

# Weather.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css">

  <meta http-equiv="X-UA-Compatible" content="IE=edge" />

  <link rel="icon" type="image/x-icon"

    href="https://cdn2.iconfinder.com/data/icons/weather-flat-14/64/weather02-512.png">

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

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

  <title>weather App</title>

</head>

<body>

  <div class="container">

    <div class="search wrapper" id="search">

      <div class="new">

        

        <h1>Get the Weather</h1>

      </div>

      <div class="input-container">
```

```
<input type="text" placeholder="Enter City Name and Hit Enter" spellcheck="false" /><br>
<button class="btn btn-outline-light" style="margin: 10px;">Locate Live Location</button>
</div>

<div class="error-container">

  <p class="error-text" id="error"></p>

</div><br>

<h4 style="color:white;">Designed and Developed by Kavi Prakash M</h4>
</div>

<div class="place wrapper">

  <div class="time-zone" id="time-zone">Asia/Kolkata</div>

  <div class="country" id="country">in</div>

</div>

<div class="current wrapper">

  <div class="date-container">

    <div class="time" id="time">

      <span id="hours">12</span>:<span id="minutes">20</span>

    </div>

    <div class="date" id="date">

      <span id="day">monday</span>,<span id="date-today">15</span>

      <span id="month">aug</span>

    </div>

    <div class="others" id="current-weather">

      <div class="weather-condition-container">

        <div class="weather-condition-wrapper">
```

```
<div class="weather-condition">

  <div class="icon"></div>

  <div class="condition">temp <span></span>&#8451;</div>

</div>

<span class="description">cloudy</span>

</div>

</div>

<div class="other-details">

  <div class="grid-left">

    <div class="spaced-content weather-deatils">

      <div class="box"><div>min_temp</div>-

      <div id="min-temp">30 &#8451;<span></span></div>

    </div></div>

    <div class="spaced-content weather-deatils">

      <div class="box">

        <div>windspeed</div>-

        <div id="wind-speed">20 <span>Km/h</span></div>

      </div></div>

    </div>

    <div class="grid-right">

      <div class="spaced-content weather-deatils">

        <div class="box">

          <div>max_temp</div>-

          <div id="max-temp">6.30 <span></span></div>
```

</div></div>

<div class="spaced-content weather-deatils">

<div class="box">

<div>humidity</div>-

<div id="humidity">50<span>%</span></div>

</div></div>

</div>

</div>

</div>

<div class="future-forecast">

<div class="today" id="current-temp">



<div class="other">

<div class="day">Monday</div>

<div class="temp">Night - 25.6 &#176;</div>

<div class="temp1">Day - 25.6 &#176;</div>

</div>

</div>

<div class="weather-forecast" id="weather-forecast">

<div class="weather-forecast-item">

<div class="day">Tuesday</div>

```

```

```
Night -<div class="temp"> 25.6 &#176;</div>
```

```
Day -<div class="temp1"> 25.6 &#176;</div>
```

```
</div>
```

```
<div class="weather-forecast-item">
```

```
<div class="day">Wednesday</div>
```

```

```

```
Night -<div class="temp"> 23.2 &#176;</div>
```

```
Day -<div class="temp1"> 22.3 &#176;</div>
```

```
</div>
```

```
<div class="weather-forecast-item">
```

```
<div class="day">Thursday</div>
```

```

```

```
Night -<div class="temp"> 26.5 &#176;</div>
```

```
Day -<div class="temp1"> 23.5 &#176;</div>
```

```
</div>
```

```
<div class="weather-forecast-item">
```

```
<div class="day">Friday</div>
```

```

```

Night -<div class="temp"> 26.7 &#176;</div>

Day -<div class="temp1"> 25.4 &#176;</div>

</div>

<div class="weather-forecast-item">

<div class="day">Saturday</div>



Night -<div class="temp"> 27.5 &#176;</div>

Day -<div class="temp1"> 26.4 &#176;</div>

</div>

<div class="weather-forecast-item">

<div class="day">Sunday</div>



Night -<div class="temp"> 27.6 &#176;</div>

Day -<div class="temp1"> 25.7 &#176;</div>

</div>

</div>

</div>

</div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

**Style.css**

```
@import url('https://fonts.googleapis.com/css2?
family=Poppins:wght@100;200;400;700&display=swap');
```

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

```
body {
  background: linear-gradient(75deg,black, purple);
  font-family: 'Poppins', sans-serif;
  min-height: 100vh;
}
```

```
.container {
  display: flex;
  flex-direction: column;
  padding: 30px;
  align-items: center;
}
```

```
.container h1 {
  color: white;
  letter-spacing: 8px;
```



```
}
```

```
.container.active .search {  
  display: none;  
}
```

```
.search .input-container input {  
  width: 100%;  
  border: none;  
  font-size: 1rem;  
  border-radius: 10px;  
}
```

```
.search .error-container .error-text {  
  font-size: .8rem;  
  font-weight: 400;  
  letter-spacing: .2rem;  
  text-transform: capitalize;  
}
```

