FINAL PROJECT: WEATHER SITE

WEATHER.IO

Members:

Macros, Daren Daye P.

Del Prado, Louis Joshua H.

Ditan, Tristan I.

Mazo, Keren A.

Objectives:

•**Provide Accurate Current Weather Information:** The primary objective of the weather site is to deliver up-to-date and accurate weather information for today's conditions in various cities around the world.

•**User-Friendly Interface:** Design the website with a user-friendly interface that allows visitors to easily access today's weather for their desired location and navigate the site efficiently.

•**Global Coverage:** Ensure comprehensive coverage of cities worldwide to cater to users from different regions and provide weather updates regardless of their location.

•**Fast and Reliable Search Functionality:** Implement a robust search feature that allows users to quickly find the weather for their desired city, enhancing user experience and satisfaction.

•**Responsive Design:** Create a responsive website that adapts to different devices and screen sizes, enabling users to access today's weather information seamlessly on desktops, tablets, and smartphones.

•**Minimalistic Design:** Keep the design clean and clutter-free, focusing on presenting essential weather information prominently without overwhelming users with unnecessary details or distractions.

•**Accessibility:** Ensure accessibility features are incorporated into the website design to accommodate users with disabilities and provide equal access to weather information for all individuals.

•**Seamless Integration with Weather APIs:** Utilize reliable weather APIs to fetch real-time weather data accurately and seamlessly integrate it into the website, ensuring consistent and reliable updates for users.

About the Website

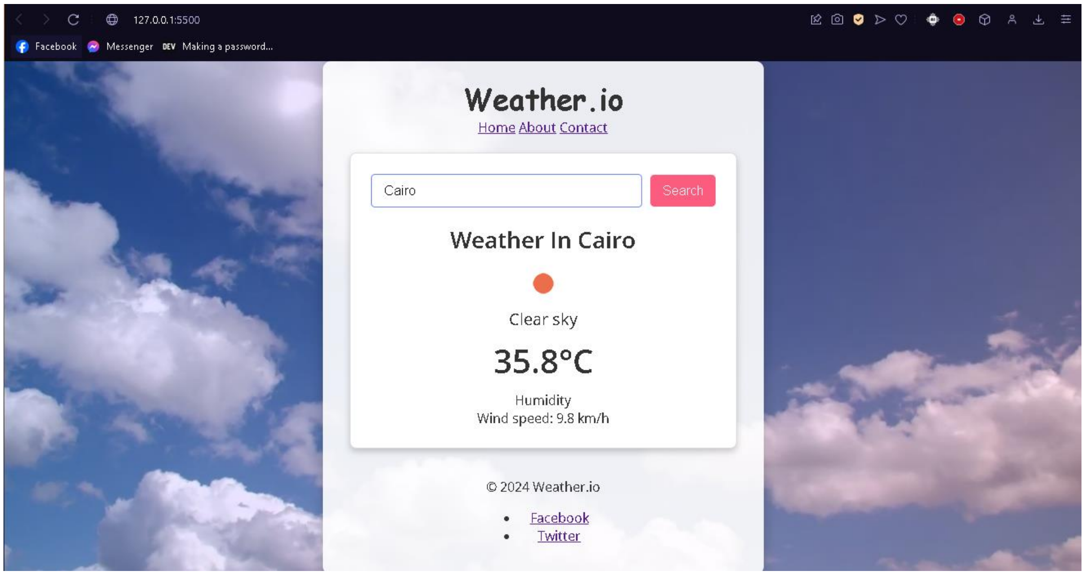
The weather site aims to deliver a seamless experience for users seeking current weather information around the globe. Its primary objective lies in providing accurate and up-to-date weather updates for today's conditions in various cities worldwide. Through a user-friendly interface, visitors can effortlessly navigate the site and access today's weather for their desired locations. With comprehensive coverage spanning cities across the globe, the site ensures that users from different regions can easily find the weather information they need. A robust search functionality enables swift retrieval of weather data, enhancing user satisfaction and experience. The website's responsive design caters to users accessing it from various devices, maintaining consistency and accessibility across desktops, tablets, and smartphones. Emphasizing a minimalistic design approach, the site presents essential weather information. By integrating reliable weather APIs, the site ensures accurate and reliable updates, seamlessly integrating real-time data into its platform. Changing backgrounds based on the weather could provide a dynamic and immersive experience for users, enhancing their engagement and connection with the site. It's a creative way to make the user interface more contextual and responsive to real-world conditions.

Error Handling:

Provide user-friendly messages and error states in the UI when data fetch fails.

