WD-MAJOR-PROJECT

NAME: *Adhada shiva kavya*

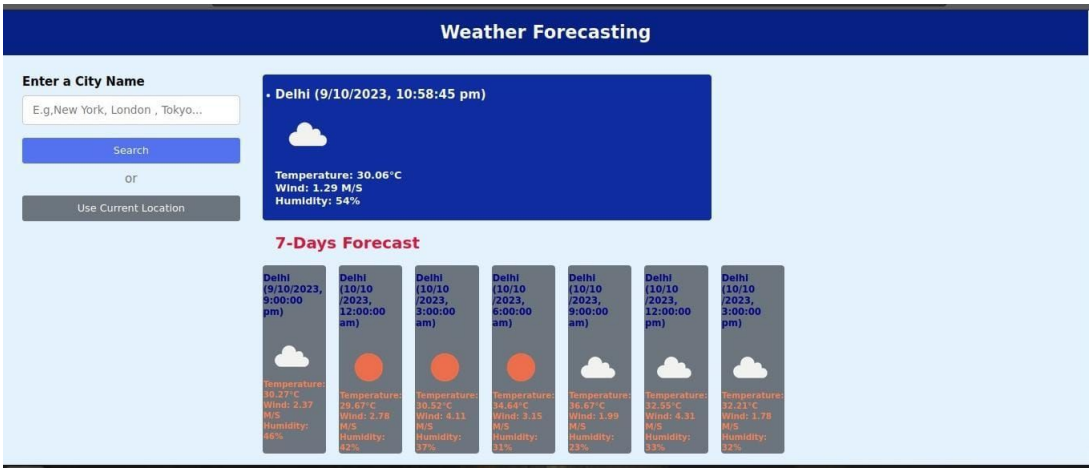
DOMAIN NAME: *Web Development*

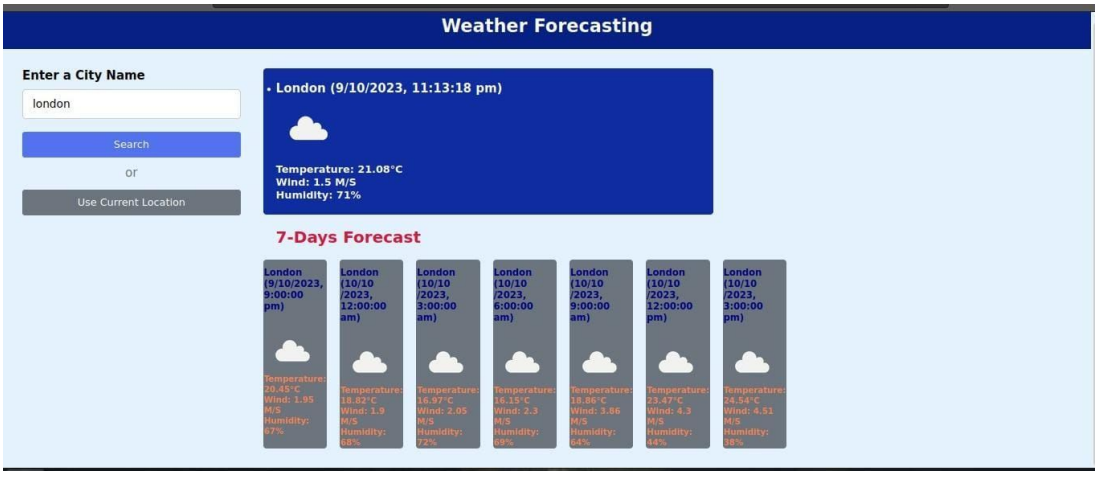
