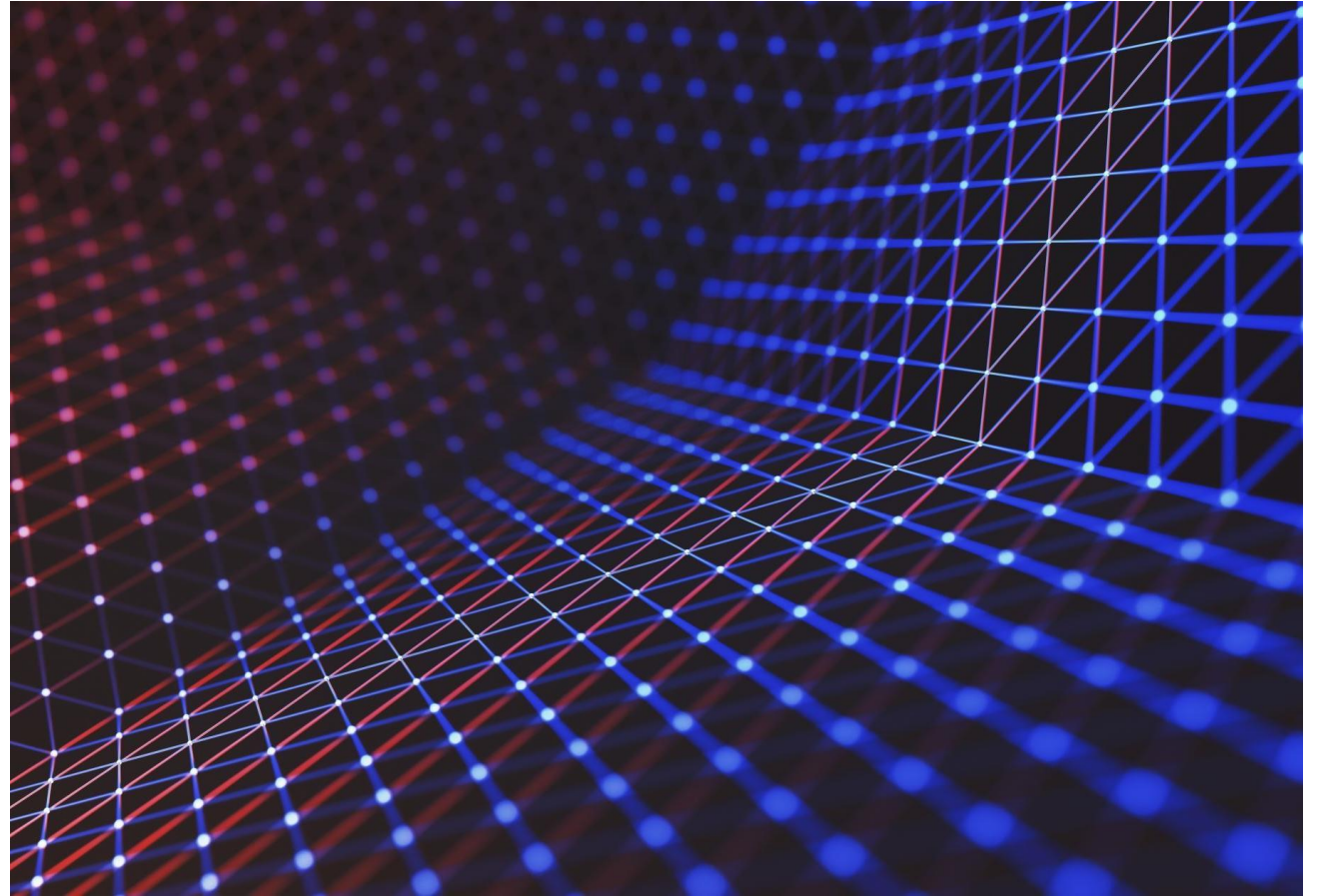


# Weather app project documentation

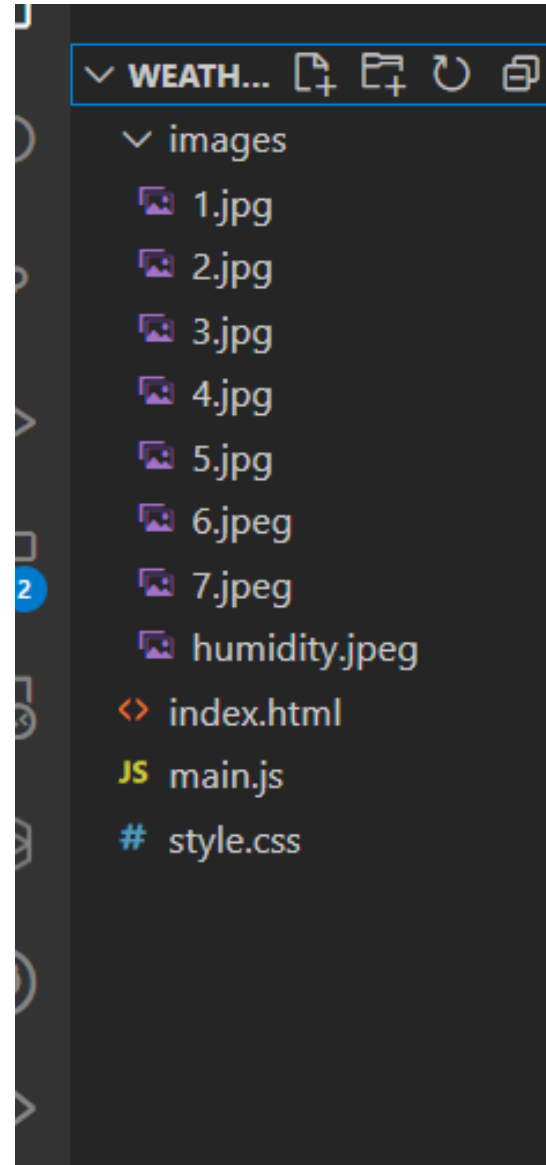
# Technologies used

HTML/CSS

JS



# Project structure



# HTML Key components

```
<div class="container">
  <div class="current-info">
    <div class="date-container">
      <div class="time" id="time">
        12:30 <span id="am-pm">PM</span>
      </div>
      <div class="date" id="date">
        Monday, 6 Nov
      </div>
    </div>
  </div>
</div>
```

- Показывает текущее время и дату:

```

<div class="container1">
  <div class="search-box">
    <i class="fa-solid fa-location-dot"></i>
    <input type="text" placeholder="Enter your location" />
    <button class="fa-solid fa-magnifying-glass"></button>
  </div>
  <div class="error">
    <p>Invalid city name</p>
  </div>
  <div class="weather">
    <div class="weather-image">
      <i class="fa-solid fa-cloud"></i>
    </div>
    <h1 class="temp">22 &#8451</h1>
    <h2 class="city">New York</h2>
    <div class="details">
      <div class="col">
        <i class="fa-solid fa-water"></i>
        <div>
          <p class="humidity">50%</p>
          <p>Humidity</p>
        </div>
      </div>
      <div class="col">
        <i class="fa-solid fa-wind"></i>
        <div>
          <p class="wind">15 km/h</p>
          <p>Wind Speed</p>
        </div>
      </div>
    </div>
  </div>
</div>

```

- Главный компонент страницы. Содержит в себе блок с input и при вводе правильного названия города, отображает информацию с api.

