

**Certificate**

*This is to certify that Mini Project Report entitled “WEATHER FORECASTING WEBSITE” which is submitted by Bhaskar Kulshrestha in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science and Engineering of Pranveer Singh Institute of Technology, affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my/our supervision. The project embodies result of original work and studies carried out by the students themselves and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else.*

|  |  |  |
| --- | --- | --- |
| *Signature:*  *Dr. Vishal Nagar*  *Head of Department*  *CSE Department,*  *PSIT, Kanpur* |  | *Signature:*  *Garima Verma*  *Assistant Professor*  *CSE Department,*  *PSIT, Kanpur* |

Date:

**DECLARATION**

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

|  |
| --- |
| *Signature*  *Name:*  *Roll No.:*  *Date:* |

***ACKNOWLEDGEMENT***

*It gives me a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. second Year. I owe special debt of gratitude to my project supervisor* ***Garima Verma****, Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur for her constant support and guidance throughout the course of my work. Her sincerely, thoroughness and perseverance have been a constant source of inspiration for me. It is only his cognizant efforts that our endeavors have seen light of the day.*

*We also take the opportunity to acknowledge the contribution of Professor Dr. Vishal Nagar, Head, Department of Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur for his full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

|  |  |
| --- | --- |
| *Signature*  *Name:*  *Roll No.:* |  |

**ABSTRACT**

**Weather forecasting** is the application of science and technology to predict the conditions of the atmosphere for a given location and time.  Weather forecasts are made by collecting quantitative data about the current state of the atmosphere, land, and ocean and using meteorology to project how the atmosphere will change at a given place. Once calculated manually based mainly upon changes in barometric pressure, current weather conditions, and sky condition or cloud cover, weather forecasting now relies on computer-based models that take many atmospheric factors into account.

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. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the land, and the ocean, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric and related processes. Hence, forecasts become less accurate as the difference between current time and the time for which the forecast is being made (the *range* of the forecast) increases. The use of ensembles and model consensus help narrow the error and provide confidence level in the forecast.

**TABLE OF CONTENT**

**CHAPTER NO. TITLE PAGE NO.**

TITLE …………………………………………………………………………………………………. 1

CERTIFICATE …………………………………………………………………………….………. 2

DECLARATION …………………………………………………………………………………… 3

ACKNOWLEDGEMENTS ……………………………………………………………...……. 4

ABSTRACT ………………………………………………………………………………….……… 5

LIST OF TABLES ………………………………………………………………………………… 6

Main Content Indexing Sample:

* 1. Introduction 7 - 8
     1. Overview 7
     2. Motivation 7
     3. Scope of Work 8
     4. Objective 8
  2. System Requirements 9-9
     1. Hardware Requirements 9
     2. Software Requirements 9
  3. Design / Implementation 10 - 19
     1. Flowchart 10
     2. Code 11 - 19
  4. Testing / Result analysis 20
  5. Conclusion and Future enhancement 21
  6. References 22

**Chapter 1: Introduction**

1. **Overview:**

## What Is a Weather Forecasting Website?

Weather forecasting is the prediction of what the atmosphere will be like in a particular place by using technology and scientific knowledge to make weather observations. In other words, it's a way of predicting things like cloud cover, rain, snow, wind speed, and temperature before they happen.

Using the website, we can accurately measure the changes in the atmosphere and then we can respond accordingly to reduce any damage and to protect the peoples.

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. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the land, and the ocean, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric and related processes

1. **Motivation:**

The motivation to make this weather forecasting website is to make a responsive website that works on any device so that anyone can check the weather according to the location given by the user so that the anyone can plan his/her day accordingly.

Sometimes a person whose eye side is weak and cannot see the numbers clearly, so for the help the website is equipped with the icons that can change accordingly according to the weather changes and the input changes. Humidity is also an important factor in weather forecasting, so the website is equipped with the graph that shows a data of humidity according to the value of the temperature and water vapours present in the atmosphere.

