

ABSTRACT

Weather is the state of the atmosphere at a given place and time regarding heat, wind, and rain. Of all the geophysical phenomena weather is the most significant one that influences us. Weather can vary greatly and largely depends on climate, seasons, and various other factors. The chief goal of this work is to get the weather forecast of any city throughout the world through an application. This paper aims at creating a web application using JavaScript in the form of JSON (JavaScript Object Notation).

2. INTRODUCTION

Weather forecasting is the scientific prediction of the state of atmospheric conditions such as temperature, humidity, dew point, rainfall, and wind speed based on reliable data. Barometers, radar, and thermometers are used to collect data for weather prediction. External factors such as current weather conditions, data of previous weather patterns, tracking the motion of air and clouds in the sky, finding, and verifying changes in air pressure are essential in forecasting weather.

The rapid development of technologies like the Internet of Things, Wireless Sensor Network, and Cloud Computing has helped weather forecasting enter the era of Big Data. API (Application Program Interface) is one of the technological advancement whose information shared is 90% accurate which is reliable.

3. ANALYSIS

3.1 Literature Survey:

Weather forecasting became an important tool for aviation during the 1920s and '30s. Many oceangoing shipping vessels as well as military ships use optimum ship routing forecasts to plan their routes to minimize the loss of time, potential damage, and fuel consumption in heavy seas. Any observer who has learned the nature's signs in the sky can interpret the appearance of the sky, can interpret the appearance of the sky, the wind, and other local effects and "foretell the weather".

Weather forecasting has been a persistent task to meteorologists, scientists, farmers, and many other people from other sections of society. People have attempted to predict the weather informally for millennia and formally since the 19th century. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, teleconnections, knowledge of model performance, and knowledge of model biases.

Sometimes due to chaotic nature of atmosphere, the computational cost forecast the weather could go in vain. Some also include while measuring initially or at the time of forecast. The theoretical models of 20th- and 21st-century atmospheric scientists and meteorologists helped for the betterment in applications. The so-called synoptic weather map came to be the principal tool of 19th-century meteorologists. This is used today in weather stations and on television weather reports all over the world. All can happen only through a comprehensive weather forecast.

3.2 Problem Statement:

Our Problem statement is “WEATHER APPLICATION INFO. USING FRONT END WEB DEVELOPMENT”. We decided to make use of two Open Weather API’s namely, Current Weather Data and 5 Day Forecast.

Current Weather Data API – We collect and process weather data from different sources such as global and local weather models, satellites, radars, and a vast network of weather stations. Data is available in JSON, XML, or HTML format. We can call weather data through city name, geographical locations, Zip codes and City ID. It retrieves the current day’s weather-related information.

5 Day Forecast API – This API retrieves the weather conditions of 5-days counting from the current day of a particular location. Data is available in JSON, XML, or HTML format. We can call weather data through city name, geographical locations, Zip codes and City ID.

3.3 Proposed System:

Main stages of the proposed system:

- ❖ **Wireframe creation** – For every web application, we tend to include the components in single screen, including its position. We design the website such that its functionalities are available and to every component has its own behaviour. It gives the client, developer, and designer an opportunity to walk through the structure of the website without getting side-tracked by design elements such as colours and images.
- ❖ **Website structuring** – HTML (Hypertext Markup Language) is a universal language for all website, it gives the tags and options to create a blueprint(prototype) after wireframing. Accordingly, we apply CSS and JS.
- ❖ **Website styling** - CSS (Cascading Style Sheets) is responsible to allocate the colour and design types to every website. Through CSS we can create an attractive web design, it also includes the font of different components, the positioning of component, the animations and other effects added to the components.
- ❖ **Website functionality** – JS(JavaScript) adds the interactivity of the website, it includes of what operations to be done by the components. The World's best programming language includes both the front end and the back-end variants. It allows us to add dynamic behaviour to the webpage and add special effects to the webpage. On websites, it is mainly used for validation purposes. JavaScript helps us to execute complex actions and enables the interaction of websites with visitors.
- ❖ **Testing the Application** – After developing the applications test cases are run against by comparing the weather information of what we already collected of some selected cities and the weather information displayed by this web application

3.4 Software Requirements:

The following software was used for the implementation of the system:

1. Any text editor for HTML, CSS, and JavaScript.
2. Debugging tools pesticide and json viewer to make debugging simpler.
3. Any search engine for web application.