# JS

- Данный код выполняется каждую секунду и возвращает текущие дату и время

```
setInterval(() =>{  
  const time=new Date();  
  const month=time.getMonth();  
  const date=time.getDate();  
  const day=time.getDay();  
  const hour=time.getHours();  
  const hoursFormatted=hour>=13?hour%12:hour  
  const minute=time.getMinutes();  
  const ampm=hour>=12?'PM':'AM';  
  time.innerHTML=hoursFormatted+':'+minute+`<span id="am-pm">${ampm}</span>`  
  date.innerHTML=days[day]+' '+date+' '+months[month]  
},1000)
```

В данном коде имеется массив с картинками и функция changeBackground, которая вызывается каждые 7 секунд

```
var imageNames = [
  "images/1.jpg",
  "images/2.jpg",
  "images/3.jpg",
  "images/4.jpg",
  "images/5.jpg",
  "images/6.jpeg",
  "images/7.jpeg"
];

var currentIndex = 0;
var container = document.querySelector(".container");
var body = document.body;

function changeBackground() {
  if (currentIndex < imageNames.length) {
    var imageUrl = imageNames[currentIndex];
    body.style.background = "url(" + imageUrl + ") no-repeat center center / cover";
    currentIndex++;
  } else {
    currentIndex = 0;
    var imageUrl = imageNames[currentIndex];
    body.style.background = "url(" + imageUrl + ") no-repeat center center / cover";
    body.style.filter = "saturate(7) sepia(100%) contrast(180%)";

    // Используем [0], чтобы обратиться к первому элементу в коллекции
    if (imageUrl === 'imgCSS/1.jpg' || imageUrl === 'imgCSS/4.jpg') {
      container[0].style.color = "red";
    } else {
      container[0].style.color = "";
    }
  }
}

window.addEventListener('load', changeBackground);

setInterval(changeBackground, 7000);
```

```
const apiKey = "6814a81120bcae7f51bdf309cea0cce5";  
const apiUrl = `https://api.openweathermap.org/data/2.5/weather?units=metric&q=`;  
const searchInput = document.querySelector(".search-box input");  
const searchButton = document.querySelector(".search-box button");  
const weatherIcon = document.querySelector(".weather-image i");  
const weather = document.querySelector(".weather");  
const errorText = document.querySelector(".error");
```

## Setting variables

- Задаем переменные для работы с api. Также я сгенерировал api ключ с сайта openweathermap, и оттуда же ссылка на саму api.



