WD-MAJOR PROJECT

Name : Arshad Ali

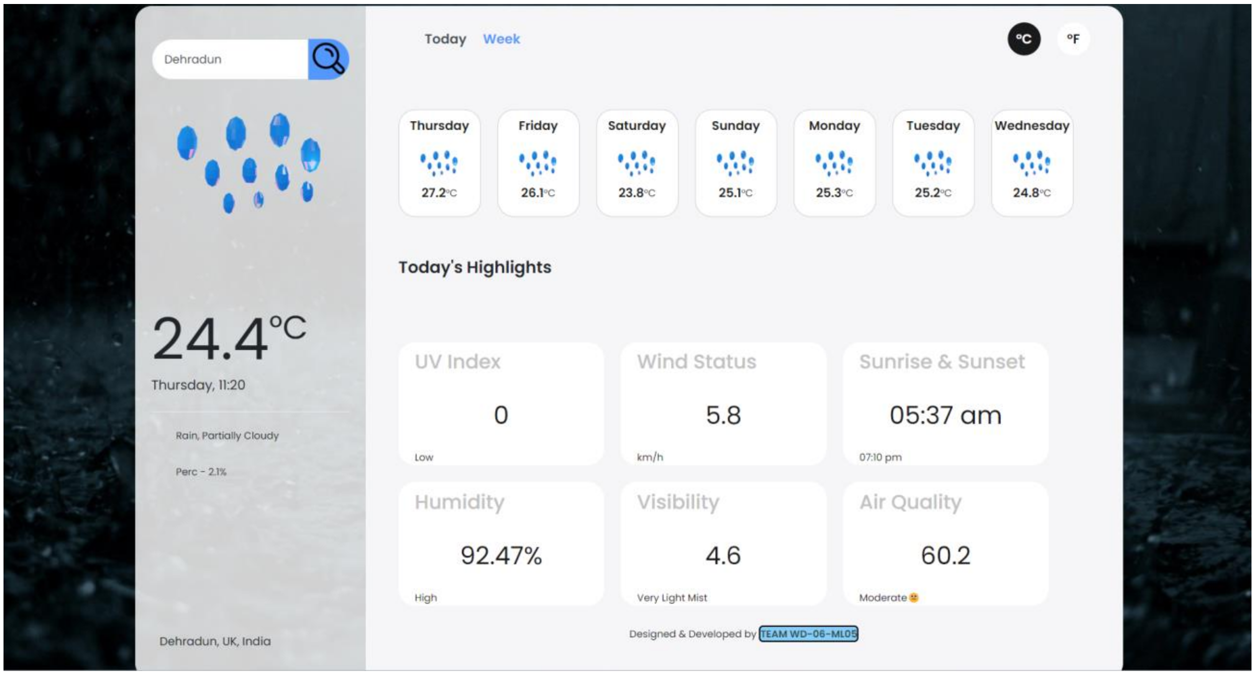
Domain : Web Development

Batch : June 2023

Website Name : **https://aak-2211.github.io/MAJOR-WD-06-ML05/**

Topic : Weather Forecast Web App

OUTPUTS:



Dehradun

24.4°C

Thursday, 11:20

Rain, Partially Cloudy

Perc - 2.1%

Dehradun, UK, India

Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday
27.2°C	26.1°C	23.8°C	25.1°C	25.3°C	25.2°C	24.8°C

Today's Highlights

UV Index 0 Low

Wind Status 5.8 km/h

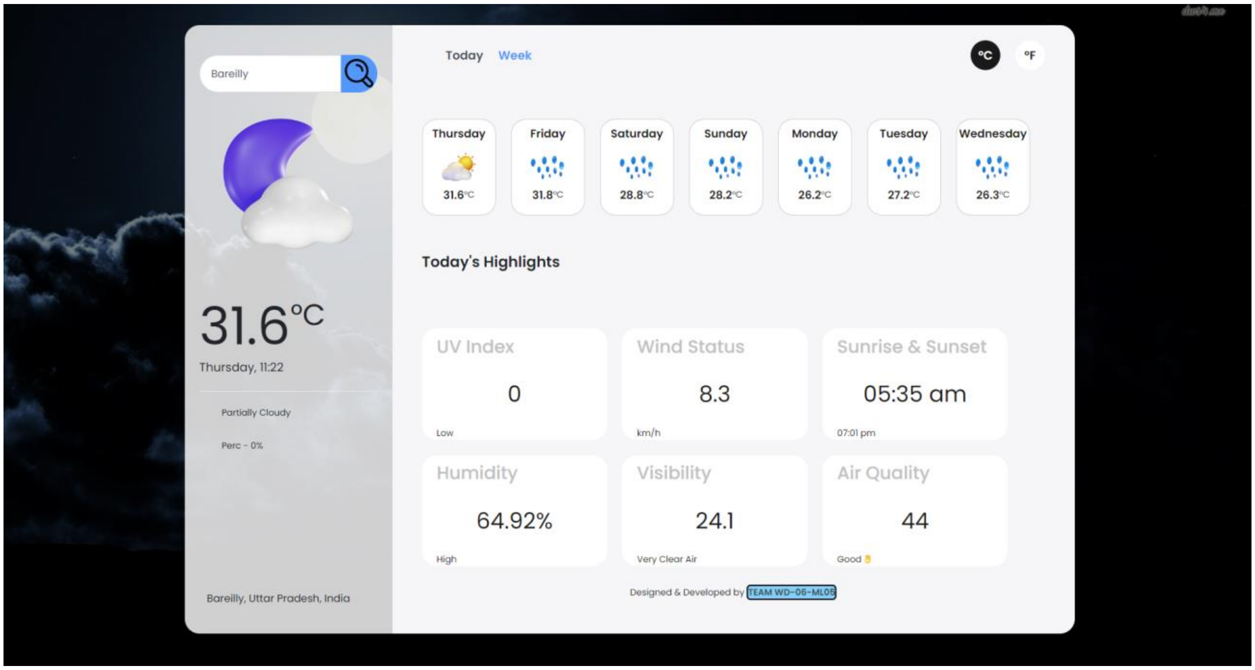
Sunrise & Sunset 05:37 am 07:10 pm

Humidity 92.47% High

Visibility 4.6 Very Light Mist

Air Quality 60.2 Moderate😐

Designed & Developed by TEAM WD-06-ML05



Bareilly

31.6°C

Thursday, 11:22

Partially Cloudy

Perc - 0%

Bareilly, Uttar Pradesh, India

Today Week

°C °F

Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday
31.6°C	31.8°C	28.8°C	28.2°C	26.2°C	27.2°C	26.3°C