Batch: *August 2023*

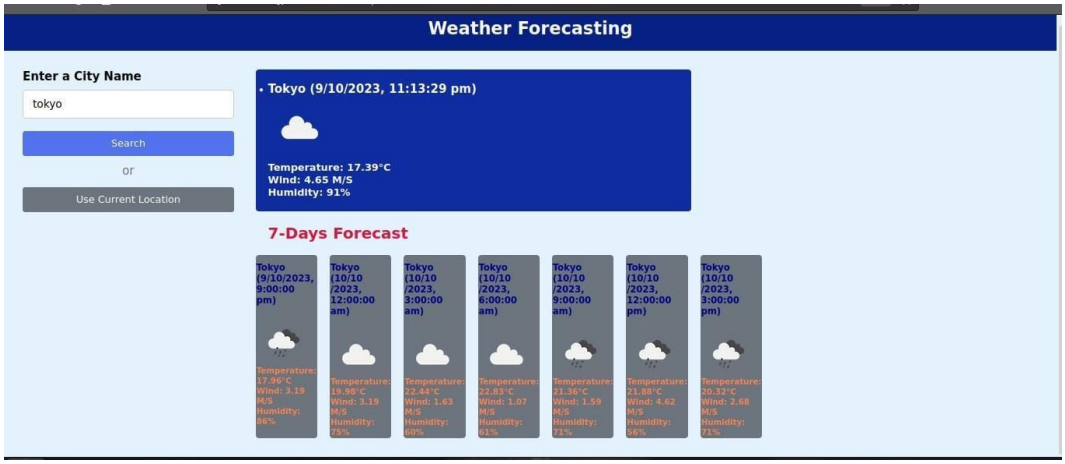
Project Team: *WD-08-SP09*

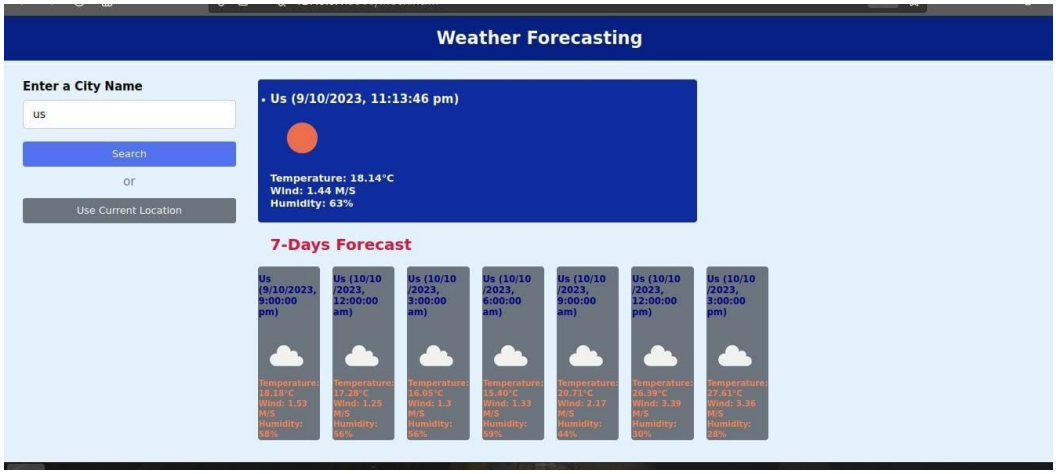
Project Name: *Weather Forecast web App*