3.5 Hardware Requirements:

The following hardware was used for the implementation of the system:

4 GB RAM

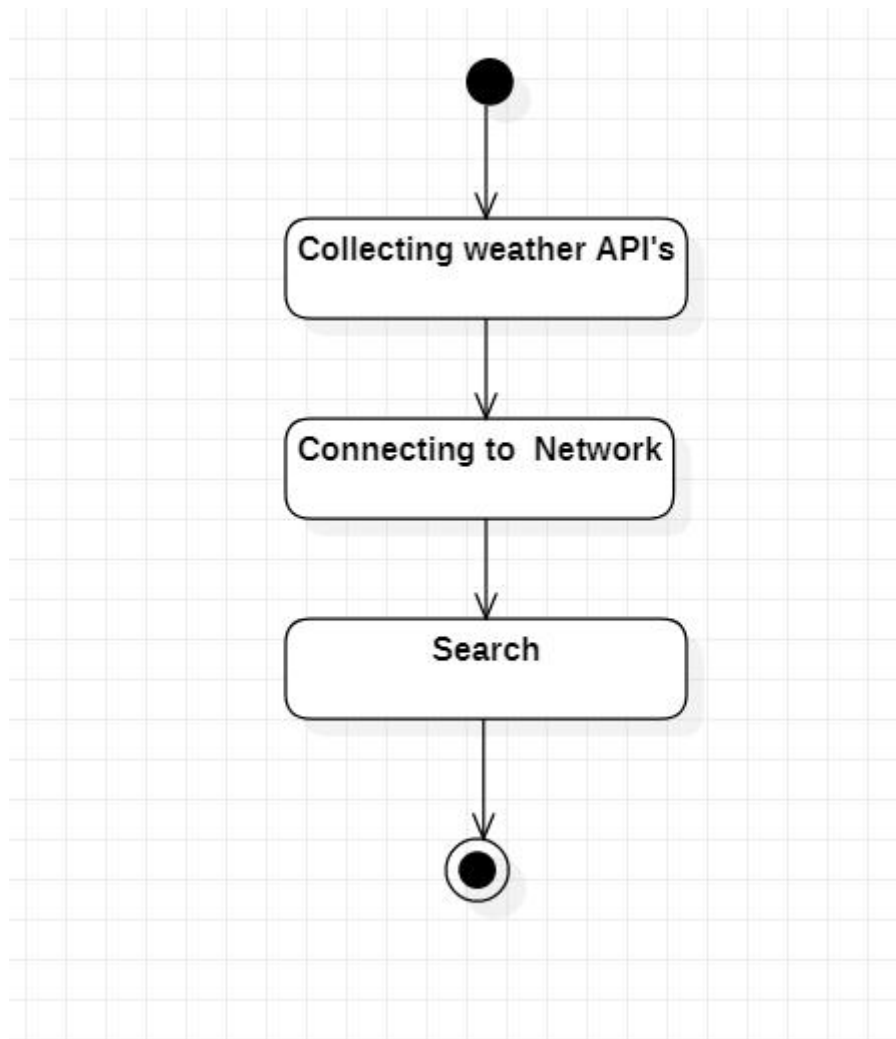
10GB HDD

Intel 1.66 GHz Processor Pentium 4

4. DESIGNING

System design is used for understanding the construction of system. We have explained the flow of our system and the software used in the system in this section.

4.1 Flow of the system:



5. IMPLEMENTATION

5.1 Implementation Steps:

❖ Collecting API's:

To implement the application, we need to collect the respective weather API's, each API has its own functionality. Thus, to get the current weather and the upcoming forecast we have selected the current Weather API and 5-day forecast Weather API.

Current Weather Data

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations
- JSON, XML, and HTML formats
- Included in both free and paid subscriptions

Hourly Forecast 4 days

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- JSON and XML formats
- Included in the Developer, Professional and Enterprise subscription plans

Daily Forecast 16 days

[API doc](#) [Subscribe](#)

- 16 days forecast is available for any location on the globe
- 1-day step for 16 days
- JSON and XML formats
- Included in all paid subscription plans

Climatic Forecast 30 days

[API doc](#) [Subscribe](#)

- Forecast weather data for 30 days
- JSON format
- Included in the Developer, Professional and Enterprise subscription plans

❖ Connecting to Network:

For Web Application which is composed of various files like: HTML, CSS, JS, and many others. The Web pages in real-time applications also utilises the in-built related files from the Operating System which requires an active Internet Connection to display the relevant data and retrieve some useful API modules to show the information.