Today's Highlights

UV Index
0
Low

Wind Status
8.3
km/h

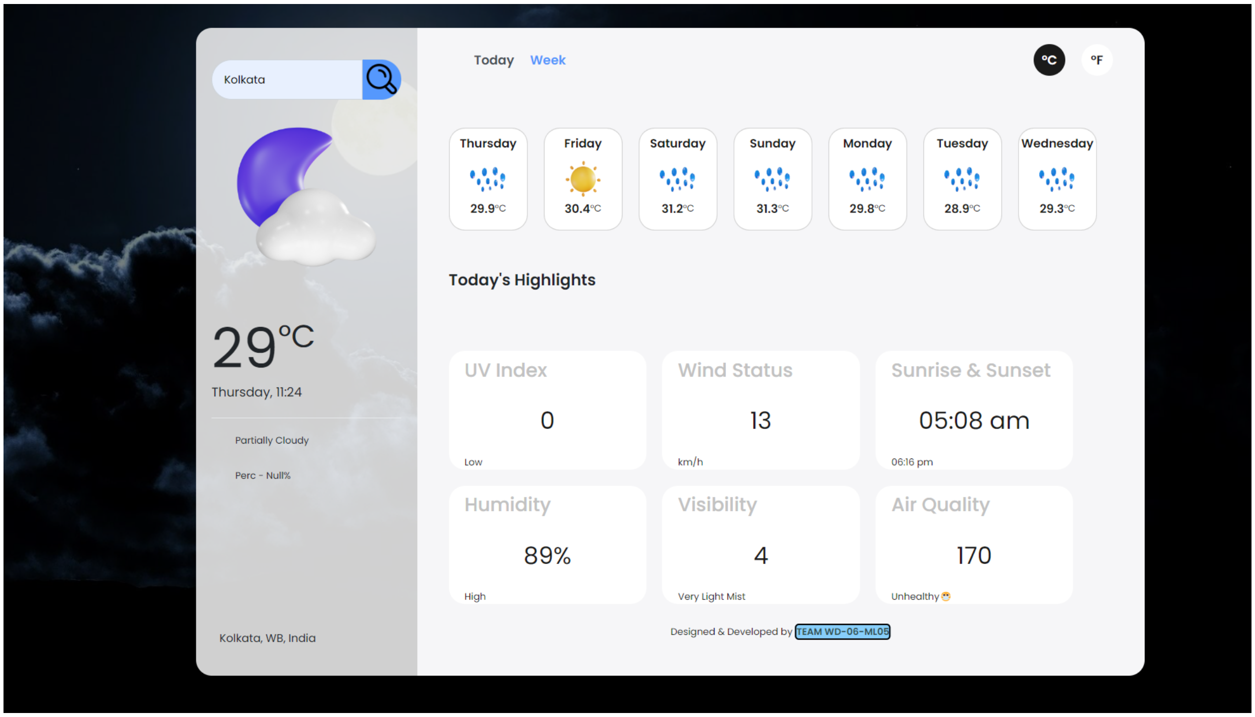
Sunrise & Sunset
05:35 am
07:01 pm

Humidity
64.92%
High

Visibility
24.1
Very Clear Air

Air Quality
44
Good

Designed & Developed by TEAM WD-06-ML05



Kolkata

29°C

Thursday, 11:24

Partially Cloudy

Perc - Null%

Kolkata, WB, India

Today Week

°C °F

Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday
29.9°C	30.4°C	31.2°C	31.3°C	29.8°C	28.9°C	29.3°C