**3. Scope of Work:**

* Weather Forecasting.
* Check the weather to plan your day.
* Precautionary measures can be taken before any disaster happen.
* To Predict the climate of a location and to study climate changes.
* Future climate prediction (example – prediction of rainfall).

**4. Objective:**

The objective of this project is to present to you a basic website in which you can find the weather of any place by given name of the place as an input.

According to the location given we can find the weather and related conditions like humidity accordingly and hence make plans accordingly to make our day better.

**Chapter 2: System Requirements**

1. **Hardware Requirements**

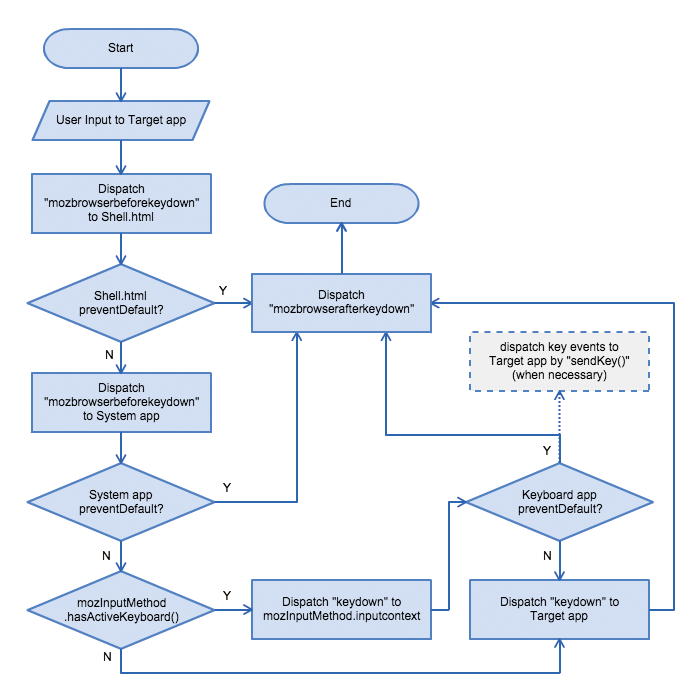
* Intel Processor (3th Gen or Above) And AMD’s 2nd Generation Ryzen 5’s and 7’s Or Higher.
* SSD Storage for Faster Review but HDD Will Also Work.
* Ram (Min 4 Gb)
* A Stable Internet Connection for Faster Accessing.

1. **Software Requirements**

* Windows 10 Operating System (Recommended).
* Chrome Browser Latest version (Version 98.0.4758.82) for secure and faster Web Accessing.
* A Text-Editor (example: Visual Studio Code, Atom)
* API (application programming interface) for connecting our website to the data server for taking the data, I am using Openweathermap API, for taking the data.

**Chapter 3: Design / Implementation**

**Flow chart for demonstrating API working:**



CODE:

**Main code / body code: (HTML)**

1. This is the head of the HTML where we have linked our style sheet (CSS) and JavaScript (JS) in the head of the HTML.

<!DOCTYPE html>

<html>

<title>Weather app</title>

<head>

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

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

</head>

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>

1. This is the code which generates the search bar and search button.

<body>

    <div class="img1">

        <form id="search-form">

            <input type="search" placeholder="Enter City Name" id="search-input" required autocomplete="off" />

            <br>

            </br>

            <button id="search-button">Search</button>

        </form>



1. This code will generate a Container in which the data is to be displayed

    <main id="app-container">

        <div id="location">

            <p>-------</p>

        </div>

        <div id="temp">

            <img id="temp-icon" src="./icons/sun.svg" alt="">

            <p><span id="temp-value">-----</span> <span id="temp-unit">&#176c</span> </p>

        </div>

        <div id="climate">

            <p>------</p>

        </div>

    </div>

    </main>

    <script src="script.js">

    </script>



1. Then this container will apply the JavaScript and will display the graph with the help of <canvas> tag of HTML.

   <div class="small">

        <canvas id="myChart" style="width:100%;max-width:700px ; background-color: white;margin: auto;margin-top: 150px ; border: 10px solid black;">

