WEATHER APP

Html code:

WD.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Weather App With JavaScript</title>
    <link rel="stylesheet" href="WD.css" />
  </head>
  <body>
    <div class="container">
      <div class="header">
        <div class="search-box">
          <input
            type="text"
            placeholder="Enter your location"
            class="input-box"
            spellcheck="false"
          />
          <button class="fa-solid fa-magnifying-glass"
id="searchBtn"></button>
        </div>
      </div>
      <div class="location-not-found">
        <h1>Sorry, Location not found!!!</h1>
        <img
src="https://res.cloudinary.com/dcladcqtf/image/upload/v1691168948/404-
error_pdn9nc.png" alt="404 Error" />
      </div>
      <div class="weather-body">
        <img
          src="https://res.cloudinary.com/dcladcqtf/image/upload/v1691169151/c
loudy.jpeg_pjrq5p.jpg"
          alt="Weather Image"
          class="weather-img"
          style="align-items: center; width: 200px; height: 200px"
        />
        <div>
          <h1 class="city" style="text-align: center; color: white">
            City Name
          </h1>
        </div>
        <div class="weather-box">
```

```
        <p class="date" id="date">Tuesday July 25, 2023</p>
        <br />
        <p class="temperature">0 <sup>°C</sup></p>
        <p class="description">light rain</p>
      </div>
      <div class="weather-details">
        <div class="humidity">
          <i class="fa-sharp fa-solid fa-droplet"></i>
          <div class="text">
            <span id="humidity">45%</span>
            <p>Humidity</p>
          </div>
        </div>
        <div class="wind">
          <i class="fa-solid fa-wind"></i>
          <div class="text">
            <span id="wind-speed">12Km/H</span>
            <p>Wind Speed</p>
          </div>
        </div>
      </div>
    </div>

    <div class="table-container">
      <h2>Weather Forecast</h2>
      <table>
        <thead>
          <tr>
            <th>Date</th>
            <th>Temperature (°C)</th>
            <th>Wind Speed (Km/H)</th>
            <th>Humidity (%)</th>
            <th>Description</th>
          </tr>
        </thead>
        <tbody id="forecastTableBody">
          <!-- Rows will be added dynamically here -->
        </tbody>
      </table>
    </div>
  </div>

  <script src="WD.js"></script>
  <script
    src="https://kit.fontawesome.com/595a890311.js"
    crossorigin="anonymous"
  ></script>
</body>
```