Today's Highlights

UV Index
0
Low

Wind Status
13
km/h

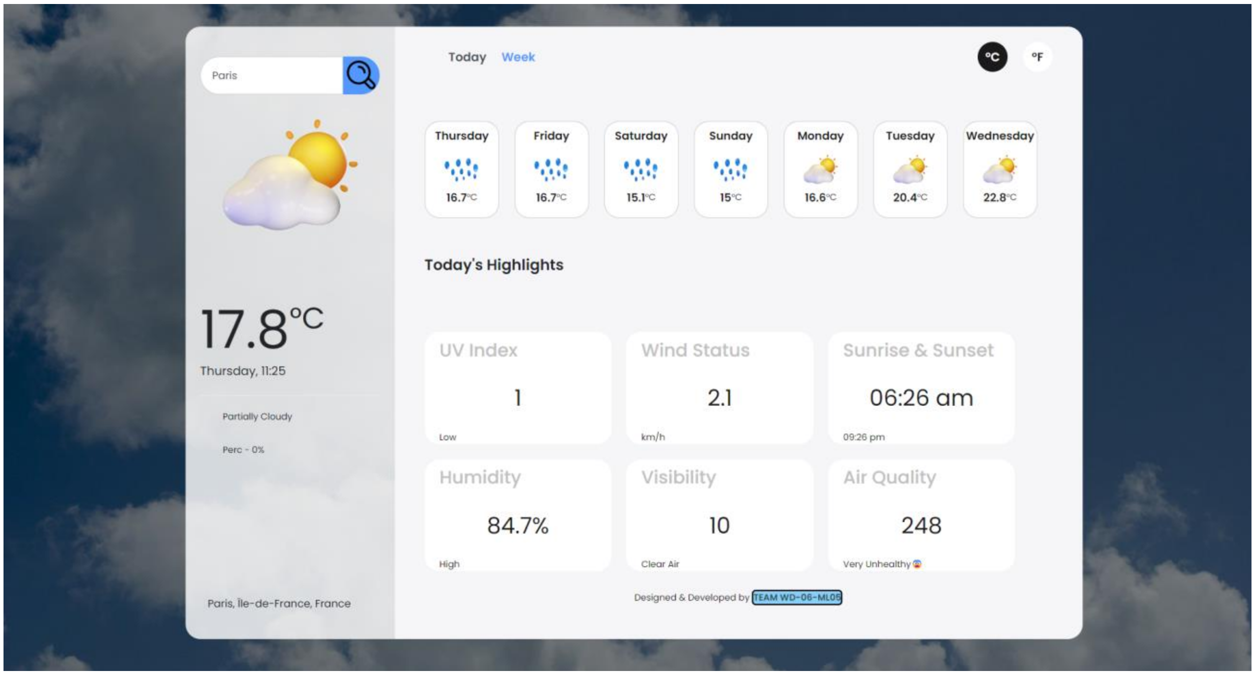
Sunrise & Sunset
05:08 am
06:16 pm

Humidity
89%
High

Visibility
4
Very Light Mist

Air Quality
170
Unhealthy

Designed & Developed by TEAM WD-06-ML05



Paris

17.8°C

Thursday, 11:25

Partially Cloudy

Perc - 0%

Paris, Île-de-France, France

Today Week

°C °F

Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday
16.7°C	16.7°C	15.1°C	15°C	16.6°C	20.4°C	22.8°C

Today's Highlights

UV Index
1
Low

Wind Status
2.1
km/h

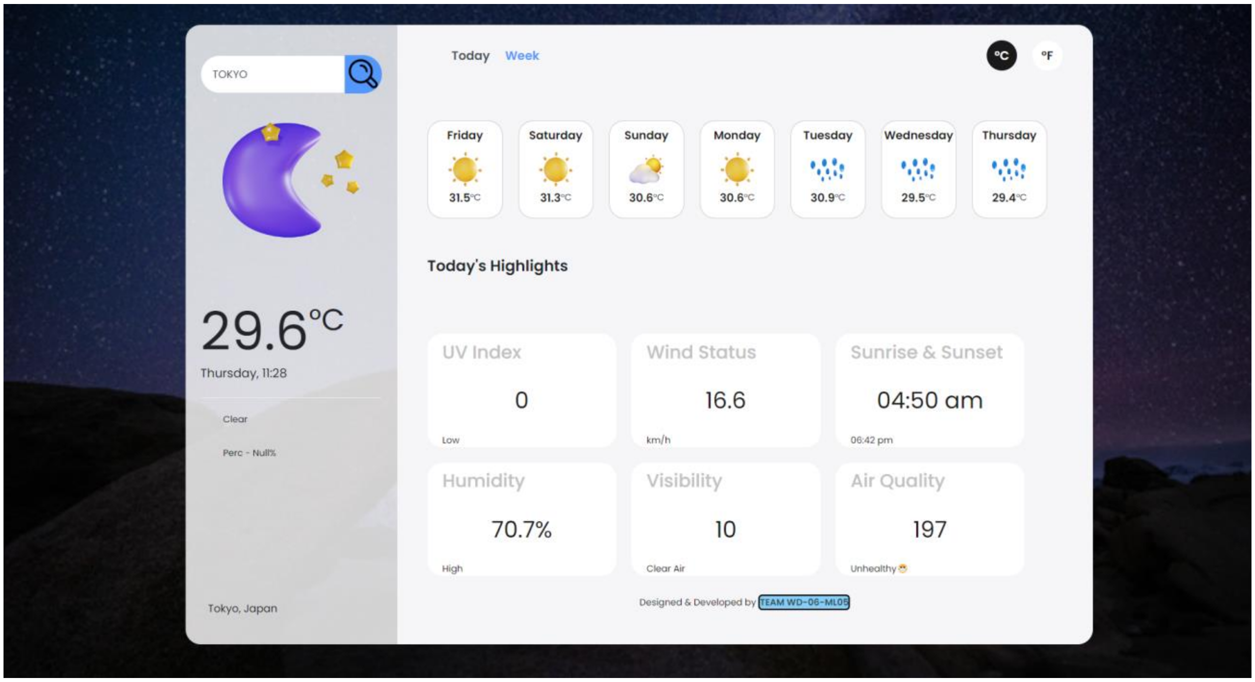
Sunrise & Sunset
06:26 am
09:26 pm

Humidity
84.7%
High

Visibility
10
Clear Air

Air Quality
248
Very Unhealthy

Designed & Developed by TEAM WD-06-ML05



TOKYO

29.6°C

Thursday, 11:28

Clear

Perc - Null%

Tokyo, Japan

Today Week

°C °F

Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday
31.5°C	31.3°C	30.6°C	30.6°C	30.9°C	29.5°C	29.4°C