        </canvas>

1. This code will generate a graph that makes a relation between Temperature and Water vapour present in the air

       <script>

        new Chart("myChart", {

            type: "line",

            data: {

                labels: xValues,

                datasets: [{

                    fill: false,

                    lineTension: 0,

                    backgroundColor: "rgba(0,0,255,1.0)",

                    borderColor: "rgba(0,0,255,0.1)",

                    data: yValues

                }]

            },

            options: {

                legend: { display: false },

                scales: {

                    yAxes: [{ ticks: { min: 6, max: 16 } }],

                }

            }

        });

        </script>

    <div class="labcont">

        <div class="labels">

            <p>Y-axis : Water vapour in Air (grams H2O per KG of Air)</p>

            <p>X-axis : Temperature (in degree C)</p>

        </div>

    </div>

</div>

    <footer class="footer">

        <p>Bhaskar Kulshrestha</p>

        <p>B.Tech second year </p>

        <p>CS - AI and ML</p>

        <p>2001641530020</p>

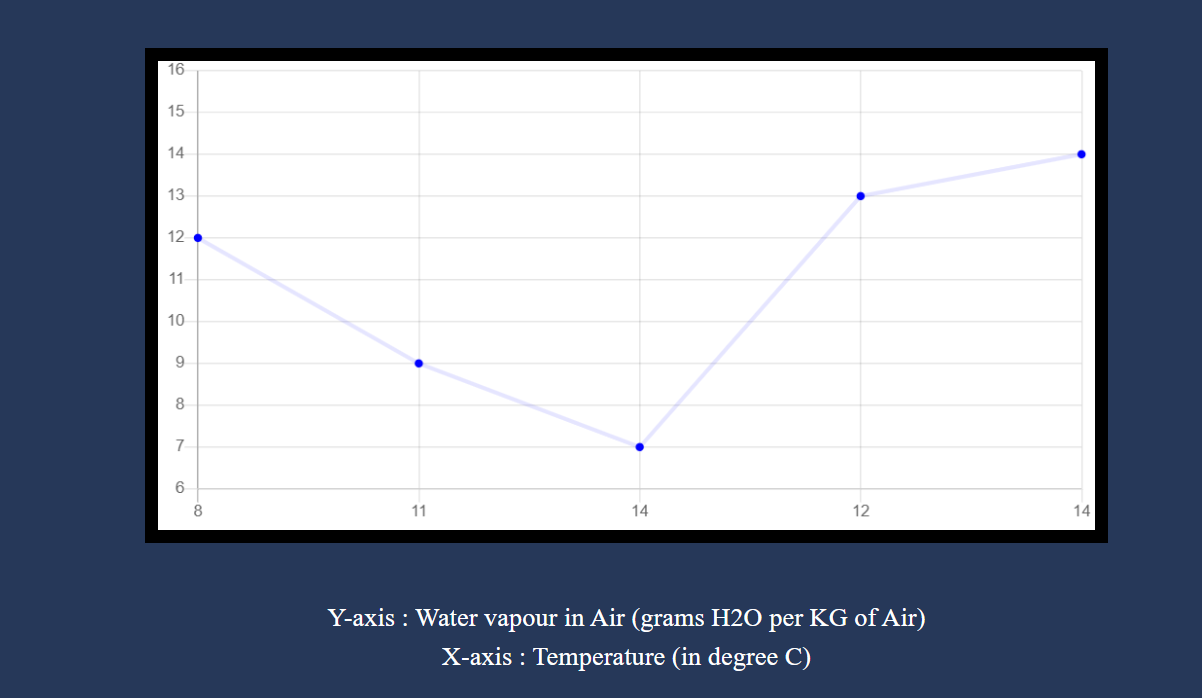
    </footer>

    <footer style="height: 100px;width: 100px;">

    </footer>

</body>

</html>



End of HTML body.

