WEATHER APP

Introduction

Overview:

- The Weather.io is a web application that provides real-time weather information for a specified location. It utilizes the Open Weather Map API to fetch weather data and displays it in a user-friendly interface. Users can search for a location by city name and receive detailed weather information, including temperature, humidity, wind speed, and weather conditions.

- Weather.io is the application opf science and technology to predict the state of the atmosphere for a given location. Ancient weather application relied on observed patterns of events ,also termed pattern recognition..For example ,it might be observed that if the sunset was particulary red,, the following day often brought fair weather.

Purpose:

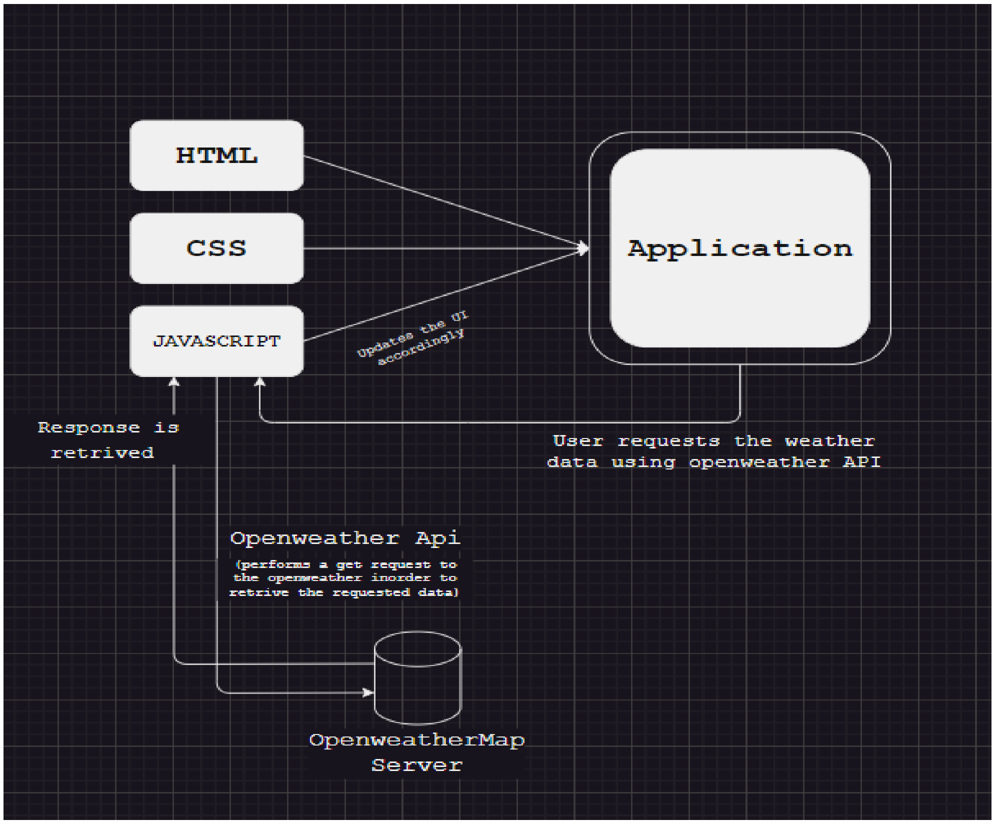
- In this weather app you enter the cities name or countries and its tell weather of the area and we add one function that tells theres Haze or cloud .So let's create this amazing weather app .Weather app used API' to retrive weather data for all cities and countries.
- Weather app is something everybody deals with, and accurate data of it like what is coming can help to make inform decisions.