```
</html>
```

WD.CSS

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    border: none;
    outline: none;
    font-family: sans-serif;
}

body {
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: rgb(0, 0, 0);
    background-size: 100%;
    background-position: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    justify-content: center;
}

.container {
    opacity: 0.7;
    width: 600px;
    height: min-content;
    background-image: linear-gradient(45deg, rgb(18, 237, 168), rgb(97, 31,
202));
    border-radius: 12px;
    padding: 28px;
}

.search-box {
    width: 100%;
    height: min-content;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.search-box input {
    width: 84%;
    font-size: 20px;
    text-transform: capitalize;
```

```
    color: #000;
    background-color: #e6f5fb;
    padding: 12px 16px;
    border-radius: 14px;
}

.search-box input::placeholder {
    color: #000;
}

.search-box button {
    width: 46px;
    height: 46px;
    background-color: #e6f5fb;
    border-radius: 50%;
    cursor: pointer;
    font-size: 20px;
}

.search-box button:hover {
    color: #fff;
    background-color: #abababab;
}

.weather-body {
    display: flex;
    flex-direction: column;
    align-items: center;
    text-align: center;
    /* Optionally center the text as well */
}

.weather-body img {
    width: 60%;
}

.weather-box {
    margin-block: 20px;
    text-align: center;
}

.weather-box .date {
    font-size: 25px;
    text-align: center;
    font-family: Arial, Helvetica, sans-serif;
    font-style: bold;
}
```

```css
.weather-box .temperature {
    font-size: 40px;
    font-weight: 800;
    position: relative;
}

.weather-box .temperature sup {
    font-size: 20px;
    position: absolute;
    font-weight: 600;
}

.weather-box .description {
    font-size: 20px;
    font-weight: 700;
    text-transform: capitalize;
}

.weather-details {
    width: 100%;
    display: flex;
    justify-content: space-between;
    margin-top: 30px;
}

.humidity,
.wind {
    display: flex;
    align-items: center;
}

.humidity {
    margin-left: 20px;
}

.wind {
    margin-right: 20px;
}

.weather-details i {
    font-size: 36px;
}

.weather-details .text {
    margin-left: 10px;
    font-size: 16px;
}
```

```css
.text span {
    font-size: 20px;
    font-weight: 700;
}

.location-not-found {
    margin-top: 20px;
    display: none;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}

.location-not-found h1 {
    font-size: 20px;
    color: #6b6b6b;
    margin-block-end: 15px;
}

.location-not-found img {
    width: 80%;
}

/* table styling*/

.table-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    text-align: center;
    margin-top: 30px;
}

table {
    width: auto;
    max-width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.1);
    table-layout: auto;
    /* Set table layout to auto */
    animation: fade-in 0.5s ease;
    margin-top: 20px;
    /* Optional margin for spacing */
}
```

```css
th,
td {
    border: 1px solid #ccc;
    padding: 8px;
    text-align: center;
}

th,
td {
    background-color: #000000;
    color: white;
}

tr:nth-child(even) {
    background-color: #f2f2f2;
}

tr:hover {
    background-color: #73ce93;
}

/* Remove border for first row */
tr:first-child {
    border-top: none;
}

/* Remove border for last row */
tr:last-child {
    border-bottom: none;
}

/* Adjust the container width and padding for smaller screens */
@media only screen and (max-width: 768px) {
    .container {
        width: 80%;
        padding: 20px;
    }
}

/* Make the search box more responsive */
@media only screen and (max-width: 768px) {
    .search-box {
        flex-direction: column;
        align-items: center;
        gap: 10px;
    }
```

```
    .search-box input {
        width: 100%;
    }

    .search-box button {
        width: 100px;
        height: 40px;
        font-size: 16px;
    }
}

/* Adjust font sizes and spacing for smaller screens */
@media only screen and (max-width: 576px) {
    .weather-box .date {
        font-size: 20px;
    }

    .weather-box .temperature {
        font-size: 30px;
    }

    .weather-box .description {
        font-size: 18px;
    }

    .weather-details .text {
        font-size: 14px;
    }

    .location-not-found h1 {
        font-size: 18px;
    }
}

/* Make the table more responsive */
@media only screen and (max-width: 768px) {
    table {
        font-size: 14px;
        width: 100%;
        table-layout: fixed;
    }

    th,
    td {
        padding: 6px;
        text-align: left;
        overflow: hidden;
        text-overflow: ellipsis;
```

```
        white-space: nowrap;
    }

    .table-container {
        margin-top: 10px;
        overflow-x: auto;
    }
}

/* Fade-in animation */
@keyframes fade-in {
    0% {
        opacity: 0;
        transform: translateY(10px);
    }

    100% {
        opacity: 1;
        transform: translateY(0);
    }
}
```