Loading Indicator:

Show a loading indicator while fetching data to improve user experience.



Cairo

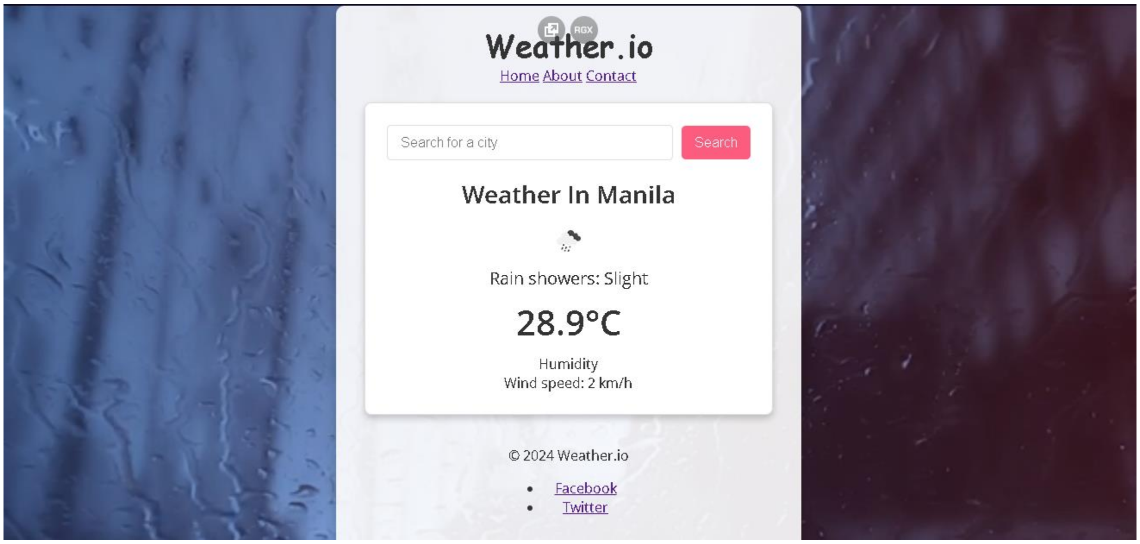
Weather In Cairo

Clear sky

35.8°C

Humidity
Wind speed: 9.8 km/h

© 2024 Weather.io



Weather In Manila

Rain showers: Slight

28.9°C

Humidity
Wind speed: 2 km/h

© 2024 Weather.io

- Facebook
- Twitter

HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Weather.io</title>

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

  <link rel="icon" href="sun-icon-vector.jpg"

</head>



<body>

  <div class="container">  <header>

      <h1>Weather.io</h1>

      <nav>

        <a href="#">Home</a>

        <a href="#">About</a>

        <a href="#">Contact</a>

      </nav>

    </header>



    <main>

      <div class="weather loading">

        <div class="search">

          <input type="text" class="search-bar" placeholder="Search for a city">
```

```
     <button>Search</button>
    </div>
    <div class="city">Weather in Manila</div>
    <div class="icon">
     <img src="" alt="Weather Icon">
    </div>
    <div class="description">Description</div>
    <div class="temp">Temperature</div>
    <div class="humidity">Humidity</div>
    <div class="wind">Wind Speed</div>
   </div>
  </main>

  <footer>
   <p>&copy; 2024 Weather.io</p>
   <ul>
    <li><a href="#">Facebook</a></li>
    <li><a href="#">Twitter</a></li>
   </ul>
  </footer>
 </div>
 <script src="script.js"></script>
</body>
</html>
```

JavaScript (1)

```
let weather = {
  fetchWeather: function (city) {
    this.showLoading();
    fetch(`https://geocoding-api.open-meteo.com/v1/search?name=${city}`)
      .then((response) => {
        if (!response.ok) {
          this.showError("City not found.");
          throw new Error("City not found.");
        }
        return response.json();
      })
      .then((geoData) => {
        if (geoData.results && geoData.results.length > 0) {
          const { latitude, longitude } = geoData.results[0];
          return fetch(
            `https://api.open-meteo.com/v1/forecast?latitude=${latitude}&longitude=${longitude}&current_weather=true`
          );
        } else {
          this.showError("No weather found.");
          throw new Error("No weather found.");
        }
      })
      .then((response) => {
        if (!response.ok) {
          this.showError("No weather found.");
          throw new Error("No weather found.");
        }
        return response.json();
      })
      .then((data) => {
        this.displayWeather(data, city);
      })
      .catch((error) => {
        console.error(error);
      });
  },
  displayWeather: function (data, city) {
    const { temperature, weathercode, windspeed } = data.current_weather;
    const description = this.getWeatherDescription(weathercode);
    const icon = this.getWeatherIcon(weathercode);