Today's Highlights

UV Index
0
Low

Wind Status
16.6
km/h

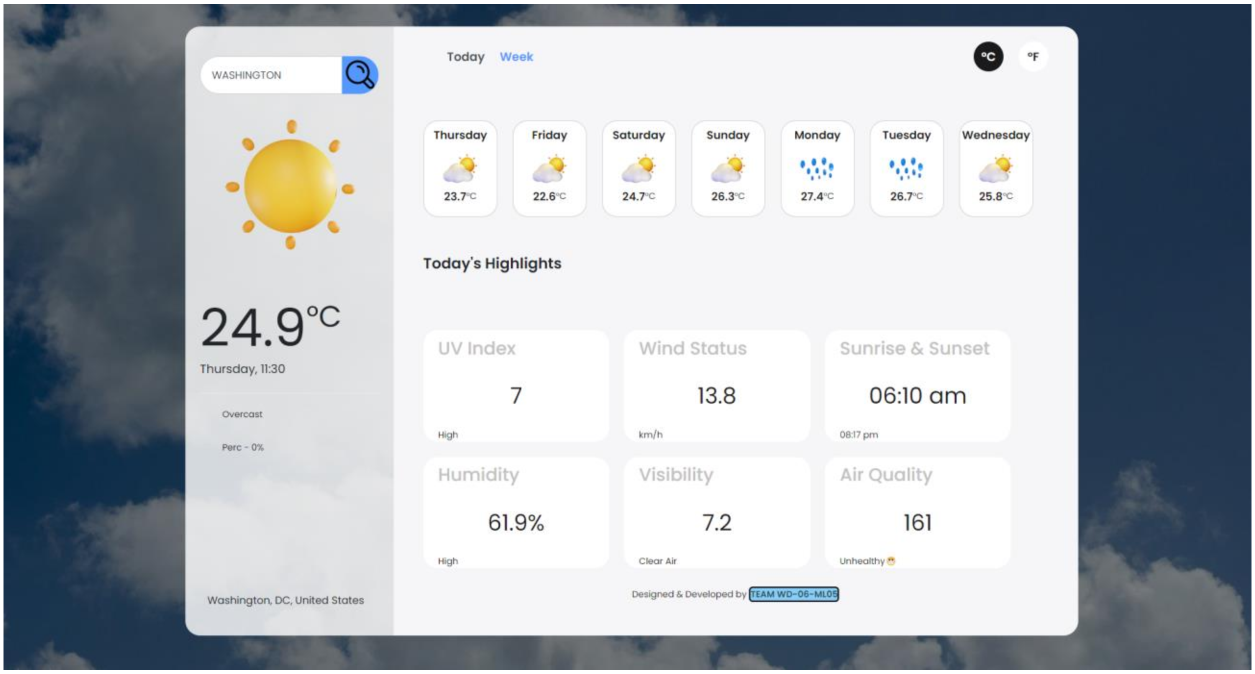
Sunrise & Sunset
04:50 am
06:42 pm

Humidity
70.7%
High

Visibility
10
Clear Air

Air Quality
197
Unhealthy

Designed & Developed by TEAM WD-06-ML05



SOURCE CODE:

<u>HTML CODE</u>(File name: **index.html**)

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Weather App</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">
    <link rel="stylesheet" href="style.css" type="text/css">
</head>

  <body>
    <div class="wrapper">
```

```
<div class="sidebar">
  <div>
    <form class="search" id="search">
      <input type="text" id="query" placeholder="Type Your City Name" />
      <button><i class="fas fa-search"><img src="https://cdn0.iconfinder.com/data/icons/business-and-finance-86/512/business_finance_money-27-64.png"></i></button>
    </form>
    <div class="weather-icon">
      <img id="icon" src="icons/sun/4.png" alt="" />
    </div>
    <div class="temperature">
      <h1 id="temp">0</h1>
      <span class="temp-unit">°C</span>
    </div>
    <div class="date-time">
      <p id="date-time">Monday, 12:00</p>
    </div>
    <div class="divider"></div>
    <div class="condition-rain">
      <div class="condition">
        <i class="fas fa-cloud"></i>
        <p id="condition">condition</p>
      </div>
      <div class="rain">
        <i class="fas fa-tint"></i>
        <p id="rain">perc - 0%</p>
      </div>
    </div>
  </div>
  <div class="location">
    <div class="location-icon">
      <i class="fas fa-map-marker-alt"></i>
    </div>
    <div class="location-text">
      <p id="location">location</p>
    </div>
  </div>
</div>
<div class="main">
  <nav>
    <ul class="options">
      <button class="hourly">today</button>
      <button class="week active">week</button>
    </ul>
    <ul class="options units">
      <button class="celcius active">°C</button>
      <button class="fahrenheit">°F</button>
    </ul>
  </nav>
  <div class="cards" id="weather-cards"></div>
  <div class="highlights">
    <h2 class="heading">today's highlights</h2>
    <div class="cards">
      <div class="card2">
        <h4 class="card-heading">UV Index</h4>
        <div class="content">
          <p class="uv-index">0</p>
          <p class="uv-text">Low</p>
  </div>
</div>
<div class="card2">
  <h4 class="card-heading">Wind Status</h4>
  <div class="content">
    <p class="wind-speed">0</p>
    <p>km/h</p>
  </div>
</div>
<div class="card2">
  <h4 class="card-heading">Sunrise & Sunset</h4>
  <div class="content">
    <p class="sun-rise">0</p>
    <p class="sun-set">0</p>
  </div>
</div>
<div class="card2">
  <h4 class="card-heading">Humidity</h4>
  <div class="content">
    <p class="humidity">0</p>
    <p class="humidity-status">Normal</p>
  </div>
</div>
<div class="card2">
  <h4 class="card-heading">Visibility</h4>
  <div class="content">
    <p class="visibilty">0</p>
    <p class="visibilty-status">Normal</p>
  </div>
```

```
       </div>
       <div class="card2">
        <h4 class="card-heading">Air Quality</h4>
        <div class="content">
         <p class="air-quality">0</p>
         <p class="air-quality-status">Normal</p>
        </div>
       </div>
      </div>
     </div>
     <p class="credits">Designed & Developed by <a href="https://www.linkedin.com/in/arshad-ali-bab2b9217/">TEAM WD-06-ML05 </a></p>
    </div>
   </div>


  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>
  <script src="script.js"></script>
 </body>
</html>
```

<u>CSS CODE</u>(File name: **style.css**)

```
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap");