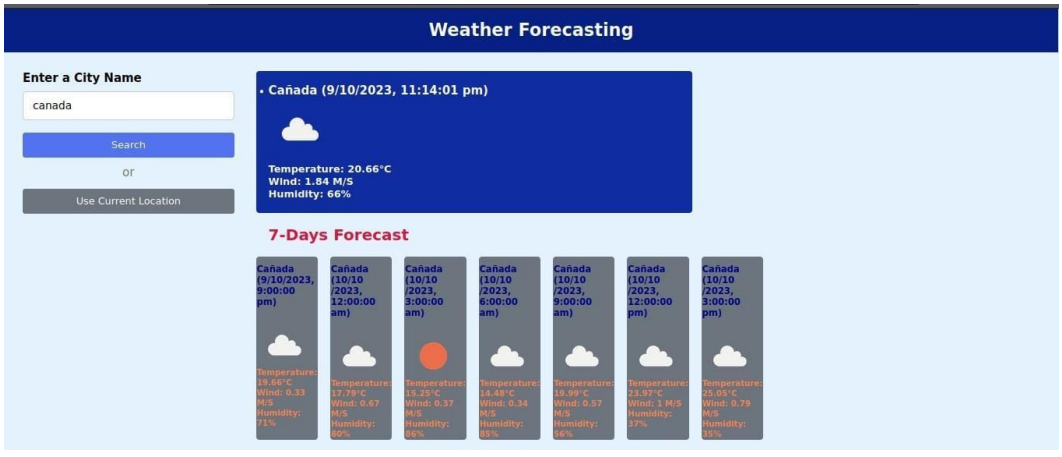
OUTPUTS






Weather Forecasting
Enter a City Name
tokyo
Search
or
Use Current Location
• Tokyo (9/10/2023, 11:13:29 pm)
Temperature: 17.39°C
Wind: 4.65 M/S
Humidity: 91%
7-Days Forecast
Tokyo (9/10/2023, 9:00:00 pm)
Temperature: 17.96°C
Wind: 3.19 M/S
Humidity: 86%
Tokyo (10/10 /2023, 12:00:00 am)
Temperature: 19.98°C
Wind: 3.19 M/S
Humidity: 75%
Tokyo (10/10 /2023, 3:00:00 am)
Temperature: 22.44°C
Wind: 1.63 M/S
Humidity: 60%
Tokyo (10/10 /2023, 6:00:00 am)
Temperature: 22.83°C
Wind: 1.07 M/S
Humidity: 61%
Tokyo (10/10 /2023, 9:00:00 am)
Temperature: 21.36°C
Wind: 1.59 M/S
Humidity: 71%
Tokyo (10/10 /2023, 12:00:00 pm)
Temperature: 21.88°C
Wind: 4.62 M/S
Humidity: 56%
Tokyo (10/10 /2023, 3:00:00 pm)
Temperature: 20.32°C
Wind: 2.68 M/S
Humidity: 71%


Weather Forecasting
Enter a City Name
us
Search
or
Use Current Location
• Us (9/10/2023, 11:13:46 pm)
Temperature: 18.14°C
Wind: 1.44 M/S
Humidity: 63%
7-Days Forecast
Us (9/10/2023, 9:00:00 pm)
Temperature: 18.18°C
Wind: 1.53 M/S
Humidity: 58%
Us (10/10 /2023, 12:00:00 am)
Temperature: 17.28°C
Wind: 1.25 M/S
Humidity: 56%
Us (10/10 /2023, 3:00:00 am)
Temperature: 16.05°C
Wind: 1.3 M/S
Humidity: 56%
Us (10/10 /2023, 6:00:00 am)
Temperature: 15.40°C
Wind: 1.33 M/S
Humidity: 59%
Us (10/10 /2023, 9:00:00 am)
Temperature: 20.71°C
Wind: 2.17 M/S
Humidity: 44%
Us (10/10 /2023, 12:00:00 pm)
Temperature: 26.39°C
Wind: 3.39 M/S
Humidity: 30%
Us (10/10 /2023, 3:00:00 pm)
Temperature: 27.61°C
Wind: 3.36 M/S
Humidity: 28%


Weather Forecasting
Enter a City Name
canada
Search
or
Use Current Location
• Cañada (9/10/2023, 11:14:01 pm)
Temperature: 20.66°C
Wind: 1.84 M/S
Humidity: 66%
7-Days Forecast
Cañada (9/10/2023, 9:00:00 pm)
Temperature: 19.66°C
Wind: 0.33 M/S
Humidity: 71%
Cañada (10/10 /2023, 12:00:00 am)
Temperature: 17.79°C
Wind: 0.67 M/S
Humidity: 80%
Cañada (10/10 /2023, 3:00:00 am)
Temperature: 15.25°C
Wind: 0.37 M/S
Humidity: 86%
Cañada (10/10 /2023, 6:00:00 am)
Temperature: 14.48°C
Wind: 0.34 M/S
Humidity: 85%
Cañada (10/10 /2023, 9:00:00 am)
Temperature: 19.99°C
Wind: 0.57 M/S
Humidity: 56%
Cañada (10/10 /2023, 12:00:00 pm)
Temperature: 23.97°C
Wind: 1 M/S
Humidity: 37%
Cañada (10/10 /2023, 3:00:00 pm)
Temperature: 25.05°C
Wind: 0.79 M/S
Humidity: 35%

HTMLCODE(index.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>Document</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js" defer></script>
</head>
<body>
    <h1>Weather Forecasting</h1>
    <div class="container">
        <div class="weather-input">
            <h3>Enter a City Name</h3>
            <input class="city-input" type="text" placeholder="E.g,New York, London , Tokyo...">
            <button class="search-btn">Search</button>
            <div class="separator"></div>
            <button class="location-btn">Use Current Location</button>
        </div>

        <!-- Example HTML for displaying weather icon -->

            <div class="1">
                <div class="current-weather">
                    <div class="deatails">
                        <h2>_____(_______)</h2>
                        <h4>Temperature :__&deg;C</h4>
                        <h4>Wind:___M/S</h4>
                        <h4>Humidity:___%</h4>
                    </div>
                </div>
                <div class="days-forecast">
                    <h2>7-Days Forecast</h2>
                    <ul class="weather-cards">
                        <li class="card">
                            <h3>(_____)</h3>
                            <h4>Temperature :__&deg;C</h4>
                            <h4>Wind:____M/S</h4>
                            <h4>Humidity:___%</h4>
                        </li>

