WEEK-4

Explore the features of ES6 like arrow functions, callbacks, promises, async/await. Implement an application for reading the weather information from openweathermap.org and display the information in the form of a graph on the webpage.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>weather</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">


    //<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300,400,600,700">
    //<link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300">
    <link rel="stylesheet" href="styles.css">
</head>
<body>



<div class="container">
        <div class="weather-side">
            <div class="weather-gradient"></div>
            <div class="date-container">
```

```
                <h2 class="date-dayname"></h2>
                <span class="date-day"></span>
                <i class="fa-solid fa-location-dot"></i>
                <span class="location"></span>
            </div>
            <div class="weather-container">
                <span class="weather-icon"></span>
                <h1 class="weather-temp"></h1>
                <h3 class="weather-desc"></h3>
            </div>
        </div>
        <div class="info-side">
            <div class="today-info-container">
                <div class="today-info">
                    <div class="humidity">
                        <span class="title"><i class="fa-solid fa-droplet"></i>
HUMIDITY</span>
                        <span class="value"></span>
                        <div class="clear"></div>
                    </div>
                    <div class="wind">
                        <span class="title"><i class="fa-solid fa-wind"></i>
WIND</span>
                        <span class="value"></span>
                        <div class="clear"></div>
                    </div>
                </div>
            </div>
```

```
<div class="week-container">
    <ul class="week-list">
        <li>
            <span class="day-name"></span>
            <span class="day-temp"></span>
            <span class="day-icon"></span>
        </li>
        <li>
            <span class="day-name"></span>
            <span class="day-temp"></span>
            <span class="day-icon"></span>
        </li>
        <li>
            <span class="day-name"></span>
            <span class="day-temp"></span>
            <span class="day-icon"></span>
        </li>
        <li>
            <span class="day-name"></span>
            <span class="day-temp"></span>
            <span class="day-icon"></span>
        </li>
        <div class="clear"></div>
    </ul>
</div>
<div class="location-container">
```

```
                <input class="location-input" type="text" id="city" value="Kabul,
Afghanistan">
            </div>
        </div>

    </div>

     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></s
cript>
    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
    <script src="script.js"></script>

</body>
</html>
```

script.js

```
document.getElementById('city').addEventListener('input', function () {
    var city = this.value;
    getWeather(city);
  });

  async function getWeather() {
    try {
        var city = document.getElementById('city').value;
        console.log('city name:', city);
        const response = await
axios.get('https://api.openweathermap.org/data/2.5/forecast', {
      params: {
        q: city,
        appid: '54a57bc234ad752a4f59e59cd372201d',
        units: 'metric'
      },
        });
        const currentTemperature = response.data.list[0].main.temp;


        document.querySelector('.weather-temp').textContent =
Math.round(currentTemperature) + '°C';


        const forecastData = response.data.list;


        const dailyForecast = {};
        forecastData.forEach((data) => {
      const day = new Date(data.dt * 1000).toLocaleDateString('en-US', {
weekday: 'long' });
      if (!dailyForecast[day]) {
        dailyForecast[day] = {
          minTemp: data.main.temp_min,
          maxTemp: data.main.temp_max,
          description: data.weather[0].description,
          humidity: data.main.humidity,
          windSpeed: data.wind.speed,
          icon: data.weather[0].icon,
        };
      } else {
        dailyForecast[day].minTemp =
Math.min(dailyForecast[day].minTemp, data.main.temp_min);
        dailyForecast[day].maxTemp =
Math.max(dailyForecast[day].maxTemp, data.main.temp_max);
      }
    });

    document.querySelector('.date-dayname').textContent = new
Date().toLocaleDateString('en-US', { weekday: 'long' });

    const date = new Date().toUTCString();
    const extractedDateTime = date.slice(5, 16);
    document.querySelector('.date-day').textContent =
extractedDateTime.toLocaleString('en-US');

    const currentWeatherIconCode = dailyForecast[new
Date().toLocaleDateString('en-US', { weekday: 'long' })].icon;
    const weatherIconElement = document.querySelector('.weather-icon');
    weatherIconElement.innerHTML =
getWeatherIcon(currentWeatherIconCode);

    document.querySelector('.location').textContent = response.data.city.name;
    document.querySelector('.weather-desc').textContent = dailyForecast[new
Date().toLocaleDateString('en-US', { weekday: 'long' })].description.split('
').map(word => word.charAt(0).toUpperCase() + word.slice(1)).join(' ');
```

```
      document.querySelector('.humidity .value').textContent =
dailyForecast[new Date().toLocaleDateString('en-US', { weekday: 'long'
})].humidity + ' %';
      document.querySelector('.wind .value').textContent = dailyForecast[new
Date().toLocaleDateString('en-US', { weekday: 'long' })].windSpeed + ' m/s';



      const dayElements = document.querySelectorAll('.day-name');
      const tempElements = document.querySelectorAll('.day-temp');
      const iconElements = document.querySelectorAll('.day-icon');

      dayElements.forEach((dayElement, index) => {
        const day = Object.keys(dailyForecast)[index];
        const data = dailyForecast[day];
        dayElement.textContent = day;
        tempElements[index].textContent = `${Math.round(data.minTemp)}º /
${Math.round(data.maxTemp)}º`;
        iconElements[index].innerHTML = getWeatherIcon(data.icon);
      });

