 255, 0.1);

    border-radius: 10px;

    border-left: 4px solid #0984e3;

}

.detail-item i {

    color: #0984e3;

    width: 20px;

    text-align: center;

}

.detail-item span:first-of-type {

    flex: 1;

    color: #666;

}

.detail-item span:last-of-type {

    font-weight: bold;

    color: #333;

}

.error-message {

    text-align: center;

    color: white;

    display: none;

    animation: slideUp 0.5s ease;

}

.error-message i {

    font-size: 3rem;

    margin-bottom: 15px;

    color: #ff6b6b;

}

.error-message h3 {

    margin-bottom: 10px;

    font-size: 1.5rem;

}

.error-message p {

    margin-bottom: 20px;

    opacity: 0.9;

}

#retryBtn {

    padding: 12px 25px;

    background: #ff6b6b;

    color: white;

    border: none;

    border-radius: 25px;

    cursor: pointer;

    font-size: 1rem;

    transition: background 0.3s ease;

}

#retryBtn:hover {

    background: #ff5252;

}

.api-info {

    margin-top: 30px;

    background: rgba(255, 255, 255, 0.1);

    padding: 20px;

    border-radius: 10px;

    color: white;

    backdrop-filter: blur(10px);

}

.api-info p {

    margin-bottom: 10px;

    font-weight: bold;

}

.api-info ol {

    margin-left: 20px;

}

.api-info li {

    margin-bottom: 5px;

}

.api-info a {

    color: #74b9ff;

    text-decoration: none;

}

.api-info a:hover {

    text-decoration: underline;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

    .container {

        padding: 15px;

    }

    header h1 {

        font-size: 2rem;

    }

    .weather-main {

        flex-direction: column;

        text-align: center;

    }

    #temp {

        font-size: 3rem;

    }

    .weather-details {

        grid-template-columns: 1fr;

    }

    .detail-item {

        justify-content: space-between;

    }

}

@media (max-width: 480px) {

    .weather-card {

        padding: 20px;

    }

    header h1 {

        font-size: 1.8rem;

    }

    #temp {

        font-size: 2.5rem;

    }

    .search-box {

        margin: 0 10px;

    }

}

3.JS – CODE

// Weather App JavaScript

// Replace 'YOUR\_API\_KEY' with your actual OpenWeatherMap API key

const API\_KEY = '1aafa0979ba6c5fc7f15ec94067c6dc7';

const BASE\_URL = 'https://api.openweathermap.org/data/2.5/weather';

// DOM Elements

const cityInput = document.getElementById('cityInput');

const searchBtn = document.getElementById('searchBtn');

const weatherContainer = document.getElementById('weatherContainer');

const loading = document.getElementById('loading');

const weatherCard = document.getElementById('weatherCard');

const errorMessage = document.getElementById('errorMessage');

const retryBtn = document.getElementById('retryBtn');

// Weather data elements

const cityName = document.getElementById('cityName');

const currentDate = document.getElementById('currentDate');

const temp = document.getElementById('temp');

const weatherIcon = document.getElementById('weatherIcon');

const weatherDescription = document.getElementById('weatherDescription');

const feelsLike = document.getElementById('feelsLike');

const humidity = document.getElementById('humidity');

const windSpeed = document.getElementById('windSpeed');

const pressure = document.getElementById('pressure');

const visibility = document.getElementById('visibility');

const uvIndex = document.getElementById('uvIndex');

// Temperature unit buttons

const celsiusBtn = document.getElementById('celsiusBtn');

const fahrenheitBtn = document.getElementById('fahrenheitBtn');

// Global variables

let currentWeatherData = null;

let isCelsius = true;

// Event Listeners

searchBtn.addEventListener('click', handleSearch);

cityInput.addEventListener('keypress', (e) => {

    if (e.key === 'Enter') {

        handleSearch();

    }

});

retryBtn.addEventListener('click', handleSearch);

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

    if (!isCelsius && currentWeatherData) {

        isCelsius = true;

        updateTemperatureDisplay();

        updateTempButtons();

    }

});

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

    if (isCelsius && currentWeatherData) {

        isCelsius = false;

        updateTemperatureDisplay();

        updateTempButtons();

    }

});

// Functions

async function handleSearch() {

    const city = cityInput.value.trim();

    if (!city) {

        showError('Please enter a city name');

        return;

    }

   /\* if (API\_KEY === '1aafa0979ba6c5fc7f15ec94067c6dc7') {

        showError('Please add your OpenWeatherMap API key to the script.js file');

        return;

    } \*/

    await getWeatherData(city);