❖ Search:

To display required weather info. from certain places the place/city has to entered as a input to this application which displays the necessary details to forecast over a particular place.

6.CODE SNIPPETS

1. APP COMPONENT 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://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700&display=swap" rel="stylesheet">
```

```
<link href="C:\Users\Akhi\Desktop\3-2 internship\style_currentday.css" rel="stylesheet">
```

```
<link rel="icon" href="C:\Users\Akhi\Desktop\Web Dev bootcamp\favicon.png" sizes="32x32">
```

```
</head>
```

```
<body>
```

```
<h1 class="title">Weather Application info</h1>
```

```
<div class="app-main">

<div class="searchInputBox">

<input type="text" id="input-box" class="input-box" placeholder="Enter city name..."
autocomplete="off">


</div>

</div>


<main class="main-container">

<div class="location-and-date">


<h1 class="location-and-date__location" id="location">London, UK</h1>

<div id="day_date">Sunday 4th August</div>

</div>


<br>


<div class = "box">
```

```
<div class="current-temperature">
```

```
<div class="current-temperature__icon-container">
```

```

```

```
</div>
```

```
<div class="current-temperature__content-container">
```

```
<div class="current-temperature__value" id="current_temperature">21&deg;</div>
```

```
<div class="current-temperature__summary" id="weather_description">Mostly Sunny</div>
```

```
</div>
```

```
</div>
```

```
<div class="current-stats">
```

```
<div class = "column1">
```

```
<div class="current-stats__value" id="weather_high_value">23&deg;</div>
```

```
<div class="current-stats__label" >High</div>
```

```
<div class="current-stats__value" id="weather_low_value">14&deg;</div>
```

```
<div class="current-stats__label">Low</div>
```

</div>

<div class = "column2">

<div class="current-stats__value" id = "wind_value">7mph</div>

<div class="current-stats__label">Wind</div>

<div class="current-stats__value">0%</div>

<div class="current-stats__label">Rain</div>

</div>

<div>

<div class="current-stats__value" id="sunrise_value">05:27</div>

<div class="current-stats__label">Sunrise</div>

<div class="current-stats__value" id="sunset_value">20:57</div>

<div class="current-stats__label">Sunset</div>

</div>

</div>

</div>

<script src="C:\Users\Akhil\Desktop\3-2 internship\app_current day.js"></script>

<script src="<C:\Users\Akhil\Desktop\main internship\moment-develop\moment.js>"></script>

</main>

<main id="end-container">

<link rel="stylesheet" href="C:\Users\Akhil\Desktop\3-2 internship\style_5dayforecast.css">

<div id = "weatherContainer">

<div id="iconsContainer">

<div class = "icons">

<p class="weather" id="day1"></p>

<div class="image1"></div>

<div class = "result" id="result1">

<p class="maxValues" id="day1Max">Loading...</p>

<p class="minValues" id="day1Min">Loading...</p>

</div>

</div>

<div class = "icons">

<p class="weather" id="day2"></p>

<div class="image2"></div>

<div class = "result" id="result1">

<p class="maxValues" id="day2Max">Loading...</p>

<p class="minValues" id="day2Min">Loading...</p>

</div>

</div>

<div class = "icons">

<p class="weather" id="day3"></p>

<div class="image3"></div>

<div class = "result" id="result1">

<p class="maxValues" id="day3Max">Loading...</p>

<p class="minValues" id="day3Min">Loading...</p>

</div>

</div>

```
<div class = "icons">
```

```
<p class="weather" id="day4"></p>
```

```
<div class="image4"></div>
```

```
<br>
```

```
<div class = "result" id="result1">
```

```
<p class="maxValues" id="day4Max">Loading...</p>
```

```
<p class="minValues" id="day4Min">Loading...</p>
```

```
</div>
```

```
</div>
```

```
<div class = "icons">
```

```
<p class="weather" id="day5"></p>
```

```
<div class="image5"></div>
```

```
<br>
```

```
<div class = "result" id="result1">
```


<p class="maxValues" id="day5Max">Loading...</p>

<p class="minValues" id="day5Min">Loading...</p>

</div>

</div>

</div>

</div>

<script src="C:\Users\Akhil\Desktop\3-2 internship\app_5 day forecast.js"></script>

</main>

</body>

</html>

2.APP COMPONENT CSS:

2.1 CURRENT WEATHER CSS:

```
* {  
  
-webkit-font-smoothing: antialiased;  
  
-moz-osx-font-smoothing: grayscale;  
  
box-sizing: border-box;  
  
}
```

```
.title{  
  
text-align: center;  
  
}
```

```
.app-main {  
  
width: fit-content;  
  
margin: 20px auto;  
  
background-color: linear-gradient(to bottom, rgb(43,50,178) 0%, rgb(20,136,204) 100%);  
  
padding: 14px;  
  
text-align: center;  
  
}
```

```
.app-main > * {  
  
margin-bottom: 6px;
```

```
}
```

```
.input-box {  
width: 70%;  
background-color: rgba(255,255,255,0.6);  
border: none;  
outline: none;  
margin-bottom: 10px;  
color: #582233;  
font-size: 0.93rem;  
height: 40px;  
border-radius: 5px;  
padding: 0 10px;  
text-align: center;  
}
```

```
body {  
margin: 12px;  
font-size: 100%;  
font-family: 'Open Sans', sans-serif;  
color: white;  
height: 100%;  
background: linear-gradient(to bottom, rgb(43,50,178) 0%, rgb(20,136,204) 100%);  
background-repeat: no-repeat;  
background-attachment: fixed;
```

```
}
```

```
.main-container {  
  
display: flex;  
  
flex-wrap: wrap;  
  
width: 880px;  
  
margin: 8px auto;  
  
font-size: 1.1em;  
  
}
```

```
/* location-and-date */  
  
.location-and-date {  
  
width: 100%;  
  
margin-top: 1em;  
  
}
```

```
.location-and-date__location {  
  
margin: 0;  
  
font-size: 2em;  
  
font-weight: 600;  
  
}
```

```
.box{  
  
display: flex;  
  
}
```

```
/* current-temperature */
```

```
.box .current-stats {  
margin-left: 6em;  
}
```

```
.current-temperature {  
display: flex;  
margin-top: 0.25em;  
width: 140.800;  
}
```

```
.current-temperature__icon-container {  
flex-grow: 1.25;  
text-align: center;  
}
```

```
.current-temperature__content-container {  
margin-left: 15%;  
flex-grow: 1;  
text-align: center;  
}
```

```
.current-temperature__icon {  
width: 10.5em;  
}
```

```
.current-temperature__value {  
font-size: 5.25em;  
font-weight: 300;  
}
```

```
.current-temperature__summary {  
margin-top: -0.5em;  
margin-left: -0.6em;  
text-align: center;  
font-size: 1.125em;  
}
```

```
/* current-stats */  
.current-stats {  
display: flex;  
justify-content: space-around;  
padding-bottom: 1em;  
padding-left: 4em;  
width: 100%;  
border-top: 1px solid rgba(255,255,255,0.5);
```

```
border-bottom: 1px solid rgba(255,255,255,0.5);
```

```
}
```

```
.current-stats {
```

```
margin-left: 1em;
```

```
margin-bottom: 1em;
```

```
padding-bottom: 0;
```

```
border-top: none;
```

```
border-bottom: none;
```

```
font-size: 22px;
```

```
border-left: 1px solid rgba(255,255,255,0.5);
```

```
}
```

```
.current-stats column1 {
```

```
margin-left: 15%;
```

```
}
```

```
.current-stats__value {
```

```
margin-top: 0.87em;
```

```
margin-right: 50px;
```

```
font-size: 1.44em;
```

```
}
```

```
.current-stats__label {
```

```
color: rgba(255,255,255,0.6);
```

```
}
```

```
.current-temperature,
```

```
.current-stats {
```

```
width: 50%;
```

```
}
```

```
.column2{
```

```
margin-left: 1.6em;
```

```
}
```

```
/* weather-by-hour */
```

```
.weather-by-hour {
```

```
display: none;
```

```
width: 100%;
```

```
}
```

```
.weather-by-hour {
```

```
display: block;
```

```
}
```

```
.weather-by-hour__container {
```

```
display: flex;
```

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

```
}
```



```
.weather-by-hour__heading,  
.next-5-days__heading {  
color: rgba(255,255,255,0.8);  
font-size: 1em;  
font-weight: normal;  
}
```