    } catch (error) {
      console.error('Error fetching weather data:', error.message);
    }
  }

  function getWeatherIcon(iconCode) {
    const iconBaseUrl = 'https://openweathermap.org/img/wn/';
    const iconSize = '@2x.png';
    return `<img src="${iconBaseUrl}${iconCode}${iconSize}" alt="Weather Icon">`;
  }

  document.addEventListener("DOMContentLoaded", function () {
    getWeather();
    setInterval(getWeather, 900000);
  });
```

styles.css

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,700,900&display=swap');

:root {
    --gradient: linear-gradient(135deg, #72EDF2 10%, #5151E5 100%);
}

* {
    box-sizing: border-box;
    line-height: 1.25em;
}

.clear {
    clear: both;
}
```

```
body {
    margin: 0;
    width: 100%;
    height: 100vh;
    font-family: 'Montserrat', sans-serif;

    display: flex;
    align-items: center;
    justify-content: center;
}

.container {
    border-radius: 25px;
    box-shadow: 0 0 70px -10px rgba(0, 0, 0, 0.2);
    background-color: #222831;
    color: #ffffff;
    height: 400px;
}

.weather-side {
    position: relative;
    height: 100%;
    border-radius: 25px;
    background-image: url("https://img.freepik.com/free-photo/sunset-beach-sea-
wave_1150-11145.jpg");
    width: 300px;
    box-shadow: 0 0 20px -10px rgba(0, 0, 0, 0.2);
    transition: transform 300ms ease;
    transform: translateZ(0) scale(1.02) perspective(1000px);
    float: left;
}

.weather-side:hover {
    transform: scale(1.1) perspective(1500px) rotateY(10deg);
}

.weather-gradient {
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    background-image: var(--gradient);
    border-radius: 25px;
    opacity: 0.4;
}

.date-container {
    position: absolute;
    top: 25px;
    left: 25px;
}

.date-dayname {
    margin: 0;
```

```
}

.date-day {
   display: block;
}

.location {
   display: inline-block;
   margin-top: 10px;
}

.location-icon {
   display: inline-block;
   height: 0.8em;
   width: auto;
   margin-right: 5px;
}

.weather-container {
   position: absolute;
   bottom: 25px;
   left: 25px;
}

.weather-icon img {
   filter: drop-shadow(0 0 2px #fff);
   width: 100%;
```

```
}

.weather-temp {
   margin: 0;
   font-weight: 700;
   font-size: 4em;
}

.weather-desc {
   margin: 0;
}

.info-side {
   position: relative;
   float: left;
   height: 100%;
   padding-top: 25px;
}

.today-info {
   padding: 15px;
   margin: 0 25px 25px 25px;
   box-shadow: 0 0 50px -5px rgba(0, 0, 0);
   border-radius: 10px;
}

.today-info>div:not(:last-child) {
    margin: 0 0 10px 0;
}

.today-info>div .title {
    float: left;
    font-weight: 700;
}

.today-info>div .value {
    float: right;
}

.week-list {
    list-style-type: none;
    padding: 0;
    margin: 10px 35px;
    box-shadow: 0 0 50px -5px rgba(0, 0, 0, 0.25);
    border-radius: 10px;
}

.week-list>li {
    float: left;
    padding: 15px;
    cursor: pointer;
    transition: 200ms ease;
    border-radius: 10px;
}
```

```
.week-list>li:hover {
   transform: scale(1.1);
   background: #fff;
   color: #222831;
   box-shadow: 0 0 40px -5px rgba(0, 0, 0, 0.2)
}

.week-list>li.active {
   background: #fff;
   color: #222831;
   border-radius: 10px;
}

.week-list>li .day-name {
   display: block;
   margin: 10px 0 0 0;
   text-align: center;
}

.week-list>li .day-icon {
   display: flex;
   margin-top: 20px;
   justify-content: center;
   align-items: center;
   height: 30px;
   width: auto;
```

```
    margin: 0 auto;
    font-size: 35px;
}

.week-list>li .day-icon img {
    filter: drop-shadow(0 0 2px white);
}

.week-list>li .day-temp {
    display: block;
    text-align: center;
    margin-top: 15px;
    font-weight: 700;
    margin-bottom: 25px;
}

.location-container {
    padding: 25px 35px;
}

.location-input {
    position: relative;
    width: 100%;
    border: none;
    border-radius: 25px;
    padding: 10px;
    font-family: 'Montserrat', sans-serif;
    background-image: var(--gradient);
    color: #000000;
    font-weight: 700;
    box-shadow: 0 0 30px -5px rgba(0, 0, 0, 0.25);
    transition: transform 200ms ease;
}

.location-input:hover {
    transform: scale(0.95);
}
```