:root {
 --primary-color: #5598fd;
}


* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: "Poppins", sans-serif;
}
body {
 display: flex;
 justify-content: center;
 min-height: 100vh;
 min-width: 1000px;
 padding: 50px;
 background: var(--primary-color);
 background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),
   url("./images/cd.jpg");
 background-size: cover;
 background-position: center;
 transition: background-image 0.3s ease;
}
img {
 width: 100%;
}
.wrapper {
 display: flex;
 width: 1200px;
 min-width: 900px;
 border-radius: 20px;
 overflow: hidden;
}
.sidebar {
 width: 30%;
 min-width: 250px;
 padding: 20px;
 background: rgba(255, 255, 255, 0.815);
 display: flex;
 flex-direction: column;
 justify-content: space-between;
}

.search {
 display: flex;
 align-items: center;
 justify-content: space-between;
 margin-bottom: 30px;
 margin-top: 20px;
 position: relative;
}
.search input {
 width: 100%;
 height: 50px;
 border: 1px solid #ced4da;
 border-top-left-radius: 25px;
 border-bottom-left-radius: 25px;
 padding: 0 15px;
 font-size: 14px;
 color: #495057;
}
.search input:focus {
 outline: none;
 border: 1px solid var(--primary-color);
}
.search button {
 min-width: 40px;
 height: 50px;
 border: none;
 border-top-right-radius: 25px;
 border-bottom-right-radius: 25px;
 background: var(--primary-color);
 color: #fff;
 font-size: 14px;
 cursor: pointer;
}

.search button:hover {
 background-color: #3e1af3;
}

.search ul {
 max-height: 300px;
 overflow-y: auto;
 position: absolute;
 width: 100%;
 top: 40px;
 border-radius: 5px;
 transition: all 0.3s ease;
 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
 background-color: #fff;
}
```

```
.search ul li {
  padding: 10px 15px;
  border-bottom: 1px solid #f1f1f1;
  cursor: pointer;
  text-transform: capitalize;
}
.search ul li:last-child {
  border-bottom: none;
}
.search ul li:hover {
  background-color: #f1f1f1;
}
.search ul li.active {
  background-color: #f1f1f1;
}
.weather-icon {
  width: 100%;
  height: 150px;
  text-align: center;
  margin-top: 20px;
  margin-bottom: 100px;
}
.weather-icon #icon {
  width: 80%;
  object-fit: cover;
}
.temperature {
  display: flex;
}
```

```
.temperature #temp {
  font-size: 70px;
  font-weight: 100;
  line-height: 1;
}
.temperature span {
  font-size: 40px;
  margin-top: -10px;
  display: block;
}
.divider {
  width: 100%;
  height: 1px;
  background: #e9ecef;
  margin: 20px 0;
}
.condition-rain {
  font-size: 12px;
  text-transform: capitalize;
}
.condition-rain div {
  display: flex;
  align-items: center;
  gap: 10px;
  margin-bottom: 10px;
}
.condition-rain div i {
  width: 20px;
}
```

```
.location {
  display: flex;
  align-items: center;
  font-size: 14px;
  gap: 10px;
  margin-top: 10px;
}
.main {
  width: 100%;
  min-width: 400px;
  padding: 20px 40px;
  background-color: #f6f6f8;
  position: relative;
  padding-bottom: 90px;
}

.main nav {
  display: flex;
  align-items: center;
  justify-content: space-between;
}
.main nav .options {
  display: flex;
  gap: 20px;
  align-items: center;
}
.main nav .options button {
  border: none;
  background: none;
  font-size: 16px;
  font-weight: 600;
  color: #495057;
  cursor: pointer;
  text-transform: capitalize;
}
.main nav .options button.active {
  color: var(--primary-color);
}

.main nav .units button {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  color: #1a1a1a;
  background-color: #fff;
}
.main nav .units button.active {
  color: #fff;
  background-color: #1a1a1a;
}
.main .cards {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  margin-top: 50px;
}

.cards .card {
  width: 100px;
  height: 130px;
  border-radius: 20px;
  color: #1a1a1a;
  background-color: #fff;
  text-align: center;
  padding: 10px 0;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}
.card h2 {
  font-size: 15px;
  font-weight: 600;
}
.card .card-icon {
  width: 50%;
  margin: 0 auto;
}
.card .day-temp {
  font-size: 12px;
  display: flex;
  justify-content: center;
  display: flex;
}
.highlights {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  margin-top: 50px;
}
.highlights .heading {
  width: 100%;
  font-size: 20px;
  font-weight: 600;
  text-transform: capitalize;
}

.card2 {
  width: 250px;
  height: 150px;
  border-radius: 20px;
  color: #1a1a1a;
  background-color: #fff;
  padding: 10px 20px;
  display: flex;
  flex-direction: column;
}
.card2 .card-heading {
  color: #c2c2c2;
}

.card2 .content {
  margin-top: 20px;
}
.card2 .content p:first-child {
  text-align: center;
  font-size: 30px;
```

```
}
.card2 .content p:nth-child(2) {
  font-size: 12px;
  margin-top: 20px;
  text-align: left;
}
.credits {
  text-align: center;
  font-size: 12px;
  color:#1a1a1a;
  position: absolute;
  bottom: 30px;
  left: 50%;
  transform: translateX(-50%);
}
.credits a{
  color:darkslategrey;
  text-decoration: none;
  font-weight: bolder;
  border: 2px solid black;
  border-radius: 5px 5px;
  background-color: lightskyblue;
}
.credits a:hover{
  background-color: bisque;
  color: red;
  border-color: red;
}
```

JS CODE (File name: **script.js**)

```
// const options = {
//         method: 'GET',
//         headers: {
//                 'X-RapidAPI-Key': '77a45e249cmsh792cd7445b30ea1p146476jsna8123226ed35',
//                 'X-RapidAPI-Host': 'weather-by-api-ninjas.p.rapidapi.com'
//         }
// };