```
.weather-by-hour__heading,  
.next-5-days__heading {  
font-size: 1.125em;  
}
```

```
.weather-by-hour__item {  
padding: 0.8em 0;  
width: 13%;  
border-radius: 5px;  
background-color: rgba(0,0,0,0.15);  
font-size: 1.125em;  
text-align: center;  
}
```

```
.weather-by-hour__item {  
width: 6.05em;  
}
```

```
.weather-by-hour__hour {
```

```
margin-bottom: 0.5em;
```

```
}
```

```
/* next-5-days */
```

```
.next-5-days {
```

```
width: 100%;
```

```
margin-top: 1em;
```

```
}
```

```
.next-5-days__container {
```

```
display: flex;
```

```
flex-wrap: wrap;
```

```
}
```

```
.next-5-days__row {
```

```
display: flex;
```

```
flex-wrap: wrap;
```

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

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

```
margin-bottom: 0.3em;
```

```
padding: 0.8em 0;
```

```
width: 100%;
```

```
border-radius: 5px;
```

```
background-color: rgba(0,0,0,0.20);
```

```
font-size: 1.19em;
```

```
text-align: center;
```

```
}
```

```
.next-5-days__label {
```

```
color: rgba(255,255,255,0.6);
```

```
font-size: 0.83em;
```

```
}
```

```
.next-5-days__date,
```

```
.next-5-days__high,
```

```
.next-5-days__low,
```

```
.next-5-days__icon,
```

```
.next-5-days__rain,
```

```
.next-5-days__wind {
```

```
width: 33.33333%;
```

```
font-size: 0.95em;
```

```
}
```

```
.next-5-days__date,
```

```
.next-5-days__high,
```

```
.next-5-days__low {
```

```
margin-bottom: 0.6em;
```

```
}
```

```
.next-5-days__date,
```

```
.next-5-days__high,
```

```
.next-5-days__low,
```

```
.next-5-days__icon,
```

```
.next-5-days__rain,

.next-5-days__wind {

width: 16.666666%;

margin-bottom: initial;

}

.next-5-days__date {

order: -2;

}

.next-5-days__icon {

order: -1;

}
```

2.2 5-DAY FORECAST :

```
.title{

font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;

font-size: 7vh;

}
```

```
.imgClass1 {

height: 40px;

}
```

```
#cityInput, #cityName, #inputContainer,button{

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

color: white;
```

```
margin: 3px;

padding: 11px;

text-align: center;

font-size: 20px;

font-weight: bold;

}
```

```
#weatherwidget-io{

width: 30%;

}
```

```
#weatherContainer{

margin-top: 13%;

margin-left: 2%;

position: absolute;

top: 70%;

left: 50%;

margin-right: -50%;

transform: translate(-50%, -50%);

height: 300px;

width: 1230px;

border: 5px solid black;

border-radius: 4px;

}
```

```
#iconsContainer{  
  
position: absolute;  
  
top: 50%;  
  
left: 50%;  
  
margin-right: -50%;  
  
transform: translate(-50%, -50%);  
  
height: 270px;  
  
width: 1200px;  
  
border: 5px solid black;  
  
border-radius: 10px;  
  
}
```

```
.icons{  
  
display: inline-block;  
  
float: left;  
  
height: 230px;  
  
width: 190px;  
  
margin: 22px;  
  
border: 5px solid rgb(48, 47, 47);  
  
border-radius: 10px;  
  
}
```

```
.image{  
  
height: 140px;  
  
width: 100%;  
  
}  
  
.imgClass1,.imgClass2,.imgClass3,.imgClass4,.imgClass5{  
  
  
  
justify-content: center;  
  
margin-top: 14px;  
  
margin-left: 50px;  
  
height: 80px;  
  
width: 44%;  
  
}  
  
.result{  
  
display: flex;  
  
}  
  
.weather{  
  
margin: 5px;  
  
/* background-color: rgb(106, 215, 255); */  
  
border-radius: 5px;  
  
text-align: center;  
  
font-weight: bold;  
  