- Главный класс для работы с api, в котором мы принимаем данные с api и подставляем их куда нужно.

```
async function checkWeather(city) {
  const response = await fetch(apiUrl + city + `&appid=${apiKey}`);
  if (response.status === 404) {
    errorText.style.display = "block";
    weather.style.display = "none";
  } else {
    const data = await response.json();
    console.log(data);
    document.querySelector(".city").innerHTML = data.name;
    document.querySelector(".temp").innerHTML =
      Math.round(data.main.temp) + "°C";
    document.querySelector(".humidity").innerHTML = data.main.humidity + "%";
    document.querySelector(".wind").innerHTML = data.wind.speed + " km/h";

    if (data.weather[0].main === "Clear") {
      weatherIcon.className = "fa-solid fa-sun";
    } else if (data.weather[0].main === "Rain") {
      weatherIcon.className = "fa-solid fa-cloud-rain";
    } else if (data.weather[0].main === "Mist") {
      weatherIcon.className = "fa-solid fa-cloud-mist";
    } else if (data.weather[0].main === "Drizzle") {
      weatherIcon.className = "fa-solid fa-cloud-drizzle";
    }
    weather.style.display = "block";
    errorText.style.display = "none";
  }
}

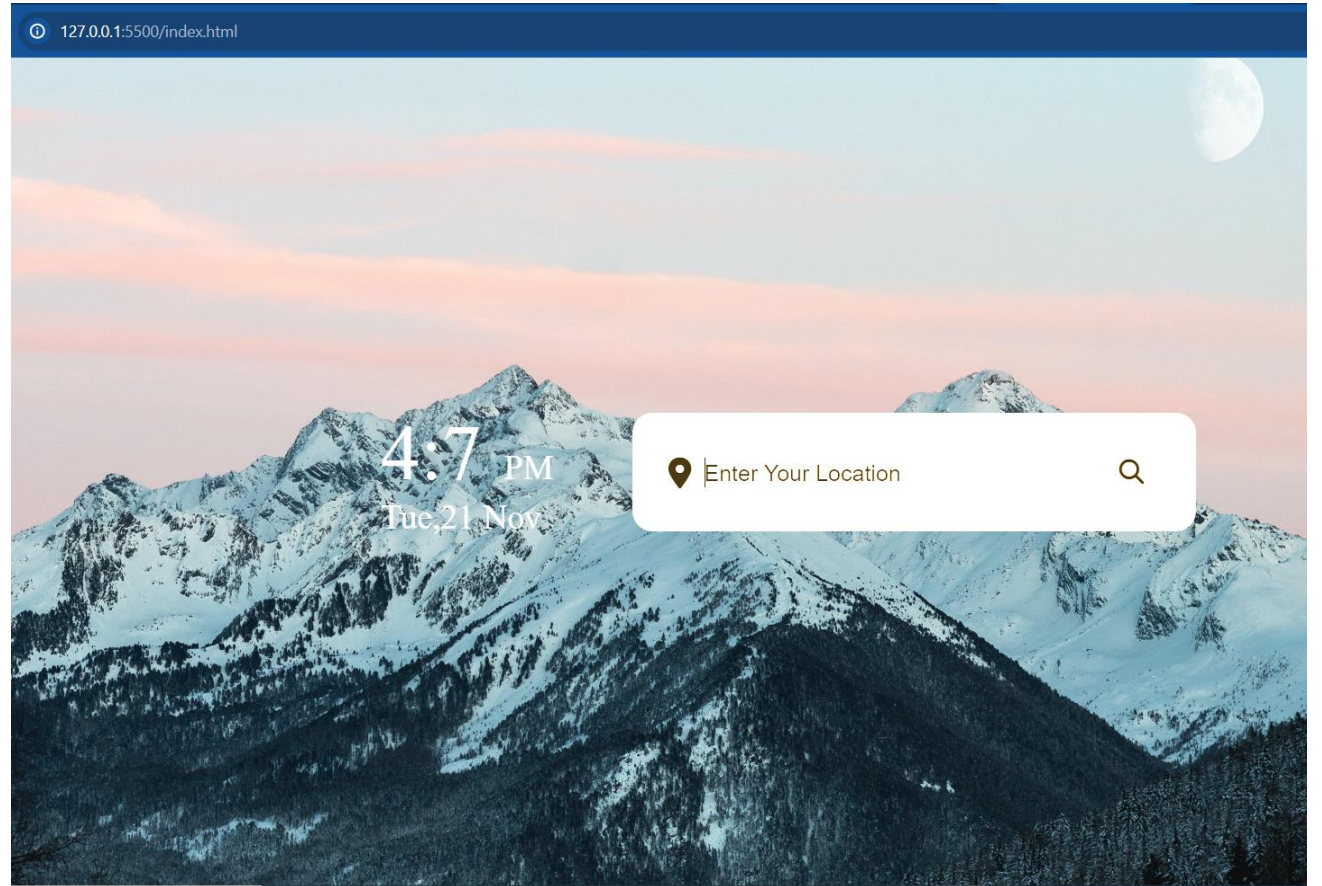
searchButton.addEventListener("click", () => {
  checkWeather(searchInput.value);
  searchInput.value = "";
});
```

- 
- Позволяет запускать код нажатием на ЛКМ или пробел

```
searchButton.addEventListener("click", () => {  
  checkWeather(searchInput.value);  
  searchInput.value = "";  
});  
//позволяет запускать код на enter  
searchInput.addEventListener("keydown", (event) => {  
  if (event.keyCode === 13 || event.keyCode === 32) {  
    checkWeather(searchInput.value);  
    searchInput.value = "";  
  }  
});
```

# Program performance

- Стартовый экран





- Вводим название города(также можно на кириллице) и получаем ответ.