// fetch('https://weather-by-api-ninjas.p.rapidapi.com/v1/weather?units=metric&q=Delhi', options)
//    .then(response => response.json())
//    .then(response => console.log(response))
//    .then(err => console.error(err));



const temp = document.getElementById("temp"),
  date = document.getElementById("date-time"),
  condition = document.getElementById("condition"),
  rain = document.getElementById("rain"),
  mainIcon = document.getElementById("icon"),
  currentLocation = document.getElementById("location"),
  uvIndex = document.querySelector(".uv-index"),
  uvText = document.querySelector(".uv-text"),
  windSpeed = document.querySelector(".wind-speed"),
  sunRise = document.querySelector(".sun-rise"),
  sunSet = document.querySelector(".sun-set"),
  humidity = document.querySelector(".humidity"),
  visibilty = document.querySelector(".visibilty"),
  humidityStatus = document.querySelector(".humidity-status"),
  airQuality = document.querySelector(".air-quality"),
  airQualityStatus = document.querySelector(".air-quality-status"),
  visibilityStatus = document.querySelector(".visibilty-status"),
  searchForm = document.querySelector("#search"),
  search = document.querySelector("#query"),
  celciusBtn = document.querySelector(".celcius"),
  fahrenheitBtn = document.querySelector(".fahrenheit"),
  tempUnit = document.querySelectorAll(".temp-unit"),
  hourlyBtn = document.querySelector(".hourly"),
  weekBtn = document.querySelector(".week"),
  weatherCards = document.querySelector("#weather-cards");

let currentCity = "";
let currentUnit = "c";
let hourlyorWeek = "week";

// function to get date and time
function getDateTime() {
  let now = new Date(),
    hour = now.getHours(),
    minute = now.getMinutes();

  let days = [
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday"
  ];
  // 12 hours format
  hour = hour % 12;
  if (hour < 10) {
    hour = "0" + hour;
  }
  if (minute < 10) {
    minute = "0" + minute;
  }
  let dayString = days[now.getDay()];
  return `${dayString}, ${hour}:${minute}`;
}

//Updating date and time
date.innerText = getDateTime();
setInterval(() => {
  date.innerText = getDateTime();
}, 1000);

// function to get public ip address
function getPublicIp() {
  fetch("https://geolocation-db.com/json/", {
    method: "GET",
    headers: {}
  })
    .then((response) => response.json())
    .then((data) => {
      currentCity = data.city;
      getWeatherData(data.city, currentUnit, hourlyorWeek);
    })
    .catch((err) => {
      console.error(err);
    });
}

getPublicIp();

// function to get weather data
function getWeatherData(city, unit, hourlyorWeek) {
  fetch(

`https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/${city}?unitGroup=metric&key=EJ6UBL2JEQGYB3AA4ENASN62J&contentType=json`,
    {
      method: "GET",
      headers: {}
    }
  )
    .then((response) => response.json())
    .then((data) => {
      let today = data.currentConditions;
      if (unit === "c") {
        temp.innerText = today.temp;
      } else {
        temp.innerText = celciusToFahrenheit(today.temp);
      }
      currentLocation.innerText = data.resolvedAddress;
      condition.innerText = today.conditions;
      rain.innerText = "Perc - " + today.precip + "%";
      uvIndex.innerText = today.uvindex;
      windSpeed.innerText = today.windspeed;
      measureUvIndex(today.uvindex);
      mainIcon.src = getIcon(today.icon);
      changeBackground(today.icon);
      humidity.innerText = today.humidity + "%";
      updateHumidityStatus(today.humidity);
      visibilty.innerText = today.visibility;
      updateVisibiltyStatus(today.visibility);
      airQuality.innerText = today.winddir;
      updateAirQualityStatus(today.winddir);
      if (hourlyorWeek === "hourly") {
        updateForecast(data.days[0].hours, unit, "day");
      } else {
        updateForecast(data.days, unit, "week");
      }
      sunRise.innerText = covertTimeTo12HourFormat(today.sunrise);
      sunSet.innerText = covertTimeTo12HourFormat(today.sunset);
    })
    .catch((err) => {
      alert("City not found in our database");
    });
}

//function to update Forecast
function updateForecast(data, unit, type) {
  weatherCards.innerHTML = "";
```

```
let day = 0;

let numCards = 0;

if (type === "day") {

  numCards = 24;

} else {

  numCards = 7;

}

for (let i = 0; i < numCards; i++) {

  let card = document.createElement("div");

  card.classList.add("card");

  let dayName = getHour(data[day].datetime);

  if (type === "week") {

    dayName = getDayName(data[day].datetime);

  }

  let dayTemp = data[day].temp;

  if (unit === "f") {

    dayTemp = celciusToFahrenheit(data[day].temp);

  }

  let iconCondition = data[day].icon;

  let iconSrc = getIcon(iconCondition);

  let tempUnit = "°C";

  if (unit === "f") {

    tempUnit = "°F";

  }

  card.innerHTML = `

              <h2 class="day-name">${dayName}</h2>

          <div class="card-icon">

            <img src="${iconSrc}" class="day-icon" alt="" />

          </div>
```

```
      <div class="day-temp">
       <h2 class="temp">${dayTemp}</h2>
       <span class="temp-unit">${tempUnit}</span>
      </div>
 `;
  weatherCards.appendChild(card);
  day++;
 }
}

// function to change weather icons
function getIcon(condition) {
 if (condition === "partly-cloudy-day") {
  return "https://i.ibb.co/PZQXH8V/27.png";
 } else if (condition === "partly-cloudy-night") {
  return "https://i.ibb.co/Kzkk59k/15.png";
 } else if (condition === "rain") {
  return "https://i.ibb.co/kBd2NTS/39.png";
 } else if (condition === "clear-day") {
  return "https://i.ibb.co/rb4rrJL/26.png";
 } else if (condition === "clear-night") {
  return "https://i.ibb.co/1nxNGHL/10.png";
 } else {
  return "https://i.ibb.co/rb4rrJL/26.png";
 }
}