}
```

```
.minValues{  
text-align: center;  
display: inline-block;  
border-radius: 5px;  
height: 25px;  
width: 85px;  
margin: 0;  
margin-bottom: 5px;  
margin-left: 5px;  
}
```

```
.maxValues{  
text-align: center;  
display: inline-block;  
border-radius: 5px;  
height: 25px;  
width: 85px;  
margin: 0;  
margin-bottom: 5px;  
margin-left: 5px;}
```


3.APP COMPONENT CURRENT WEATHER JAVASCRIPT :

```
const weatherApi = {
key: "bab281d79e5f1e9755a68d754cc313e7",
baseUrl: "https://api.openweathermap.org/data/2.5/weather",
}

// Event Listener Function on keypress - here event calls are made

var eal = document.getElementById('input-box');
if(eal){

eal.addEventListener('keypress', (event) => {

if(event.keyCode === 13) { //keyCode=13 denotes enter key
console.log(eal.value);
Weatherdata(eal.value);
document.querySelector('.main-container').style.display = "block"; //displaying searched value
}
});
}

function Weatherdata (city) {
fetch(`${weatherApi.baseUrl}?q=${city}&appid=${weatherApi.key}&units=metric`)
.then(weather => {
return weather.json();
}).then(displayWeather);
}
```

```

function displayWeather(weather) {

console.log(weather);

let place = document.getElementById('location');
place.innerHTML = `${weather.name}`, `${weather.sys.country}`;

let date = document.getElementById('day_date');
let todayDate = new Date();           //new Date() - Thu Oct 28 2021 20:35:24 GMT+0530 (India Standard
Time) {}
date.innerHTML = dateManage(todayDate);
//icon pending

let temperature = document.getElementById('current_temperature');           //getElementById denotes fetching
values from assigned id's in html file
temperature.innerHTML = `${Math.round(weather.main.temp)}&deg;C`;

let description = document.getElementById('weather_description');
description.innerHTML = `${weather.weather[0].description}`;

let maxtemp = document.getElementById('weather_high_value');
maxtemp.innerHTML = `${Math.floor(weather.main.temp_max)}`;

let mintemp = document.getElementById('weather_high_value');
mintemp.innerHTML = `${Math.ceil(weather.main.temp_min)}`;

let wind_measure = document.getElementById('wind_value');
wind_measure.innerHTML = `${Math.ceil(weather.wind.speed)}mph`;

let sunrise_time = document.getElementById('sunrise_value');

```

```
var sunval = weather.sys.sunrise;
var timestamp1 = moment.unix(sunval);

sunrise_time.innerHTML = timestamp1.format("HH:mm");
var sunset_time = document.getElementById('sunset_value');

var sunsetval = weather.sys.sunset;
var timestamp2 = moment.unix(sunsetval);

sunrise_time.innerHTML = timestamp2.format("HH:mm");

function dateManage(dateAg) {

let days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

let months = ["January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"];

//return value
let day = days[dateAg.getDay()];    //some day
let date = dateAg.getDate();        //some date
let month = months[dateAg.getMonth()]; //October
return `${day} ,${date} ${month} `;}}
```

APP COMPONENT 5-DAY FORECAST :

```
const dayweatherApi = {

  baseUrl: "https://api.openweathermap.org/data/2.5/forecast",
  key : "cc66856dc8f8bee7b60dc6af5083e432"
}

var eal1 = document.getElementById('input-box');
if(eal1){

  eal1.addEventListener('keypress', (event) => {

    if(event.keyCode === 13) { //keyCode=13 denotes enter key
      console.log(eal1.value);
      GetInfo(eal1.value);
      document.querySelector('.end-container').style.display = "block"; //displaying searched value
    }

  });
}

function GetInfo(city) {
  fetch(`${dayweatherApi.baseUrl}?q=${city}&appid=${dayweatherApi.key}&units=metric`)
    .then(weather => {
      return weather.json();
    }).then(displayWeatherforecast);
}
```