                        <li class="card">
                                    <h3>(_______)</h3>
                                    <h4>Temperature:_____&deg;C</h4>
                                    <h4>Wind:___M/S</h4>
                                    <h4>Humidity:____%</h4>
                                </li>
                                <li class="card">
                                    <h3>(_______)</h3>
                                    <h4>Temperature :____&deg;C</h4>
                                    <h4>Wind:____M/S</h4>
                                    <h4>Humidity:__%</h4>
                                </li>
```

```
<li class="card">
                    <h3>(_______)</h3>
                    <h4>Temperature :____&deg;C</h4>
                    <h4>Wind:_____M/S</h4>
                    <h4>Humidity:____%</h4>
                </li>
                <li class="card">
                    <h3>(_______)</h3>
                    <h4>Temperature :_____&deg;C</h4>
                    <h4>Wind:_____M/S</h4>
                    <h4>Humidity:____%</h4>
                </li>
                <li class="card">
                    <h3>(__________)</h3>
                    <h4>Temperature :____&deg;C</h4>
                    <h4>Wind:___M/S</h4>
                    <h4>Humidity:___%</h4>
                </li>
                <li class="card">
                    <h3>(______)</h3>
                    <h4>Temperature :___&deg;C</h4>
                    <h4>Wind:___M/S</h4>
                    <h4>Humidity:___%</h4>
                </li>
            </ul>
        </div>
    </div>
  </div>
</body>
</html>
```

CSS CODE(style.css)

```
*{
    margin:0px;
    padding:0px;
    box-sizing:border-box;
    font-family:'open Sans',sans-serif;
}
body{
    background-color: #E3F2FD;
}
h1{
color:#fff;
padding:18px 0;
text-align:center;
font-size:1.75rem;
background:#062085;
}

.container{
    display:flex;
    gap:35px;
    padding:30px;
}
.weather-input{
    width:550 px;
}
```

```
.weather-input input{
    width:100%;
    height:46px;
    outline:none;
    padding:0 15px;
    margin:10px 0 20px 0;
    font-size:1.07rem;
    border-radius:4px;
    border:1px solid #ccc;
}
.weather-input .separator{
    height:1px;
    width:100%;
    margin:25px 0;
    background:#bbb;
    display:flex;
    align-items:center;
    justify-content: center;
}
.weather-input .separator::before{
    content:"or";
    color:#6C7570;
    font-size:1.18rem;
    padding:0 15px;
    margin-top:-4px;
    background:#E3F2FD;
}
.weather-input button{
    width:100%;
    padding:10px 0;
    cursor:pointer;
    outline:none;
    border:none;
    color:#fff;
    border-radius:4px;
    background-color: #5372F0;
    font-size:1rem;
}
  .weather-input .location-btn{
      background-color: #6C757D;
  }
  .weather-data{
      width:100%;
  }
  .current-weather{
      color:#fff;
      display:flex;
      border-radius:5px;
      padding:20px 70px 20px 20px;
      background-color: #0f2ca1;
  }
  .current-weather h2{
      font-size:1.7rem;
  }
  .weather-data  h4{
      margin-top:12px;
      font-size:1rem;
      font-weight:500;
  }
```

```
/* Example CSS for styling weather icons */

.current-weather .icon{
    text-align:center;
}
.current-weather .icon img{
    max-width:120px;
    margin-top:-15px;
}
.current-weather .icon h4{
    margin-top:-10px;
    text-transform: capitalize;
}
.days-forecast h2{
    color:crimson;
    font-size: 25px;
    align-items:center;
    margin:20px;
}
.days-forecast h3{
    font-size:0.90rem;
    margin:10px 0px;
    color:darkblue;
}
.days-forecast li{
    font-size:0.8rem;
    color:white;
    color:coral;
}
.weather-cards{
    display:flex;
    gap:20px;
    width:700px;
}
.weather-cards .card{
    width:170px;
    list-style:none;
    padding:1px;
    border-radius:5px;
    background-color: #6C757D;
    width:calc (100% / 7);
}
.weather-cards .card img{
    max-width:90px;
    margin:5px 0 -12px 0;
}
```

javascript code(script.js)