**Styling The Website (CSS):**

\*{

        box-sizing: border-box;

        margin: 0;

        padding: 0;

}

body{

    /\* height: 100px; \*/

    background-color: #263859;

    background-image: url('https://source.unsplash.com/900x500/?weather');

    background-repeat: no-repeat;

    background-size: cover;

}

#search-form

{

    display: flex;

    justify-content: center;

    align-items: center;

    margin-top: 100px;

}

#temp

{

text-align: center;

}

#temp #temp-unit

{

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

    font-size: 1.5rem;

    color: #ffffff;

}

#temp #temp-value

{

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

    font-size: 2rem;

    color: #ffffff;

}

#climate

{

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

    font-size: 2.5rem;

    color: #ffffff;

    text-align: center;

}

#location

{

    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

    font-size: 2.5rem;

    color: #ffffff;

    text-align: center;

}

#temp-icon

{

    margin: 1rem;

    width: 40%;

    height: auto;

}

main{

    padding: 20px;

    margin-top: 10px;

    background-color: #6b778d;

    border-radius: 20px;

    width: 360px;

    height: 360px;

    display: block;

    margin-left: auto;

    margin-right: auto;

    box-shadow: 10px 10px 20px 10px rgba(112, 102, 102, 0.2);

    border: 5px solid black;

}

#search-button

{

    color: #494949;

    text-transform: uppercase;

    text-decoration: none;

    background: white;

    border-radius: 5px;

    margin-left: 5px;

    padding: 11px;

    border: 2px solid #494949;

    display: inline-block;

    transition: all 0.4 ease 0s;

    border: 5px solid purple;

}

#search-button:hover{

color: #ffffff;

background: #ff6768;

border-color: #ff6768;

transition: all 0.4s ease 0s;

}

#search-input

{

    width: 260px;

    overflow: hidden;

    font-size: 20px;

    margin: 8px 0;

    padding: 8px 0;

    border-bottom: 3px solid #ff6768;

    color: black;

    outline: none;

    background-color: #ffffff;

    border: 3px solid black;

}

#search-input:hover

{

    width: 260px;

    overflow: hidden;

    font-size: 20px;

    margin: 8px 0;

    padding: 8px 0;

    border-bottom: 3px solid #ff6768;

    color: black;

    outline: none;

    background-color: antiquewhite;

}

input

{

    border: none;

}

/\* .labcont{

    background-colo;

    } \*/

.labels{

    margin-top: 40px;

    text-align: center;

    color: white;

    font-size: larger;

    line-height: 1.5;

    background-color: purple;

}

.footer{

    /\* padding: 50px; \*/

    background-color: orangered ;

    line-height: 1.5;

    color: white;

    width: 200px;

    height: 150px;

    margin: auto;

    position: fixed;

    top: 75%;

    /\* left: 10px; \*/

    text-align: center;

    padding-top: 2%;

}

.footer:hover{

    /\* padding: 50px; \*/

    background-color: blueviolet ;

    line-height: 1.5;

    color: white;

    width: 200px;

    height: 150px;

    margin: auto;

    position: fixed;

    top: 75%;

    /\* left: 10px; \*/

    text-align: center;

    padding-top: 2%;

}

**Code For JavaScript:**

let loc =document.getElementById("location");

let tempicon=document.getElementById("temp-icon");

let tempvalue=document.getElementById("temp-value");

let climate =document.getElementById("climate");

let iconfile;

const searchInput=document.getElementById("search-input");

const searchButton=document.getElementById("search-button");

searchButton.addEventListener('click', (e)=>

{

e.preventDefault();

getWeather(searchInput.value);

searchInput.value='';