    document.querySelector(".city").innerText = "Weather in " + city;
    document.querySelector(".icon img").src = icon;
    document.querySelector(".description").innerText = description;
    document.querySelector(".temp").innerText = temperature + "°C";
    document.querySelector(".wind").innerText = "Wind speed: " + windspeed + " km/h";
    document.querySelector(".weather").classList.remove("loading");

    this.updateBackgroundVideo(weathercode);
```

```
},

getWeatherDescription: function (weathercode) {

  const descriptions = {

    0: "Clear sky",

    1: "Mainly clear",

    2: "Partly cloudy",

    3: "Overcast",

    45: "Fog",

    48: "Depositing rime fog",

    51: "Drizzle: Light",

    53: "Drizzle: Moderate",

    55: "Drizzle: Dense intensity",

    56: "Freezing Drizzle: Light",

    57: "Freezing Drizzle: Dense intensity",

    61: "Rain: Slight",

    63: "Rain: Moderate",

    65: "Rain: Heavy intensity",

    66: "Freezing Rain: Light",

    67: "Freezing Rain: Heavy intensity",

    71: "Snow fall: Slight",

    73: "Snow fall: Moderate",

    75: "Snow fall: Heavy intensity",

    77: "Snow grains",

    80: "Rain showers: Slight",

    81: "Rain showers: Moderate",
    82: "Rain showers: Violent",
    85: "Snow showers: Slight",
    86: "Snow showers: Heavy",
    95: "Thunderstorm: Slight or moderate",
    96: "Thunderstorm with slight hail",
    99: "Thunderstorm with heavy hail",
  };
  return descriptions[weathercode] || "Unknown weather";
},
getWeatherIcon: function (weathercode) {
  const iconMap = {
    0: "01d",
    1: "02d",
    2: "03d",
    3: "04d",
    45: "50d",
    48: "50d",
    51: "09d",
    53: "09d",
    55: "09d",
    56: "13d",
    57: "13d",
    61: "10d",
    63: "10d",
    65: "10d",
  66: "13d",
  67: "13d",
  71: "13d",
  73: "13d",
  75: "13d",
  77: "13d",
  80: "09d",
  81: "09d",
  82: "09d",
  85: "13d",
  86: "13d",
  95: "11d",
  96: "11d",
  99: "11d",
 };
 const iconCode = iconMap[weathercode] || "01d";
 return `http://openweathermap.org/img/wn/${iconCode}.png`;
},
updateBackgroundVideo: function (weathercode) {
 const videoMap = {
  0: "clear_sky.mp4",
  1: "overcast.mp4",
  2: "overcast.mp4",
  3: "overcast.mp4",
  45: "overcast.mp4",
  48: "overcast.mp4",

  51: "drizzle.mp4",

  53: "drizzle.mp4",

  55: "drizzle.mp4",

  56: "freezing_drizzle.mp4",

  57: "freezing_drizzle.mp4",

  61: "rain.mp4",

  63: "rain.mp4",

  65: "rain.mp4",

  66: "rain.mp4",

  67: "rain.mp4",

  71: "snow.mp4",

  73: "snow.mp4",

  75: "heavy_snow.mp4",

  77: "snow.mp4",

  80: "rain.mp4",

  81: "rain.mp4",

  82: "violent_rain_showers.mp4",

  85: "snow.mp4",

  86: "snow.mp4",

  95: "thunderstorm.mp4",

  96: "thunderstorm.mp4",

  99: "thunderstorm.mp4",

};

const backgroundVideo = videoMap[weathercode] || "default_weather.mp4";
```

```
  const videoElement = document.createElement("video");
  videoElement.src = `/bg/${backgroundVideo}`;
  videoElement.autoplay = true;
  videoElement.loop = true;
  videoElement.muted = true;
  videoElement.classList.add("background-video");

  const existingVideo = document.querySelector(".background-video");
  if (existingVideo) {
    existingVideo.remove();
  }

  document.body.appendChild(videoElement);
},
showLoading: function () {
  document.querySelector(".weather").classList.add("loading");
  document.querySelector(".city").innerText = "Loading...";
  document.querySelector(".icon img").src = "";
  document.querySelector(".description").innerText = "";
  document.querySelector(".temp").innerText = "";
  document.querySelector(".wind").innerText = "";
},
showError: function (message) {
  document.querySelector(".weather").classList.add("loading");
  document.querySelector(".city").innerText = message;
```