1 Literature Survey

This specific project is not only the most easy topic but also rarely used. For instance there exist more than 15 Weather Applications and only 3 have managed to climb the cliff Downloads. With the market so low , entering into this was indeed difficult and the greatest risk. As Steve Jobs once said " There is no such thing as a simple project, make it complex " , just what I did here. With every review , every guidance, every step I have reached one step closer to what is now a baby. The project , the Application in itself is enough explain and need no marketing at any point of time. Application stands for itself. It is indeed very important to take care of oneself, no one ever can ever predict rain , nor a storm. Climatic change is subject to risk. In interest of social service and user convenience that this app is perfect of a person who is climate conscious and doesn't use his /her phone well. This app will stand out from others for sure.

2 THEORATICAL ANALYSIS

Block diagram:


HTML
CSS
JAVASCRIPT
Application
Updates the UI accordingly
Response is retrived
User requests the weather data using openweather API
Openweather Api
(performs a get request to the openweather inorder to retrive the requested data)
OpenweatherMap Server

Hardware and software designing

Hardware and software requirements of the project:

Accesing a database:-

- The system should allow administrator to add historical weather data.
- The system should be able to recognize patterns in temperature ,humidity,and wind eith use of historical data .

Software constraints:-

- The development of the system will be constrained by the availability of required software such as webservers,dataset,and development tools.

Hardware Requirements:-

- The system requires a database in order to store persistent data.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta name="viewport" content="width=device-width", initial-scale="1.0">
   <title>Weather App - weatheri</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="card">
      <div class="search">
         <input type="text" placeholder="enter city name" spellcheck="false">
         <button> <img src="search.png"></button>
      </div>
      <div class="weather">
         <img src="rain.png" class="weather-icon">
         <h1 class="temp">22°c</h1>
         <h2 class="city">Visakhapatnam</h2>
         <div class="details">
            <div class="col">
               <img src="humidity.png">
               <div>
                  <p class="humidity">50%</p>
                   <p>Humidity</p>
               </div>
            </div>
            <div class="col">
```

```
                    <img src="wind.png">
                    <div>
                        <p class="Wind">15km/h</p>
                         <p>Wind speed</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <script>
const apiKey ="32a924705a0b7890746985360aa8023f";
const apiurl ="https://api.openweathermap.org/data/2.5/weather?units=metric&q=";
const searchBox = document.querySelector(".search input");
const searchBtn = document.querySelector(".search button");
const weatherIcon = document.querySelector(".weather-icon");
async function checkWeather(city){
    const response = await fetch(apiurl + city + `&appid=${apiKey}`);
    var 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/";
    if(data.weather[0].main == "Clouds"){
        weatherIcon.src = "clouds.png";
    }
    else if(data.weather[0].main == "Clear"){
        weatherIcon.src = "clear.png";
    }
    else if(data.weather[0].main == "Rain"){
        weatherIcon.src = "rain.png";
    }
    else if(data.weather[0].main== "Drizzle"){
        weatherIcon.src = "drizzle.png";
    }
    else if(data.weather[0].main =="Mist"){
        weatherIcon.src = "mist.png";
    }
}
searchBtn.addEventListener("click", ()=>{
    checkWeather(searchBox.value);
})
    </script>
</body>
</html>
```

CSS:

```
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
.container {
    background-color: #4295d0;
    color: #fff;
    padding: 2rem;
    width: 40%;
    margin: 4rem auto;
    border-radius: 10px;
}
.weather__header {
    display: flex;
    justify-content: space-between;
    align-items: center;
}
input {
    border: none;
    background: #83b8de;
    outline: none;
    color: #4d82a8;
    padding: 0.5rem 2.5rem;
    border-radius: 5px;
}
input::placeholder {
    color: #6da798;
}

.weather__search {
    position: relative;
}
.weather__search i {
  position: absolute;
    left: 10px;
    top: 10px;
    font-size: 20px;
    color: #959ea4;
}
.weather__units {
    font-size: 2.0rem;
}
.weather__units span {
    cursor: pointer;
}
.weather__units span:first-child {
    margin-right: 0.5rem;
}
.weather__body {
    text-align: center;
    margin-top: 3rem;
}