// function to change background depending on weather conditions
function changeBackground(condition) {
  const body = document.querySelector("body");
  let bg = "";
  if (condition === "partly-cloudy-day") {
    bg = "https://i.ibb.co/qNv7NxZ/pc.webp";
  } else if (condition === "partly-cloudy-night") {
    bg = "https://i.ibb.co/RDfPqXz/pcn.jpg";
  } else if (condition === "rain") {
    bg = "https://i.ibb.co/h2p6Yhd/rain.webp";
  } else if (condition === "clear-day") {
    bg = "https://i.ibb.co/WGry01m/cd.jpg";
  } else if (condition === "clear-night") {
    bg = "https://i.ibb.co/kqtZ1Gx/cn.jpg";
  } else {
    bg = "https://i.ibb.co/qNv7NxZ/pc.webp";
  }
  body.style.backgroundImage = `linear-gradient( rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5) ),url(${bg})`;
}

//get hours from hh:mm:ss
function getHour(time) {
  let hour = time.split(":")[0];
  let min = time.split(":")[1];
  if (hour > 12) {
    hour = hour - 12;
    return `${hour}:${min} PM`;
  } else {
    return `${hour}:${min} AM`;
  }
}
```

```
// convert time to 12 hour format
function covertTimeTo12HourFormat(time) {
  let hour = time.split(":")[0];
  let minute = time.split(":")[1];
  let ampm = hour >= 12 ? "pm" : "am";
  hour = hour % 12;
  hour = hour ? hour : 12; // the hour '0' should be '12'
  hour = hour < 10 ? "0" + hour : hour;
  minute = minute < 10 ? minute : minute;
  let strTime = hour + ":" + minute + " " + ampm;
  return strTime;
}

// function to get day name from date
function getDayName(date) {
  let day = new Date(date);
  let days = [
    "Sunday",
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday"
  ];
  return days[day.getDay()];
}
```

```
// function to get uv index status
function measureUvIndex(uvIndex) {
  if (uvIndex <= 2) {
    uvText.innerText = "Low";
  } else if (uvIndex <= 5) {
    uvText.innerText = "Moderate";
  } else if (uvIndex <= 7) {
    uvText.innerText = "High";
  } else if (uvIndex <= 10) {
    uvText.innerText = "Very High";
  } else {
    uvText.innerText = "Extreme";
  }
}

// function to get humidity status
function updateHumidityStatus(humidity) {
  if (humidity <= 30) {
    humidityStatus.innerText = "Low";
  } else if (humidity <= 60) {
    humidityStatus.innerText = "Moderate";
  } else {
    humidityStatus.innerText = "High";
  }
}

// function to get visibility status
function updateVisibiltyStatus(visibility) {
  if (visibility <= 0.03) {
    visibilityStatus.innerText = "Dense Fog";
  } else if (visibility <= 0.16) {
    visibilityStatus.innerText = "Moderate Fog";
  } else if (visibility <= 0.35) {
    visibilityStatus.innerText = "Light Fog";
  } else if (visibility <= 1.13) {
    visibilityStatus.innerText = "Very Light Fog";
  } else if (visibility <= 2.16) {
    visibilityStatus.innerText = "Light Mist";
  } else if (visibility <= 5.4) {
    visibilityStatus.innerText = "Very Light Mist";
  } else if (visibility <= 10.8) {
    visibilityStatus.innerText = "Clear Air";
  } else {
    visibilityStatus.innerText = "Very Clear Air";
  }
}

// function to get air quality status
function updateAirQualityStatus(airquality) {
  if (airquality <= 50) {
    airQualityStatus.innerText = "Good 👌";
  } else if (airquality <= 100) {
    airQualityStatus.innerText = "Moderate 😐";
  } else if (airquality <= 150) {
    airQualityStatus.innerText = "Unhealthy for Sensitive Groups 😷";
  } else if (airquality <= 200) {
    airQualityStatus.innerText = "Unhealthy 😷";
  } else if (airquality <= 250) {
    airQualityStatus.innerText = "Very Unhealthy😨";
  } else {
    airQualityStatus.innerText = "Hazardous😱";
  }
}

// function to handle search form
searchForm.addEventListener("submit", (e) => {
  e.preventDefault();
  let location = search.value;
  if (location) {
    currentCity = location;
    getWeatherData(location, currentUnit, hourlyorWeek);
  }
});

// function to conver celcius to fahrenheit
function celciusToFahrenheit(temp) {
  return ((temp * 9) / 5 + 32).toFixed(1);
}

var currentFocus;
search.addEventListener("input", function (e) {
  removeSuggestions();
  var a,
    b,
    i,
    val = this.value;
```

```
if (!val) {
  return false;
}
currentFocus = -1;

a = document.createElement("ul");
a.setAttribute("id", "suggestions");

this.parentNode.appendChild(a);

for (i = 0; i < cities.length; i++) {
  /*check if the item starts with the same letters as the text field value:*/
  if (
    cities[i].name.substr(0, val.length).toUpperCase() == val.toUpperCase()
  ) {
    /*create a li element for each matching element:*/
    b = document.createElement("li");
    /*make the matching letters bold:*/
    b.innerHTML =
      "<strong>" + cities[i].name.substr(0, val.length) + "</strong>";
    b.innerHTML += cities[i].name.substr(val.length);
    /*insert a input field that will hold the current array item's value:*/
    b.innerHTML += "<input type='hidden' value='" + cities[i].name + "'>";
    /*execute a function when someone clicks on the item value (DIV element):*/
    b.addEventListener("click", function (e) {
      /*insert the value for the autocomplete text field:*/
      search.value = this.getElementsByTagName("input")[0].value;
      removeSuggestions();
    });
```