WD.JS

```
const inputBox = document.querySelector(".input-box");
const searchBtn = document.getElementById("searchBtn");
const weather_img = document.querySelector(".weather-img");
const date = document.getElementById("date");
let todayDate = new Date();
const temperature = document.querySelector(".temperature");
const description = document.querySelector(".description");
const humidity = document.getElementById("humidity");
const wind_speed = document.getElementById("wind-speed");
const location_not_found = document.querySelector(".location-not-found");
const weather_body = document.querySelector(".weather-body");

const apikey = "0affb46e7c900cceb8e0871dbee5fb16";
const apiUrl = "https://api.openweathermap.org/data/2.5/weather?units=metric";

// Function to get user's current location and fetch weather data
function getCurrentLocationWeather() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(
      async (position) => {
        // Get latitude and longitude
        const latitude = position.coords.latitude;
        const longitude = position.coords.longitude;
```

```
        // Fetch weather data based on user's location
        try {
          const response = await fetch(
            `${apiUrl}&lat=${latitude}&lon=${longitude}&appid=${apikey}`
          );
          const weatherData = await response.json();

          // Update weather information
          updateWeatherInfo(weatherData);
        } catch (error) {
          console.error("Error fetching weather data:", error);
          displayLocationNotFound();
        }
      },
      (error) => {
        console.error("Error getting user's location:", error);
        // If geolocation is denied or not available, use Delhi as default
location
        getDefaultLocationWeather();
      }
    );
  } else {
    // If geolocation is not supported, use Delhi as default location
    getDefaultLocationWeather();
  }
}

// Function to fetch weather data for Delhi (default location)
async function getDefaultLocationWeather() {
  try {
    const response = await fetch(`${apiUrl}&q=Delhi&appid=${apikey}`);
    const weatherData = await response.json();

    // Update weather information
    updateWeatherInfo(weatherData);
  } catch (error) {
    console.error("Error fetching weather data:", error);
    displayLocationNotFound();
  }
}

// Function to handle search button click and fetch weather data for the
searched location
async function handleSearch() {
  const city = inputBox.value.trim(); // Trim the input to remove
leading/trailing spaces
  if (city !== "") {
    checkWeather(city);
```

```
  }
}

// Function to get weather data based on the city name
async function checkWeather(city) {
  try {
    const response = await fetch(`${apiUrl}&q=${city}&appid=${apikey}`);
    const weatherData = await response.json();

    // Update weather information
    updateWeatherInfo(weatherData);
  } catch (error) {
    console.error("Error fetching weather data:", error);
    displayLocationNotFound();
  }
}

// Function to update weather information on the page
function updateWeatherInfo(weatherData) {
  // Update weather information on the page as before
  console.log(weatherData);

  name = weatherData.name;
  updateTable(name);

  date.innerText = dateManage(new Date());
  temperature.innerHTML = `${Math.round(weatherData.main.temp)}°C`;
  description.innerHTML = weatherData.weather[0].description;
  humidity.innerHTML = `${weatherData.main.humidity}%`;
  wind_speed.innerHTML = `${weatherData.wind.speed}Km/H`;

  // Update weather image
  switch (weatherData.weather[0].main) {
    case "Clouds":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169151/cloudy.jpeg_pjrq5p.jpg";
      break;
    case "Clear":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169205/clear-sky.jpeg_v0sk6s.jpg";
      break;
    case "Rain":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169619/rain.jpeg_gwhpsm.jpg";
      break;
```

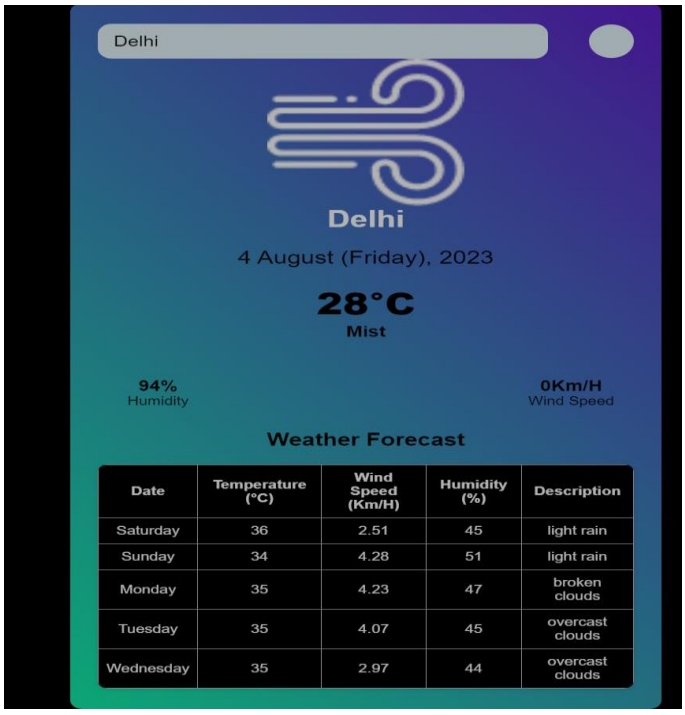
```
    case "Mist":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691077732/wind_qkwsx8.png
";
      break;
    case "Snow":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169577/snowy.jpeg_ucir
2z.jpg";
      break;
    case "Haze":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169406/haze.jpeg_hilu6
q.jpg";
      break;
    case "Thunderstorm":
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169545/thunder.jpeg_z6
to7j.jpg";
      break;
    default:
      weather_img.src =
"https://res.cloudinary.com/dcladcqtf/image/upload/v1691169465/default.jpeg_y1
2eey.jpg";
  }

  // Update city name on the page
  document.querySelector(".city").innerHTML = weatherData.name;

  updateTable(name);

  // Hide the location not found message
  location_not_found.style.display = "none";
  weather_body.style.display = "flex";
}

let apiurl2 = "https://api.openweathermap.org/data/2.5/forecast?units=metric";

// Function to get weather data based on coordinate
async function updateTable(city) {
  try {
    const response = await fetch(`${apiurl2}&q=${city}&appid=${apikey}`);
    const forecastData = await response.json();
    console.log(forecastData);

    // Update weather information
    console.log("updateTable");
    console.log(forecastData);
```