```
    document.querySelector(".icon img").src = "";

    document.querySelector(".description").innerText = "";

    document.querySelector(".temp").innerText = "";

    document.querySelector(".wind").innerText = "";

  },

  search: function () {

    this.fetchWeather(document.querySelector(".search-bar").value);

  },

};


document.querySelector(".search button").addEventListener("click", function () {

  weather.search();

});


document.querySelector(".search-bar").addEventListener("keyup", function (event) {

  if (event.key == "Enter") {

    weather.search();

  }

});


weather.fetchWeather("Manila");
```

JavaScript (2)

```
const WebSocket = require('ws');
const fetch = require('node-fetch');

const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', (ws) => {
  console.log('Client connected');

  ws.on('message', (message) => {
    const city = message.toString();
    fetchWeather(city, ws);
  });

  ws.on('close', () => {
    console.log('Client disconnected');
  });
});

function fetchWeather(city, ws) {
  fetch(`https://geocoding-api.open-meteo.com/v1/search?name=${city}`)
    .then((response) => response.json())
    .then((geoData) => {
      if (geoData.results && geoData.results.length > 0) {
        const { latitude, longitude } = geoData.results[0];
        return fetch(
          `https://api.open-meteo.com/v1/forecast?latitude=${latitude}&longitude=${longitude}&current_weather=true`
        );
      } else {
        throw new Error('City not found');
      }
    })
    .then((response) => response.json())
    .then((data) => {
      ws.send(JSON.stringify(data));
    })
    .catch((error) => {
      ws.send(JSON.stringify({ error: error.message }));
    });
}

console.log('WebSocket server is running on ws://localhost:8080');
```

CSS

```
@import url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&display=swap');


body {

  font-family: 'Open Sans', sans-serif;

  background: linear-gradient(to right, #6a82fb, #fc5c7d);

  margin: 0;

  padding: 0;

  overflow: hidden;

  color: #333;

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

}


.background-video {

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  object-fit: cover;

  z-index: -1;

  opacity: 0.8;
```

```
}

.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  background: rgba(255, 255, 255, 0.9);
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
  max-width: 500px;
  width: 90%;
  text-align: center;
}

.header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 15px 25px;
  background: linear-gradient(to right, #6a82fb, #fc5c7d);
  color: #fff;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  border-radius: 8px;
  width: 100%;
  margin-bottom: 20px;
}

h1 {
  font-family: 'Lobster', 'Pacifico', cursive;
  font-size: 36px;
  margin: 0;
}

.navigation {
  display: flex;
  list-style: none;
  margin: 0;
  padding: 0;
}

.navigation a {
  padding: 10px 20px;
  text-decoration: none;
  color: #fff;
  font-weight: bold;
  transition: background-color 0.3s, color 0.3s;
}

.navigation a:hover {
  color: #fc5c7d;
  background-color: rgba(255, 255, 255, 0.2);
}

.weather {
  background-color: #ffffff;
  border: 1px solid #ddd;
  border-radius: 8px;
  padding: 25px;
  margin: 20px auto;
  max-width: 600px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  text-align: center;
  animation: fadeIn 1s ease-in-out;
}

.city {
  font-size: 28px;
  font-weight: bold;
  margin-bottom: 10px;
  text-transform: capitalize;
}

.description {
  font-size: 20px;
  margin-bottom: 10px;
}

.temp {
  font-size: 40px;
  font-weight: bold;
  margin-bottom: 10px;
}

.search {
  display: flex;
  align-items: center;
  margin-bottom: 20px;
}

.search-bar {
  padding: 10px 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
  font-size: 16px;
  width: 300px;
  transition: border-color 0.3s;
}

.search-bar:focus {
  outline: none;
  border-color: #6a82fb;
}

.search button {
  border: none;
  background-color: #fc5c7d;
  color: white;
  padding: 10px 15px;
  margin-left: 10px;
  border-radius: 5px;
  cursor: pointer;
  font-size: 16px;
  transition: background-color 0.3s;
}

.search button:hover {
  background-color: #ff8a8a;
}

.weather-icon {
  width: 120px;
  height: 120px;
  margin: 0 auto;
  margin-bottom: 10px;
```

```
}

@keyframes fadeIn {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}

/* Responsive Design */
@media (max-width: 768px) {
  .container {
    padding: 10px;
  }

  .header {
    flex-direction: column;
    text-align: center;
  }

  .navigation {
    flex-direction: column;
  }
  .navigation a {
    padding: 10px;
  }

  .weather {
    width: 90%;
  }

  .search-bar {
    width: 100%;
  }
}
```