```
      a.appendChild(b);
    }
  }
});
/*execute a function presses a key on the keyboard:*/
search.addEventListener("keydown", function (e) {
  var x = document.getElementById("suggestions");
  if (x) x = x.getElementsByTagName("li");
  if (e.keyCode == 40) {
    /*If the arrow DOWN key
      is pressed,
      increase the currentFocus variable:*/
    currentFocus++;
    /*and and make the current item more visible:*/
    addActive(x);
  } else if (e.keyCode == 38) {
    /*If the arrow UP key
      is pressed,
      decrease the currentFocus variable:*/
    currentFocus--;
    /*and and make the current item more visible:*/
    addActive(x);
  }
  if (e.keyCode == 13) {
    /*If the ENTER key is pressed, prevent the form from being submitted,*/
    e.preventDefault();
    if (currentFocus > -1) {
      /*and simulate a click on the "active" item:*/
      if (x) x[currentFocus].click();
    }
  }
});
function addActive(x) {
  /*a function to classify an item as "active":*/
  if (!x) return false;
  /*start by removing the "active" class on all items:*/
  removeActive(x);
  if (currentFocus >= x.length) currentFocus = 0;
  if (currentFocus < 0) currentFocus = x.length - 1;
  /*add class "autocomplete-active":*/
  x[currentFocus].classList.add("active");
}
function removeActive(x) {
  /*a function to remove the "active" class from all autocomplete items:*/
  for (var i = 0; i < x.length; i++) {
    x[i].classList.remove("active");
  }
}

function removeSuggestions() {
  var x = document.getElementById("suggestions");
  if (x) x.parentNode.removeChild(x);
}

fahrenheitBtn.addEventListener("click", () => {
  changeUnit("f");
});
```

```
celciusBtn.addEventListener("click", () => {
  changeUnit("c");
});

// function to change unit
function changeUnit(unit) {
  if (currentUnit !== unit) {
    currentUnit = unit;
    tempUnit.forEach((elem) => {
      elem.innerText = `°${unit.toUpperCase()}`;
    });
    if (unit === "c") {
      celciusBtn.classList.add("active");
      fahrenheitBtn.classList.remove("active");
    } else {
      celciusBtn.classList.remove("active");
      fahrenheitBtn.classList.add("active");
    }
    getWeatherData(currentCity, currentUnit, hourlyorWeek);
  }
}

hourlyBtn.addEventListener("click", () => {
  changeTimeSpan("hourly");
});
weekBtn.addEventListener("click", () => {
  changeTimeSpan("week");
});
```

```
// function to change hourly to weekly or vice versa
function changeTimeSpan(unit) {
  if (hourlyorWeek !== unit) {
    hourlyorWeek = unit;
    if (unit === "hourly") {
      hourlyBtn.classList.add("active");
      weekBtn.classList.remove("active");
    } else {
      hourlyBtn.classList.remove("active");
      weekBtn.classList.add("active");
    }
    getWeatherData(currentCity, currentUnit, hourlyorWeek);
  }
}


// Cities add your own to get in search

cities = [
  {
    country: "in",
    name: "Delhi",
    lat: "28.679079",
    lng: "77.069710",
  },
  {
    country: "in",
    name: "New Delhi",
    lat: "28.6139",
    lng: "77.2090",
  },
  {
    country: "in",
    name: "Agra",
    lat: "27.17042035",
    lng: "78.01502071",
  },
  {
    country: "in",
    name: "Aligarh",
    lat: "27.89221092",
    lng: "78.06178788",
  },
  {
    country: "in",
    name: "Lucknow",
    lat: "30.67791",
    lng: "71.74344",
  },
  {
    country: "in",
    name: "Allahabad",
    lat: "25.45499534",
    lng: "81.84000688",
  },
  {
    country: "in",
    name: "Bareilly",
    lat: "28.34538739",
    lng: "79.41999955",
  },
  {
    country: "in",
    name: "Budaun",
    lat: "28.03000612",
    lng: "79.08999385",
  },
  {
    country: "in",
    name: "Bulandshahr",
    lat: "28.4103705",
    lng: "77.84841589",
  },
  {
    country: "in",
    name: "Etawah",
    lat: "26.78545677",
    lng: "79.01495968",
  },
  {
    country: "in",
    name: "Faizabad",
    lat: "26.75039431",
    lng: "82.17001257",
  },
  {
    country: "in",
    name: "Fatehpur",
    lat: "25.88036989",
    lng: "80.80001868",
  },
  {
    country: "in",
    name: "Farrukhabad",
    lat: "27.387049",
    lng: "79.588127",
  },
  {
    country: "in",
    name: "Ghaziabad",
    lat: "28.66038108",
    lng: "77.40839107",
  },
  {
    country: "in",
    name: "Gorakhpu",
    lat: "26.75039431",
    lng: "83.38001623",
  },
  {
    country: "in",
    name: "jaipur",
    lat: "26.922070",
    lng: "75.778885",
  },
  {
    country: "in",
    name: "Jodhpur",
    lat: "26.223370",
    lng: "72.998917",
  },
  {
    country: "in",
    name: "Udaipur",
    lat: "24.571270",
    lng: "73.691544",
  },
  {
    country: "in",
    name: "Jaisalmer",
    lat: "26.9157",
    lng: "70.9083",
  },
  {
    country: "in",
    name: "Patna",
    lat: "25.5941",
    lng: "85.1376",
  },
  {
    country: "in",
    name: "Kanpur",
    lat: "26.4499",
    lng: "80.3319",
  },
  {
    country: "in",
    name: "Kolkata",
    lat: "22.5726",
    lng: "88.363892",
  },
  {
    country: "in",
    name: "Mumbai",
    lat: "19.076090",
    lng: "72.877426",
  },
];
```