.weather__datetime {
    margin-bottom: 2rem;
    font-size: 14px;
}
.weather__forecast {
    background: hsl(215, 61%, 52%);
    display: inline-block;
    padding: 0.5rem 1rem;
    border-radius: 30px;
}
.weather__icon img {
    width: 200px;
}
.weather__temperature {
    font-size: 2.75rem;
}
.weather__minmax {
    display: flex;
    justify-content: center;
}
.weather__minmax p {
    font-size: 14px;
    margin: 0.5rem;
}
.weather__info {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    grid-gap: 1rem;
    margin-top: 3rem;
}
.weather__card {
    display: flex;
    align-items: center;
    background: #8d136611;
    padding: 1rem;
    border-radius: 10px;
}
.weather__card i {
    font-size: 1.5rem;
    margin-right: 1rem;
```

```
}
.weather__card p {
    font-size: 14px;
}
@media(max-width: 936px) {
    .container {
        width: 90%;
    }
    .weather__header {
        flex-direction: column;
    }

    .weather__units {
        margin-top: 1rem;
    }
}
@media(max-width: 400px) {
    .weather__info {
        grid-template-columns: none;
    }
}
```

JAVASCRIPT

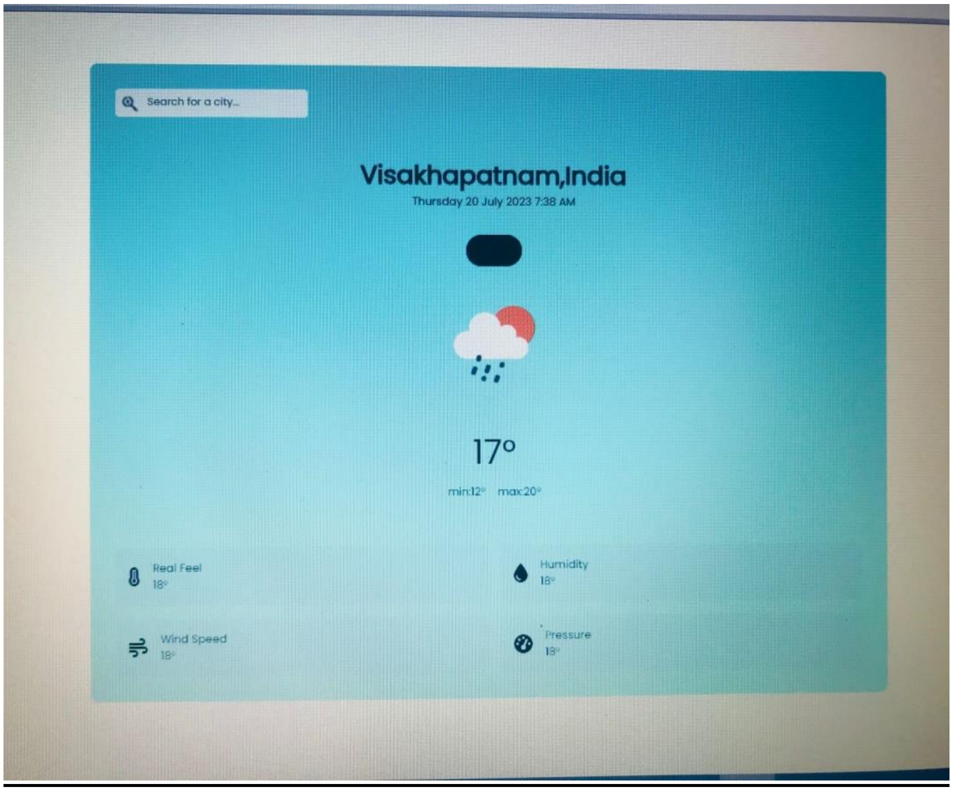
```
const cityElement = document.querySelector(".weather__city");
const datetimeElement = document.querySelector(".weather__datetime");
const forecastElement = document.querySelector(".weather__forecast");
const iconElement = document.querySelector(".weather__icon");
const temperatureElement = document.querySelector(".weather__temperature");
const minMaxElement = document.querySelector(".weather__minmax");
const realFeelElement = document.querySelector(".weather__realfeel");
const humidityElement = document.querySelector(".weather__humidity");
const windElement = document.querySelector(".weather__wind");
const pressureElement = document.querySelector(".weather__pressure");
const searchForm = document.querySelector(".weather__search");
const searchInput = document.querySelector(".weather__searchform");
const locationBtn = document.querySelector(".weather__location-btn");
const celsiusUnit = document.querySelector(".weather_unit_celsius");
const fahrenheitUnit = document.querySelector(".weather_unit_farenheit");