});

const getWeather=async (city)=>

{

    try{

        const response= await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=dab3af44de7d24ae7ff86549334e45bd`,

            {mode: 'cors'}

        );

        const weatherData= await response.json();

        console.log(weatherData);

        const{name}=weatherData;

        const{feels\_like}=weatherData.main;

        const{id,main}=weatherData.weather[0];

        loc.textContent=name;

        climate.textContent=main;

        tempvalue.textContent=Math.round(feels\_like-273);

        if(id<300 && id>200)

        {

            tempicon.src="./icons/thunderstorm.svg"

        }

       else  if(id<400 && id>300)

        {

            tempicon.src="./icons/cloud-solid.svg"

        }

       else if(id<600&& id>500)

        {

            tempicon.src="./icons/rain.svg"

        }

       else  if(id<700 && id>600)

        {

            tempicon.src="./icons/snow.svg"

        }

       else  if(id<800 && id>700)

        {

            tempicon.src="./icons/clouds.svg"

        }

         else if(id==800)

        {

            tempicon.src="./icons/clouds-and-sun.svg"

        }

    }

    catch(error)

    {

        alert('city not found');

    }

};

window.addEventListener("load" ,()=>{

let long;

let lat;

if(navigator.geolocation)

{

    navigator.geolocation.getCurrentPosition((position)=>

    {

    long=position.coords.longitude;

    lat=position.coords.latitude;

    const proxy="https://cors-anywhere.herokuapp.com/";

        const api=`${proxy}api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${long}&appid=dab3af44de7d24ae7ff86549334e45bd     `

        fetch(api).then((response)=>{

            return response.json();

        })

        {

            tempicon.src="./icons/rain.svg"

        }

       else  if(id<700 && id>600)

        {

            tempicon.src="./icons/snow.svg"

        }

        .then (data =>

            {

                    const{name}=data;

                    const{feels\_like}=data.main;

                    const{id,main}=data.weather[0];

                    loc.textContent=name;

                    climate.textContent=main;

                    tempvalue.textContent=Math.round(feels\_like-273);

                    if(id<300 && id>200)

                    {

                        tempicon.src="./icons/thunderstorm.svg"

                    }

                   else  if(id<400 && id>300)

                    {

                        tempicon.src="./icons/cloud-solid.svg"

                    }

                   else if(id<600&& id>500)

                    {

                        tempicon.src="./icons/rain.svg"

                    }

                   else  if(id<700 && id>600)

                    {

                        tempicon.src="./icons/snow.svg"

                    }

                   else  if(id<800 && id>700)

                    {

                        tempicon.src="./icons/clouds.svg"

                    }

                     else if(id==800)

                    {

                        tempicon.src="./icons/clouds-and-sun.svg"

                    }

                    console.log(data);

            })

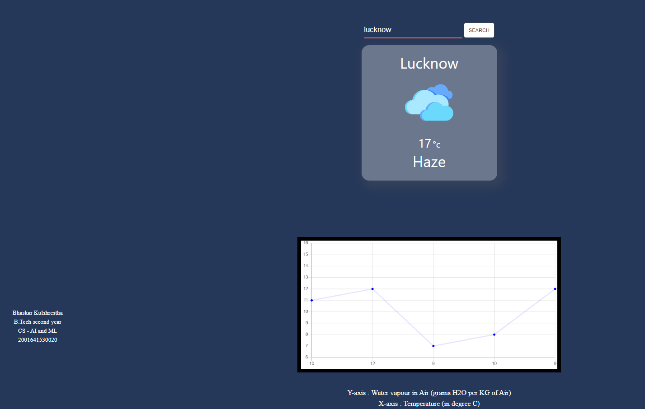
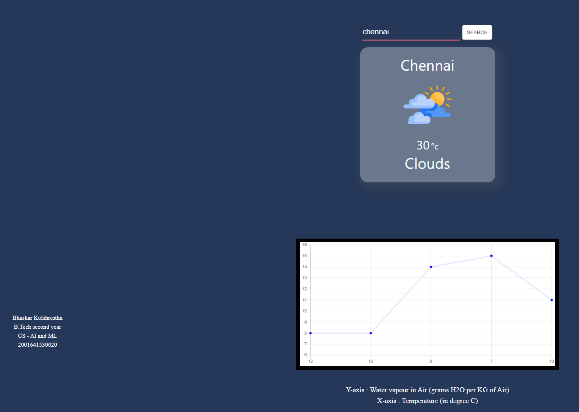
}

