**SOURCE CODE**

**JAVASCRIPT :**

const cityInput = document.querySelector(".city-input");

const searchButton = document.querySelector(".search-btn");

const locationButton = document.querySelector(".location-btn");

const currentWeatherDiv = document.querySelector(".current-weather");

const weatherCardsDiv = document.querySelector(".weather-cards");

const API\_KEY = "5f1a9773b38a2103c5e606577d124abc"; // API key for OpenWeatherMap API

const createWeatherCard = (cityName, weatherItem, index) => {

if(index === 0) { // HTML for the main weather card

return `<div class="details">

<h2>${cityName} (${weatherItem.dt\_txt.split(" ")[0]})</h2>

<h6>Temperature: ${(weatherItem.main.temp - 273.15).toFixed(2)}°C</h6>

<h6>Wind: ${weatherItem.wind.speed} M/s</h6>

<h6>Humidity: ${weatherItem.main.humidity}%</h6>

<h6>Pressure: ${weatherItem.main.pressure}hPa</h6>

</div>

<div class="icon">

<img src="https://openweathermap.org/img/wn/${weatherItem.weather[0].icon}@4x.png" alt="weather-icon">

<h6>${weatherItem.weather[0].description}</h6>

</div>`;

} else { // HTML for the other five day forecast card

return `<li class="card">

<h3>(${weatherItem.dt\_txt.split(" ")[0]})</h3>

<img src="https://openweathermap.org/img/wn/${weatherItem.weather[0].icon}@4x.png" alt="weather-icon">

<h6>Temp: ${(weatherItem.main.temp - 273.15).toFixed(2)}°C</h6>

<h6>Wind: ${weatherItem.wind.speed} M/s</h6>

<h6>Humidity: ${weatherItem.main.humidity}%</h6>

<h6>Pressure: ${weatherItem.main.pressure}hPa</h6>

<h6>Description: ${weatherItem.weather[0].description}</h6>

</li>`;

}

}

const getWeatherDetails = (cityName, latitude, longitude) => {

const WEATHER\_API\_URL = https://api.openweathermap.org/data/2.5/forecast?lat=${latitude}&lon=${longitude}&appid=${API\_KEY};

fetch(WEATHER\_API\_URL).then(response => response.json()).then(data => {

// Filter the forecasts to get only one forecast per day

const uniqueForecastDays = [];

const fiveDaysForecast = data.list.filter(forecast => {

const forecastDate = new Date(forecast.dt\_txt).getDate();

if (!uniqueForecastDays.includes(forecastDate)) {

return uniqueForecastDays.push(forecastDate);

}

});

// Clearing previous weather data

cityInput.value = "";

currentWeatherDiv.innerHTML = "";

weatherCardsDiv.innerHTML = "";

// Creating weather cards and adding them to the DOM

console.log(fiveDaysForecast);

fiveDaysForecast.forEach((weatherItem, index) => {

const html = createWeatherCard(cityName, weatherItem, index);

if (index === 0) {

currentWeatherDiv.insertAdjacentHTML("beforeend", html);

} else {

weatherCardsDiv.insertAdjacentHTML("beforeend", html);

}

});

}).catch(() => {

alert("An error occurred while fetching the weather forecast!");

});

}

const getCityCoordinates = () => {

const cityName = cityInput.value.trim();

if (cityName === "") return;

const API\_URL = https://api.openweathermap.org/geo/1.0/direct?q=${cityName}&limit=1&appid=${API\_KEY};

// Get entered city coordinates (latitude, longitude, and name) from the API response

fetch(API\_URL).then(response => response.json()).then(data => {

if (!data.length) return alert(No coordinates found for ${cityName});

const { lat, lon, name } = data[0];

getWeatherDetails(name, lat, lon);

}).catch(() => {

alert("An error occurred while fetching the coordinates!");

});

}

const getUserCoordinates = () => {

navigator.geolocation.getCurrentPosition(

position => {

const { latitude, longitude } = position.coords; // Get coordinates of user location

// Get city name from coordinates using reverse geocoding API

const API\_URL = https://api.openweathermap.org/geo/1.0/reverse?lat=${latitude}&lon=${longitude}&limit=1&appid=${API\_KEY};

fetch(API\_URL).then(response => response.json()).then(data => {

const { name } = data[0];

getWeatherDetails(name, latitude, longitude);

}).catch(() => {

alert("An error occurred while fetching the city name!");

});

},

error => { // Show alert if user denied the location permission

if (error.code === error.PERMISSION\_DENIED) {

alert("Geolocation request denied. Please reset location permission to grant access again.");

} else {

alert("Geolocation request error. Please reset location permission.");

}

});

}

locationButton.addEventListener("click", getUserCoordinates);

searchButton.addEventListener("click", getCityCoordinates);

cityInput.addEventListener("keyup", e => e.key === "Enter" && getCityCoordinates());