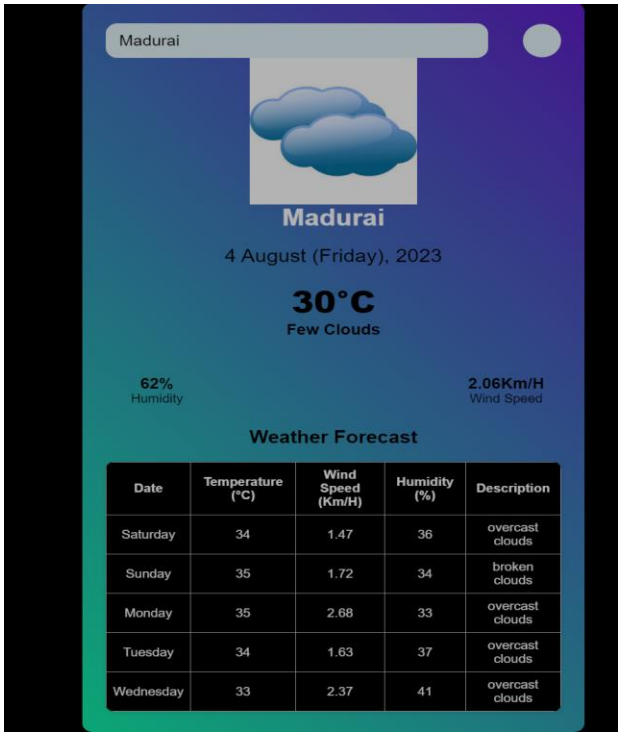
```
    updateWeatherTable(forecastData);
  } catch (error) {
    console.error("Error fetching weather data:", error);
    displayLocationNotFound();
  }
}

// updateTable();

function updateWeatherTable(forecast) {
  console.log(" update weatherupdate Table");
  console.log(forecast);
  const tableBody = document.getElementById("forecastTableBody");

  // Clear existing table rows
  tableBody.innerHTML = "";

  // Days of the week
  const daysOfWeek = [
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday",
  ];

  // Loop through forecastData for the first 8 days
  for (let i = 4; i < forecast.list.length; ) {
    const data = forecast.list[i];

    const row = document.createElement("tr");

    const dateCell = document.createElement("td");
    // const datePart = data.dt_txt.split(" ")[0]; // Extract the date part
    const timestamp = new Date(data.dt_txt);
    const dayOfWeek = daysOfWeek[timestamp.getDay()]; // Get the day of the
week
    dateCell.textContent = dayOfWeek; // Update with the actual property from
your data
    row.appendChild(dateCell);

    // Math.round(data.main.temp)
    const tempCell = document.createElement("td");
    tempCell.textContent = Math.round(data.main.temp); // Update with the
actual property from your data
    row.appendChild(tempCell);
```