    )}

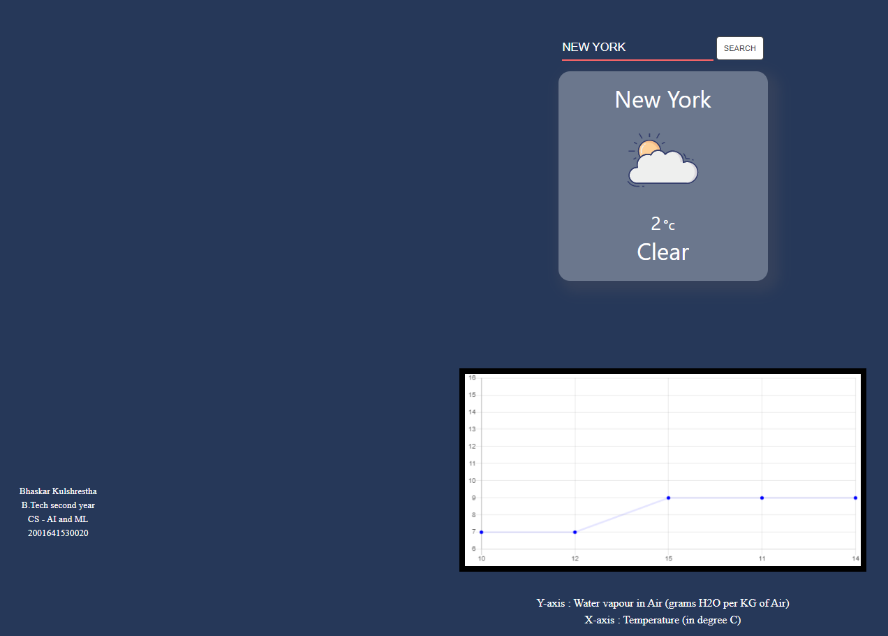
})

**Chapter 4: Testing / Result and Analysis**

**TEST 1: TEST 2:**

** **

**TEST 3:**

****

Hence, according to the results, if we give location as an input then it will give us the data of relevant temperature and a graph which describes the humidity of the location accordingly.

**Chapter 5: Conclusion and Future** Enhancement

**Conclusion:**

Due to the climate changes, there is lots of destruction happens in different parts of the world like damage caused by Typhoon, earthquake, floods etc.

With the rapid changing weather climate changes happen and this affects the whole environment hence a weather monitoring system is required to focus on the changes happened.

With this website you can find out the weather of any location in the world and hence you can work accordingly.

**Future Enhancement:**

This website can be used as fulltime working website for finding condition of the weather.

The website is fully responsive and hence can be used in any device.

This website can also be adjusted according to user’s needs

In future this website can give accurate information about, temperature, wind speed, clouds, humidity, atmospheric pressure changes etc.

**Limitations:**

1. This website does not have a feature of autocorrect that is if we enter the name of city wrong then it will give output as “incorrect – input” so we have to give input of the city carefully.
2. This website cannot give the data of very small city accurately.

**References**

* Books
* <https://openweathermap.org/current>
* <https://www.w3schools.com/css/css3_mediaqueries.asp>
* <https://mausam.imd.gov.in/>

**Signature of Mini Project Co-ordinator:**

**Signature of H.O.D.:**