}

async function getWeatherData(city) {

    showLoading();

    try {

        const url = `${BASE\_URL}?q=${encodeURIComponent(city)}&appid=${API\_KEY}&units=metric`;

        const response = await fetch(url);

        if (!response.ok) {

            if (response.status === 404) {

                throw new Error('City not found. Please check the spelling and try again.');

            } else if (response.status === 401) {

                throw new Error('Invalid API key. Please check your OpenWeatherMap API key.');

            } else {

                throw new Error('Failed to fetch weather data. Please try again later.');

            }

        }

        const data = await response.json();

        currentWeatherData = data;

        displayWeatherData(data);

    } catch (error) {

        console.error('Error fetching weather data:', error);

        showError(error.message);

    }

}

function displayWeatherData(data) {

    // Update city name and date

    cityName.textContent = `${data.name}, ${data.sys.country}`;

    currentDate.textContent = new Date().toLocaleDateString('en-US', {

        weekday: 'long',

        year: 'numeric',

        month: 'long',

        day: 'numeric'

    });

    // Update weather icon and description

    const iconCode = data.weather[0].icon;

    weatherIcon.src = `https://openweathermap.org/img/wn/${iconCode}@2x.png`;

    weatherIcon.alt = data.weather[0].description;

    weatherDescription.textContent = data.weather[0].description;

    // Update temperature

    updateTemperatureDisplay();

    // Update weather details

    feelsLike.textContent = isCelsius ?

        `${Math.round(data.main.feels\_like)}°C` :

        `${Math.round(celsiusToFahrenheit(data.main.feels\_like))}°F`;

    humidity.textContent = `${data.main.humidity}%`;

    windSpeed.textContent = `${data.wind.speed} m/s`;

    pressure.textContent = `${data.main.pressure} hPa`;

    visibility.textContent = data.visibility ? `${(data.visibility / 1000).toFixed(1)} km` : 'N/A';

    // UV Index is not available in current weather API, so we'll show N/A

    uvIndex.textContent = 'N/A';

    showWeatherCard();

}

function updateTemperatureDisplay() {

    if (!currentWeatherData) return;

    const tempValue = isCelsius ?

        Math.round(currentWeatherData.main.temp) :

        Math.round(celsiusToFahrenheit(currentWeatherData.main.temp));

    temp.textContent = `${tempValue}°`;

    // Update feels like temperature

    const feelsLikeValue = isCelsius ?

        Math.round(currentWeatherData.main.feels\_like) :

        Math.round(celsiusToFahrenheit(currentWeatherData.main.feels\_like));

    feelsLike.textContent = `${feelsLikeValue}°${isCelsius ? 'C' : 'F'}`;

}

function updateTempButtons() {

    celsiusBtn.classList.toggle('active', isCelsius);

    fahrenheitBtn.classList.toggle('active', !isCelsius);

}

function celsiusToFahrenheit(celsius) {

    return (celsius \* 9/5) + 32;

}

function showLoading() {

    hideAllStates();

    loading.style.display = 'block';

}

function showWeatherCard() {

    hideAllStates();

    weatherCard.style.display = 'block';

}

function showError(message) {

    hideAllStates();

    document.getElementById('errorText').textContent = message;

    errorMessage.style.display = 'block';

}

function hideAllStates() {

    loading.style.display = 'none';

    weatherCard.style.display = 'none';

    errorMessage.style.display = 'none';

}

// Initialize the app

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

    // Set current date on page load

    currentDate.textContent = new Date().toLocaleDateString('en-US', {

        weekday: 'long',

        year: 'numeric',

        month: 'long',

        day: 'numeric'

    });

    // Focus on input field

    cityInput.focus();

    // Load default city (optional)

    // You can uncomment the line below to load a default city on page load

    // cityInput.value = 'London';

    // handleSearch();

});

// Additional utility functions

function getCurrentPosition() {

    return new Promise((resolve, reject) => {

        if (!navigator.geolocation) {

            reject(new Error('Geolocation is not supported by this browser.'));

            return;

        }

        navigator.geolocation.getCurrentPosition(resolve, reject);

    });

}

// Function to get weather by current location (optional feature)

async function getWeatherByLocation() {

    try {

        showLoading();

        const position = await getCurrentPosition();

        const { latitude, longitude } = position.coords;

        const url = `${BASE\_URL}?lat=${latitude}&lon=${longitude}&appid=${API\_KEY}&units=metric`;

        const response = await fetch(url);

        if (!response.ok) {

            throw new Error('Failed to fetch weather data for your location.');

        }

        const data = await response.json();

        currentWeatherData = data;

        displayWeatherData(data);

        cityInput.value = data.name;

    } catch (error) {

        console.error('Error getting location weather:', error);

        showError(error.message);

    }

}

// You can add a button to get current location weather if needed

// Example: <button onclick="getWeatherByLocation()">Use My Location</button>