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

startClock();

let lat;

let lon;

const appId = '6ae9e4516899cbbe9c1a98da589136a5';

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

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

const buttons = document.querySelectorAll('button');

buttons.forEach(button => {

button.addEventListener('click', onButtonClicked);

});

navigator.geolocation.getCurrentPosition(pos => {

lat = pos.coords.latitude;

lon = pos.coords.longitude;

});

function onButtonClicked(evt) {

const buttonName = evt.target.dataset.id;

switch (buttonName) {

case 'getCurrentWx':

getCurrentWx();

break;

case 'getFiveDay':

getFiveDay();

break;

}

}

function getCurrentWx() {

if (!lat || !lon) {

console.error("Latitude and longitude are not available.");

return;

}

const url = `https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${appId}&units=imperial`;

fetch(url)

.then(r => r.json())

.then(wx => {

console.log(wx);

const locationName = wx.name;

const currentTemp = wx.main.temp;

const maxTemp = wx.main.temp\_max;

const humidity = wx.main.humidity;

let s = `

<h2>${locationName}</h2>

<div>Current Temp: ${currentTemp}&#8457;</div>

<div>Max Temp: ${maxTemp}&#8457;</div>

<div>Humidity: ${humidity}%</div>

`;

currentWxHolder.innerHTML = s;

})

.catch(e => {

console.error(e);

});

}

function getFiveDay() {

if (!lat || !lon) {

console.error("Latitude and longitude are not available.");

return;

}

const url = `https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lon}&cnt=5&appid=${appId}&units=imperial`;

fetch(url)

.then(r => r.json())

.then(wx => {

console.log(wx);

const locationName = wx.city.name;

const population = wx.city.population;

const f = wx.list;

const forecastArray = [];

for (let i = 0; i <= 2; i++) {

console.log(f[i]);

let w = new WeatherForecast(f[i].dt\_txt, f[i].main.temp, f[i].main.temp\_min, f[i].main.temp\_max);

forecastArray.push(w);

}

let h2 = document.createElement('h2');

let h3 = document.createElement('h3');

h2.innerHTML = locationName;

h3.innerHTML = '3 Hour Forecast';

fiveDayInfoHolder.appendChild(h2);

fiveDayInfoHolder.appendChild(h3);

forecastArray.forEach(forecast => {

let div = document.createElement('div');

let h4 = document.createElement('h3');

h4.innerHTML = forecast.getDayString();

div.appendChild(h4);

let d = document.createElement('div');

d.innerHTML = 'Forecast Time (UTC): ' + forecast.getDate().getHours() + " hrs.";

div.appendChild(d);

d = document.createElement('div');

d.innerHTML = 'Temperature: ' + forecast.getTemp() + '&#8457;';

div.appendChild(d);

d = document.createElement('div');

d.innerHTML = 'Max Temperature: ' + forecast.getMaxTemp() + '&#8457;';

div.appendChild(d);

d = document.createElement('div');

d.innerHTML = 'Min Temperature: ' + forecast.getMinTemp() + '&#8457;';

div.appendChild(d);

let hr = document.createElement('hr');

div.appendChild(hr);

fiveDayInfoHolder.append(div);

});

})

.catch(e => {

console.error(e);

});

}

});

class WeatherForecast {

constructor(d, t, tM, tX) {

console.log(d, t, tM, tX);

this.date = d;

this.temp = t;

this.tempMin = tM;

this.tempMax = tX;

}

getDate() {

return new Date(this.date);

}

getTemp() {

return this.temp;

}

getMaxTemp() {

return this.tempMax;

}

getMinTemp() {

return this.tempMin;

}

getDayString() {

let d = new Date(this.date);

let day = d.getDay();

let date = d.getDate();

let mon = d.getMonth() + 1;

let dayName = '';

switch (day) {

case 0:

dayName = 'Sun';

break;

case 1:

dayName = 'Mon';

break;

case 2:

dayName = 'Tue';

break;

case 3:

dayName = 'Wed';

break;

case 4:

dayName = 'Thu';

break;

case 5:

dayName = 'Fri';

break;

case 6:

dayName = 'Sat';

break;

}

let s = `${dayName}, ${mon}/${date}`;

return s;

}

}