const apiKey = "8e08555847e00b019b8c4c917e2c78df";
const baseUrl = "https://api.openweathermap.org/data/2.5/weather";
let units = "metric";
searchForm.addEventListener("submit", (e) => {
  e.preventDefault();
  const city = searchInput.value.trim();
  if (city !== "") {
    fetchWeatherData(city);
  }
  searchInput.value = "";
});
locationBtn.addEventListener("click", getCurrentLocationWeather);
function getCurrentLocationWeather() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(
      (position) => {
        const latitude = position.coords.latitude;
        const longitude = position.coords.longitude;
        fetchWeatherByCoordinates(latitude, longitude);
      },
      (error) => {
        console.log(error.message);
      }
    );
  } else {
    console.log("Geolocation is not supported by this browser.");
  }
}
async function fetchWeatherByCoordinates(latitude, longitude) {
  try {
    const response = await fetch(
      `${baseUrl}?lat=${latitude}&lon=${longitude}&appid=${apiKey}&units=${units}`
    );
    if (!response.ok) {
      throw new Error("Weather data not available.");
    }
    const data = await response.json();
    updateWeatherInfo(data);
  } catch (error) {
    console.log(error);
  }
}
celsiusUnit.addEventListener("click", () => {
  if (units !== "metric") {
    units = "metric";
    fetchWeatherData(cityElement.textContent); // Update weather data with new unit
  }
});
fahrenheitUnit.addEventListener("click", () => {
  if (units !== "imperial") {
    units = "imperial";
    fetchWeatherData(cityElement.textContent); // Update weather data with new unit
  }
});

async function fetchWeatherData(city) {
  try {
    const response = await fetch(
      `${baseUrl}?q=${city}&appid=${apiKey}&units=${units}`
    );
    if (!response.ok) {
      throw new Error("Weather data not available.");
    }
    const data = await response.json();
    updateWeatherInfo(data);
  } catch (error) {
    console.log(error);
  }
}
function updateWeatherInfo(data) {
  cityElement.textContent = data.name;
  datetimeElement.textContent = getCurrentTime();
  forecastElement.textContent = data.weather[0].description;
  iconElement.innerHTML = `<img src="http://openweathermap.org/img/wn/${data.weather[0].icon}.png"
alt="Weather Icon">`;
  temperatureElement.innerHTML = `${Math.round(data.main.temp)}&#176;${
    units === "metric" ? "C" : "F"
  }`;
  minMaxElement.innerHTML = `<p>Min: ${Math.round(data.main.temp_min)}&#176;${
    units === "metric" ? "C" : "F"
  }</p><p>Max: ${Math.round(data.main.temp_max)}&#176;${
    units === "metric" ? "C" : "F"
  }</p>`;
  realFeelElement.innerHTML = `<p>${Math.round(data.main.feels_like)}&#176;${
    units === "metric" ? "C" : "F"
  }</p>`;
  humidityElement.textContent = `${data.main.humidity}%`;
  windElement.textContent = `${data.wind.speed} ${
    units === "imperial" ? "mph" : "m/s"
  }`;
  pressureElement.textContent = `${data.main.pressure} hPa`;
}
function getCurrentTime() {
  const date = new Date();
  return date.toLocaleString();
}