```
.error-text.error {  
  color: white;  
  background: rgba(255, 0, 0, 0.6);  
  padding: 10px 25px;
```

```
border-radius: 8px;  
}
```

```
.error-text.wait {  
  color: white;  
  padding: 10px 25px;  
  background: rgba(42, 43, 42, 0.418);  
  border-radius: 8px;  
}
```

```
.container .current {  
  justify-content: space-between;  
}
```

```
.date-container .time {  
  font-size: 3rem;  
  font-weight: 100;  
}
```

```
.date-container .time #am-pm {  
  font-size: 2.5rem;  
  text-transform: uppercase;  
}
```

```
.date-container .date {  
    font-size: 1.7rem;  
}
```

```
.place {  
    text-align: center;  
    display: none;  
    flex-direction: column;  
    color: white;  
}
```

```
.container.active .place {  
    display: flex;  
}
```

```
.place .time-zone {  
    font-size: 2rem;  
    font-weight: 100;  
}
```

```
.place .country {  
    font-size: 1rem;  
    font-weight: 700;  
}
```

```
.current {  
    display: none;  
    color: white;  
}
```

```
.container.active .current {  
    display: flex;  
}
```

```
.current .others {  
    font-size: 1rem;  
    font-weight: 200;  
    text-transform: capitalize;  
    background-color: rgba(42, 43, 42, 0.418);  
    border-radius: 10px;  
}
```

```
.current .others .weather-condition-container {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    padding: 20px;  
    font-size: 1.5rem;
```

```
margin: 15px;  
}
```

```
.current .others .other-details {  
    display: flex;  
    flex-direction: column;  
    padding: 20px;  
}
```

```
.others .other-details .grid-left,  
.grid-right {  
    display: flex;  
    flex-direction: column;  
    margin: 10px;  
}
```

```
.others .other-details .grid-left .spaced-content {  
    display: flex;  
    justify-content: space-between;  
}
```

```
.others .other-details .grid-right .spaced-content {  
    display: flex;  
    justify-content: space-between;
```

```
}
```

```
.other-details .grid-left .spaced-content div {
```

```
    font-size: 1rem;
```

```
}
```

```
.other-details .grid-right .spaced-content div {
```

```
    font-size: 1rem;
```

```
}
```

```
@media only screen and (min-width: 769px) {
```

```
    h1 {
```

```
        font-size: 3rem;
```

```
    }
```

```
.container .search {
```

```
    margin-top: 15%;
```

```
    padding: 15px 15px;
```

```
    width: 100%;
```

```
    align-items: center;
```

```
    justify-content: center;
```

```
}
```

```
.search .input-container {
```

```
    font-size: 1.2rem;
    padding: 10px 0px;
}
```

```
.search .input-container input {
    width: 70%;
    padding: 20px 15px;
}
```

```
.search .error-container .error-text {
    font-size: .9rem;
}
```

```
.place {
    display: none;
}
```

```
.current {
    display: none;
}
```

```
.container.active .current {
    display: flex;
}
```

```
.current .others {  
    display: flex;  
    flex-direction: row;  
}
```

```
.current .others .weather-condition-container {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    padding: 30px;  
    font-size: 2rem;  
    margin: 15px;  
}
```

```
.current .others .other-details {  
    display: flex;  
    flex-direction: row;  
    padding: 20px;  
    margin-left: 40px;  
}
```

```
.others .other-details .grid-left,  
.grid-right {
```



```
display: flex;

flex-direction: column;

justify-content: space-between;
}
```

```
.others .other-details .grid-left .spaced-content {

display: flex;

justify-content: space-around;
}
```

```
.others .other-details .grid-right .spaced-content {

display: flex;

justify-content: space-between;
}
```

```
.other-details .grid-left .spaced-content div {

margin: 0 50px;

font-size: 1.2rem;
}
```

```
.other-details .grid-right .spaced-content div {

margin: 0 50px;

font-size: 1.2rem;
}
```

```
}
```

```
.future-forecast {  
  margin-top: 20px;  
  background-color: rgba(42, 43, 42, 0.418);  
  border-radius: 10px;  
  padding: 25px;  
  display: flex;  
  color: white;  
  width: 100%;  
  align-items: center;  
  justify-content: center;  
}
```

```
.future-forecast .today {  
  display: flex;  
  align-items: center;  
  text-align: center;  
}
```

```
.future-forecast .today .day {  
  padding: 5px 15px;  
  background-color: #3c3c44;
```

```
    text-align: center;

    border-radius: 50px;
}
```

```
.future-forecast .today .temp {

    font-size: 18px;

    padding-top: 15px;
}
```

```
.future-forecast .weather-forecast {

    display: flex;
}
```

```
.weather-forecast .weather-forecast-item {

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    margin: 0px 10px;

    border: 1px solid #eee;

    border-radius: 10px;

    padding: 15px;
}
```

```
.weather-forecast .weather-forecast-item .day {  
    padding: 5px 15px;  
    background: #3c3c44;  
    border-radius: 50px;  
    text-align: center;  
}
```