```

function displayWeatherforecast(weather)
{
console.log(weather);
for(i = 0; i<5; i++){
document.getElementById(`day${i} + 1}Max`).innerHTML = "Max: " +
Math.round(weather.list[i].main.temp_max ) + "°C";
}

for(i = 0; i<5; i++){
document.getElementById("day" + (i+1) + "Min").innerHTML = "Min: " +
Math.round(weather.list[i].main.temp_min )+ "°C";
}

for(i = 0; i<5; i++){
document.getElementById("img" + (i+1)).src = "http://openweathermap.org/img/wn/"+
weather.list[i].weather[0].icon
+".png";
}}

//Getting and displaying the text for the upcoming five days of the week
var d = new Date();
var weekday = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",];

```

```
//Function to get the correct integer for the index of the days array
function CheckDay(day){
  if(day + d.getDay() > 6){
    return day + d.getDay() - 7;
  }
  else{
    return day + d.getDay();
  }
}
for(i = 0; i<5; i++){
  document.getElementById("day" + (i+1)).innerHTML = weekday[CheckDay(i)];
}
```

7. RESULTS

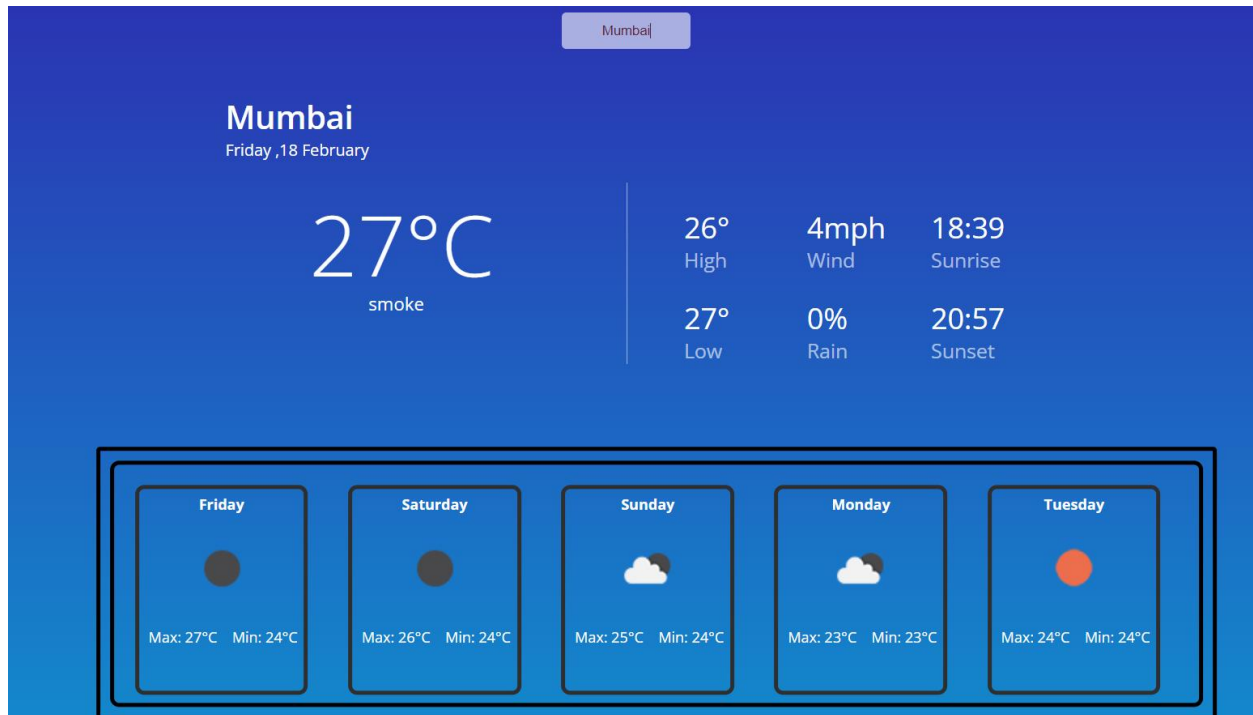


Figure 7.1: Weather Forecast of Mumbai

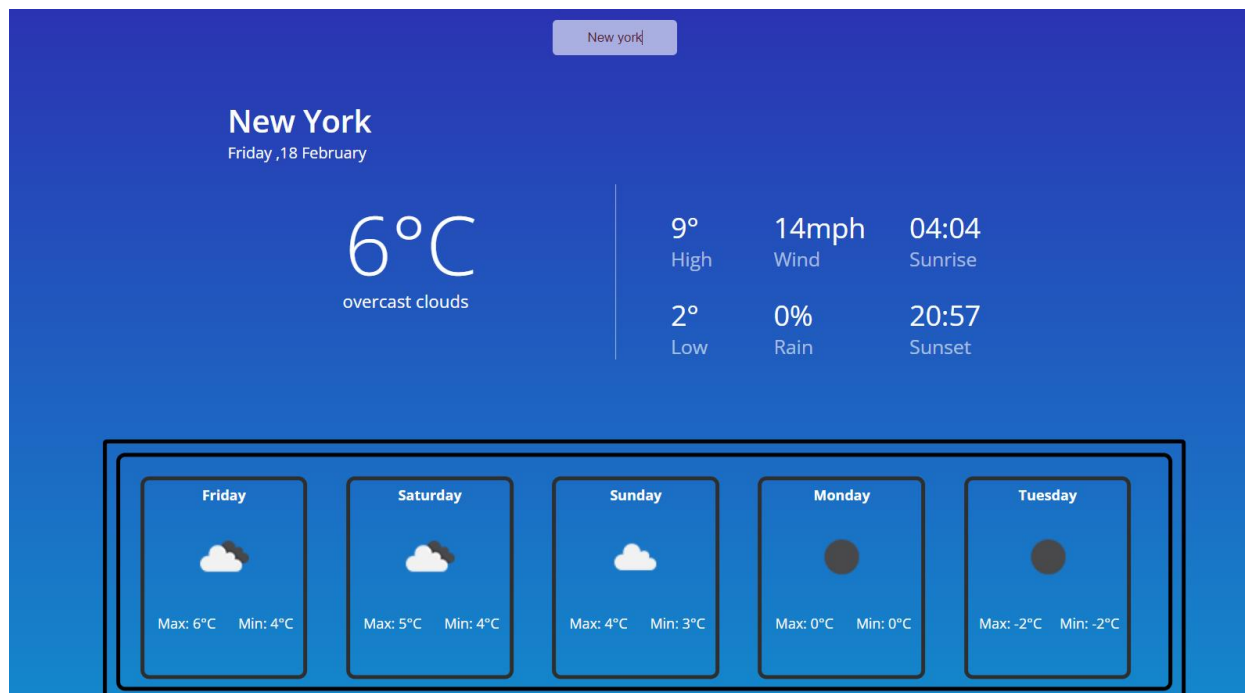


Figure 7.2: Weather Forecast of New York



Figure 7.3: Weather Forecast of Hyderabad

8. CONCLUSION

Weather plays a major role in our daily life, and without the meteorologist and forecaster we would have difficulty planning our daily activities. As we can see, the weather is not a simple subject like we may have been thinking. The study of weather phenomenon requires the use of science, math, and different types