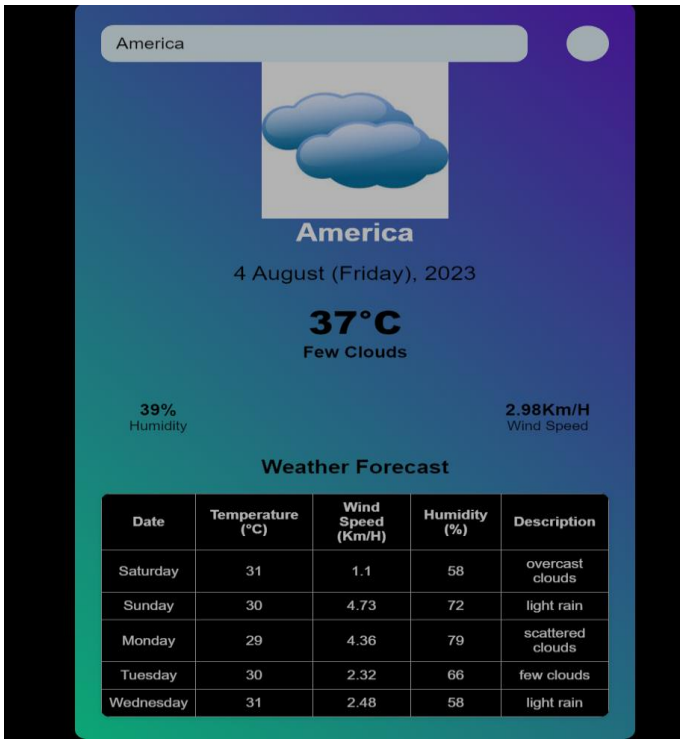
```
    const windCell = document.createElement("td");
    windCell.textContent = data.wind.speed; // Update with the actual property
from your data
    row.appendChild(windCell);

    const humidityCell = document.createElement("td");
    humidityCell.textContent = data.main.humidity; // Update with the actual
property from your data
    row.appendChild(humidityCell);

    const descCell = document.createElement("td");
    descCell.textContent = data.weather[0].description; // Update with the
actual property from your data
    row.appendChild(descCell);

    // Append the row to the table body
    tableBody.appendChild(row);
    i = i + 8;
  }
}

// Function to display location not found message
function displayLocationNotFound() {
  location_not_found.style.display = "flex";
  weather_body.style.display = "none";
}

// Call the function to get user's current location weather on page load
document.addEventListener("DOMContentLoaded", () => {
  getCurrentLocationWeather();
});

// Call the function to handle search button click
searchBtn.addEventListener("click", handleSearch);

function dateManage(dateArg) {
  let days = [
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday",
  ];

  let months = [
```

```
    "January",
    "February",
    "March",
    "April",
    "May",
    "June",
    "July",
    "August",
    "September",
    "October",
    "November",
    "December",
  ];

  //   let year = dateArg.getFULLYear();
  let year = dateArg.getFullYear(); // Corrected method name
  let month = months[dateArg.getMonth()];
  let date = dateArg.getDate();
  let day = days[dateArg.getDay()];

  return `${date} ${month} (${day}), ${year}`;
}
```

OUTPUT SCREENSHOT:




Madurai
Madurai
4 August (Friday), 2023
30°C
Few Clouds
62%
Humidity
2.06Km/H
Wind Speed
Weather Forecast
Date
Temperature (°C)
Wind Speed (Km/H)
Humidity (%)
Description
Saturday
34
1.47
36
overcast clouds
Sunday
35
1.72
34
broken clouds
Monday
35
2.68
33
overcast clouds
Tuesday
34
1.63
37
overcast clouds
Wednesday
33
2.37
41
overcast clouds


America
America
4 August (Friday), 2023
37°C
Few Clouds
39%
Humidity
2.98Km/H
Wind Speed
Weather Forecast
Date
Temperature (°C)
Wind Speed (Km/H)
Humidity (%)
Description
Saturday
31
1.1
58
overcast clouds
Sunday
30
4.73
72
light rain
Monday
29
4.36
79
scattered clouds
Tuesday
30
2.32
66
few clouds
Wednesday
31
2.48
58
light rain