```
const cityInput = document.querySelector(".city-input");
const searchButton = document.querySelector(".search-btn");
const locationButton = document.querySelector(".location-btn");
const currentWeatherDiv = document.querySelector(".current-weather");
const weatherCardsDiv = document.querySelector(".weather-cards");


 const API_KEY = "54ddb18954f4074d8959e1c3aee04978";
```

```
const weatherIcons = {
  Clear: "01d", // Sunny
  Clouds: "03d", // Cloudy
  Rain: "09d", // Rainy
  Thunderstorm: "11d", // Thunderstorm
  Snow: "13d", // Snowy
  Mist: "50d", // Misty
};

// Function to fetch and display weather data for Delhi on page load
const fetchDefaultCityWeather = () => {
  const defaultCityName = "Delhi";
  const GEOCODING_API_URL = `https://api.openweathermap.org/geo/1.0/direct?q=${defaultCi

  fetch(GEOCODING_API_URL)
    .then((res) => res.json())
    .then((data) => {
      if (!data.length) {
        alert(`No coordinates found for ${defaultCityName}`);
        return;
      }
      const { name, lat, lon } = data[0];
      getWeatherDetails(name, lat, lon, true);
    })
    .catch(() => {
      alert("An error occurred while fetching the coordinates!");
    });
};
// Call the fetchDefaultCityWeather function when the page loads
window.addEventListener("load", fetchDefaultCityWeather);

const createWeatherCard = (cityName, weatherItem, isCurrentWeather = false, isDefaultCity = fa
  const datetime = isCurrentWeather
    ? new Date().toLocaleString()
    : new Date(weatherItem.dt_txt).toLocaleString();

  const weatherDescription = weatherItem.weather[0].main;

  const weatherIcon = weatherIcons[weatherDescription] || "01d"; // Default to sunny icon

  // Customize the card's appearance for the default city

  const cardClassName = isDefaultCity ? "card default-city" : "card";

   return `
     <li class="${cardClassName}">
       <h3>${cityName} (${datetime})</h3>
       <img src="https://openweathermap.org/img/wn/${weatherIcon}@2x.png" alt="weather-icon">
       <h4>Temperature: ${(weatherItem.main.temp - 273.15).toFixed(2)}&deg;C</h4>
       <h4>Wind: ${weatherItem.wind.speed} M/S</h4>
       <h4>Humidity: ${weatherItem.main.humidity}%</h4>
     </li>`;
 };
const displayCurrentWeather = (cityName, currentWeather, isDefaultCity = false) => {
  currentWeatherDiv.innerHTML = createWeatherCard(
    cityName,
    currentWeather,
    true,
    isDefaultCity
  );
};
```

```
const getWeatherDetails = (cityName, lat, lon, isDefaultCity = false) => {
  const WEATHER_API_URL = `https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=$

  fetch(WEATHER_API_URL)
    .then((res) => res.json())
    .then((data) => {
      // Clear previous weather data
      currentWeatherDiv.innerHTML = "";
      weatherCardsDiv.innerHTML = "";

      const currentWeather = data.list[0];
      displayCurrentWeather(cityName, currentWeather, isDefaultCity);

      data.list.slice(1).forEach((weatherItem) => {
        weatherCardsDiv.insertAdjacentHTML(
          "beforeend",
          createWeatherCard(cityName, weatherItem, false, isDefaultCity)
        );
      });
    })
    .catch(() => {
      alert("An error occurred while fetching the weather forecast!");
    });
};

const getCityCoordinates = () => {
  const cityName = cityInput.value.trim();
  if (!cityName) return;
  const GEOCODING_API_URL = `https://api.openweathermap.org/geo/1.0/direct?q=${cityName}&

  fetch(GEOCODING_API_URL)
    .then((res) => res.json())
    .then((data) => {
      if (!data.length) return alert(`No coordinates found for ${cityName}`);
      const { name, lat, lon } = data[0];
      getWeatherDetails(name, lat, lon);
    })
    .catch(() => {
      alert("An error occurred while fetching the coordinates!");
    });
};

const getUserCoordinates = () => {
  navigator.geolocation.getCurrentPosition(
    (position) => {
      const { latitude, longitude } = position.coords;
      const REVERSE_GEOCODING_URL = `https://api.openweathermap.org/geo/1.0/reverse?lat=$

      fetch(REVERSE_GEOCODING_URL)
        .then((res) => res.json())
        .then((data) => {
          const { name, lat, lon } = data[0];
          getWeatherDetails(name, lat, lon);
        })
        .catch(() => {
          alert("An error occurred while fetching the city!");
        });
    },
    (error) => {
      if (error.code === error.PERMISSION_DENIED) {
        alert("Geolocation request denied.")
}
  );
```