of equipment and technology and data. Even with all these equipment, data, and observation tools, the weather continues to be a topic to study because it is constantly changing. Meteorologist and forecasters predict the weather and its possible changes, but weather is still unpredictable.

9. FUTURE SCOPE

We can use NodeJS which is lightweight, can handle number of requests with high throughput. We can also use Django, a framework written in the language of python is a powerful and flexible toolkit. It also has several in-built functions which assure a high performance over other JavaScript based frameworks. Both frameworks are suitable for building scalable web applications.

10. REFERENCES

1. K. Srinivasan, A. Nema, C. -H. Huang and T. Y. Ho, "Weather Forecasting Application Using Web-Based Model-View-Whatever Framework," *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2018, pp. 1-2, doi: 10.1109/ICCE-China.2018.8448971.\
2. CSIRO. "Providing specialized weather forecasts". Retrieved May 25, 2008. April 19, 2008, at the Wayback Machine
3. The Weather Research and Forecasting Model: Overview, System Efforts, and Future Directions
<https://doi.org/10.1175/BAMS-D-15-00308.1>
4. "94.05.01: Meteorology". *teachersinstitute.yale.edu*. Retrieved January 14, 2020.
<https://teachersinstitute.yale.edu/curriculum/units/1994/5/94.05.01/2>
5. Lynch, Peter (2006). The Emergence of Numerical Weather Prediction. Cambridge University Press
https://assets.cambridge.org/97811074/14839/frontmatter/9781107414839_frontmatter.pdf