```
.weather-forecast .weather-forecast-item .temp {  
    font-weight: 100;  
}
```

```
@media only screen and (max-width:730px) {  
    .future-forecast {  
        justify-content: start;  
        align-items: none;  
        overflow-y: scroll;  
    }  
}
```

```
.box  
{  
    background: rgba( 255, 255, 255, 0.35 );  
    box-shadow: 0 8px 32px 0 rgba( 31, 38, 135, 0.37 );  
    backdrop-filter: blur( 13px );  
    -webkit-backdrop-filter: blur( 13px );  
}
```

```
border-radius: 10px;

border: 1px solid rgba( 255, 255, 255, 0.18 );

display: flex;

padding: 20px 0px;

color: white;

align-items: center;

justify-content: center;

}
```

## Script.js

```
const API = "5f8bd5df4a8f994029aefd3fa0b2efe6",

currentWeatherItemsEl = document.getElementById("current-weather"),
```

```
timeZoneEL = document.getElementById("time-zone"),
countryEL = document.getElementById("country"),
weeklyForecastEl = document.getElementById("weekly-forecast"),
currentTempEl = document.getElementById("current-temp"),
inputField = document.querySelector("input"),
loactionBtn = document.querySelector("button"),
textError = document.querySelector(".error-text"),
errContainer = document.querySelector(".error-container"),
container = document.querySelector(".container"),
weatherIcon = document.querySelector(".icon"),
days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",],
months = ["January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December",];

let api;

inputField.addEventListener("keyup", (e) => {
  if (e.key == "Enter" && inputField.value != "") {
    requestApi(inputField.value);
  }
});

function requestApi(city) {
  console.log(city);

  api = `https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&cnt=7&appid=${API}`;

  fetchData();
}
```

```

locationBtn.addEventListener("click", () => {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(onSuccess, onError);
  } else {
    alert("Your browser does not support geolocation api");
  }
});

function onSuccess(position) {
  const { latitude, longitude } = position.coords;

  api = `https://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&exclude=hourly,daily&units=metric&appid=${API}`;

  fetchData();
}

function onError(error) {
  textError.innerText = error.message;
  textError.classList.add("error");
}

function fetchData() {
  textError.innerText = "Getting weather info. Please wait...";
  textError.classList.add("wait");

  fetch(api)
    .then((response) => response.json())
    .then((result) => weatherDetails(result));
}

function weatherDetails(info) {

```

```
textError.classList.replace("wait", "error");

if (info.cod == "404") {

    textError.innerText = `${inputField.value}` isn't a valid city name`;

} else {

    const city = info.name,

        timezone = info.timezone,

        { country } = info.sys,

        { description, icon } = info.weather[0],

        { humidity, temp, temp_max, temp_min } = info.main, speed = info.wind;

    console.log(city)

    container.querySelector(".place .time-zone").innerText = city;

    container.querySelector(".place .country").innerText = country;

    container.querySelector("#min-temp").innerText = temp_min;

    container.querySelector("#max_temp"), (innerText = temp_max);

    container.querySelector("#humidity"), (innerText = humidity);

    container.querySelector("#wind-speed"), (innerText = speed);

    container.querySelector(".weather-condition-container .description").innerText = description;

    container.querySelector(".weather-condition-container .condition span").innerText = temp;

    container.querySelector(".weather-forecast .weather-forecast-item .temp").innerText = temp;

    container.querySelector(".weather-forecast-item .temp1").innerText = temp_min;


    let iconImg = document.createElement("img");

    iconImg.src = `http://openweathermap.org/img/wn/${icon}@2x.png`;

    iconImg.alt = "weather-icon";
```



```
weatherIcon.appendChild(iconImg);

textError.classList.remove("wait", "error");

container.classList.add("active");

console.log(info);

console.log(timezone);

function displayTime() {

    let utcHours = timezone / 360 / 10;

    const d = new Date();

    const localTime = d.getTime();

    const localOffset = d.getTimezoneOffset() * 60000;

    const utc = localTime + localOffset;

    const getTime = utc + 3600000 * utcHours;

    const getTimeNow = new Date(getTime);

    const hrs = getTimeNow.getHours();

    const min = getTimeNow.getMinutes();

    const month = getTimeNow.getMonth();

    const date = getTimeNow.getDate();

    const day = getTimeNow.getDay();

    document.getElementById("day").innerHTML = days[day];

    document.getElementById("month").innerHTML = months[month];

    document.getElementById("date-today").innerHTML = date;

    if (min < 10) {

        document.getElementById("minutes").innerHTML = "0" + min;

    } else {
```

```
    document.getElementById("minutes").innerHTML = min;
}

if (hrs < 10) {
    document.getElementById("hours").innerHTML = "0" + hrs;
} else {
    document.getElementById("hours").innerHTML = hrs;
}
}

setInterval(displyTime, 100);
}

}
```