window.addEventListener("load", () => {
  fetchWeatherData("Hyderabad");
  datetimeElement.textContent = getCurrentTime();
});
```

OUTPUT:



Visakhapatnam,India

Thursday 20 July 2023 7:38 AM

17°

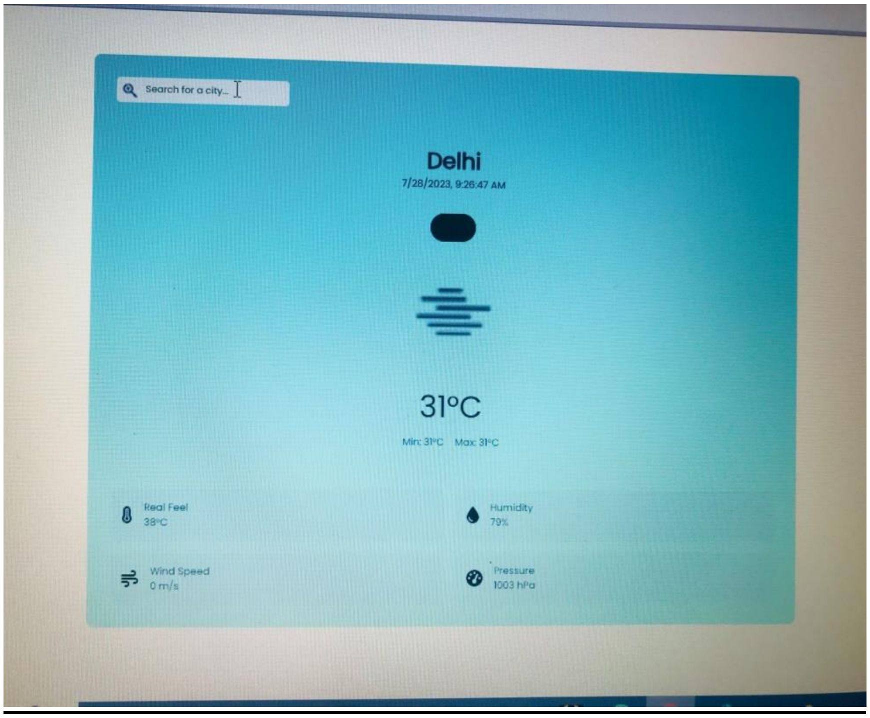
min:12° max:20°

Real Feel
18°

Humidity
18°

Wind Speed
18°

Pressure
18°



Delhi

7/28/2023, 9:26:47 AM

31°C

Min: 31°C Max: 31°C

Real Feel
38°C

Humidity
79%

Wind Speed
0 m/s

Pressure
1003 hPa

5 Advantages and Disadvantages:

- The weather app being able to predict the data to be gathered to build up a more detailed picture of a nations climate, trends within it.
- Weather forecast are not available for many remote or sparsely populated areas, making it difficult for people in these areas to prepare for severe weather.

6 Applications:

- A major part of app is the severe weather alerts and advosiries ,which the national weather services issue in the cast that severe or hazard weather is expected.
- Some of the most commonly known severe weather advisories are the severe thunderstorms about areas that are prone to flood.

7. Conclusion:

- The weather apps are increasingly accurate and useful ,and their benefits extend widely across the economy .While much has been accomplished in improving weather forecasts ,their remains much room for improvements.
- Simultaneously ,they are developing new technologies and observational networks that can enhance forecaster skill and the value of their services to their users.

8 Future scope

- The demand for weather and climate forecast information in support of critical decision-making has grown rapidly during the last decade, and will grow even faster in the coming years. Great advances have been made in the utilization of predictions in many areas of human activities.

- The future of weather applications is promising, with the increasing demand for real-time and accurate weather information. One potential development is the improvement in accuracy through the use of advanced data collection and analysis techniques, as well as sophisticated algorithms.