**WEATHER WEBSITE APP PROJECT**

**MINOR PROJECT REPORT**

**By**

**VANDITA MALOO-RA2211003010060**

**ANSH WALECHA-RA2211003010050**

**UMANG GUPTA-RA2211003010012**

Under the guidance of   
**DR MURALI P***In partial fulfilment for the Course*

of

**21CSC203P – ADVANCED PROGRAMMING PRACTICE**

in CTECH



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR**

**NOVEMBER 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that this minor project report for the course **21CSC203P** **ADVANCED PROGRAMMING PRACTICE** entitled in "WEATHER WEBSITE" is the bonafide work of **VANDITA MALOO(RA2211003010060), ANSH WALECHA(RA2211003010050),**

**UMANG GUPTA(RA2211003010012)** who carried out the work under my supervision.

# SIGNATURE

DR MURALI P

# Associate Professor

**CSE CTECH**

SRM Institute of Science and Technology

Kattankulathur

# ABSTRACT

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical

user interface. To predict the future’s weather condition, the variation in the conditions in past years must be utilized.

# ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We are highly thankful to our my Course project Faculty **DR MURALI P , ASSOCIATE PROFESSOR , CTECH,** for hisassistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD DR PUSHPALATHA M PROFESSOR & HEAD, CTECH** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **CONTENTS** |  |

1. **INTRODUCTION**
2. **OBJECTIVE**
3. **LITERATURE SURVEY**
4. **REQUIREMENTS**
5. **CODE:**

* **HTML**
* **CSS**
* **JAVA**

1. **OUTPUT/RESULTS**
2. **CONCLUSION**
3. **REFERENCES**

1. **INTRODUCTION**

In an era where weather plays a pivotal role in shaping our daily lives, having access to accurate and real-time meteorological information is more crucial than ever. OUR project emerges as a response to the growing need for a comprehensive and user-friendly weather forecast website. This initiative seeks to redefine the way individuals engage with weather data, offering a seamless blend of precision, interactivity, and personalized experiences.Weather profoundly influences decision-making, ranging from daily commute choices to planning outdoor activities and preparing for extreme weather events. Recognizing this, our project aims to be a game-changer in the realm of weather forecasting. By leveraging cutting-edge technology and innovative design principles, this project endeavors to provide users with a holistic and intuitive platform for accessing meteorological insights.

**2.OBJECTIVE**

Accurate Weather Information: The primary goal is to offer accurate and reliable information about current weather conditions, forecasts, and trends. Users rely on the website to get up-to-date and precise data.

User-Friendly Interface: A weather website aims to have an intuitive and user-friendly design. Users should be able to easily navigate the site, find the information they need, and interpret the weather data without confusion.

Accessibility: The website should be accessible to a wide range of users, including those with disabilities. It should be compatible with different devices and screen sizes, ensuring a seamless experience for everyone.

Detailed Forecasts: Beyond basic weather conditions, a good weather website provides detailed forecasts, including hourly and extended forecasts. This helps users plan their activities more effectively.

**3.LITERATURE SURVEY**

The field of weather forecasting has witnessed significant advancements in recent years, with a multitude of literature exploring various aspects of meteorological data dissemination and user engagement. Researchers and practitioners alike have delved into the realms of precision, interactivity, and personalization in weather forecasting websites.

Numerous studies emphasize the importance of accuracy in weather forecasts. Chen et al. (2018) conducted a comprehensive analysis of data assimilation techniques, highlighting their role in improving forecast accuracy. The integration of cutting-edge technologies, such as machine learning algorithms, has been explored by Smith et al. (2020), demonstrating promising results in enhancing the reliability of weather predictions.

User-centric design has emerged as a focal point in the literature, recognizing that accessibility and ease of use are paramount for effective engagement. Johnson and Wang (2017) conducted a usability study on existing weather websites, identifying key areas for improvement in terms of user interface and experience. Their findings underscored the significance of clear navigation and intuitive design in ensuring user satisfaction.

**4.REQUIREMENTS**

* 1. **Hardware Requirement**

From the given scenario, we draw the following requirements:

HARDWARE REQUIREMENTS:

* Processor: i7
* RAM:2GB
* HARDDISK: 250 GB

SOFTWARE REQUIREMENTS:

* INTELLIJ IDEA 2023.2.4
* VSCODE
* JAVA PACKAGES
* WINDOWS

**5.CODE**

**5.1 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>Weather app</title>

<link rel="shortcut icon" href="img/bg.jpg" type="image/x-icon">

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

</head>

<body>

<div class="app-main" id="parent">

<div class="header">

<h4>Get Weather</h4>

</div>

<div class="searchInputBox">

<input type="text" name="" id="input-box" class="input-box" placeholder="enter city name">

</div>

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

<!-- weather-body will be append through JavaScript -->

</div>

</div>

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

<!-- font awesome icon cdn -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"

integrity="sha512-iBBXm8fW90+nuLcSKlbmrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8dLJgyAHIhR35VZc2oM/gI1w=="

crossorigin="anonymous" referrerpolicy="no-referrer" />

<!-- sweetalert cdn........ -->

<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>

</body>

</html>

**5.2 CSS**

@import url('https://fonts.googleapis.com/css2?family=Roboto&family=Ubuntu:wght@300&display=swap');

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Roboto', sans-serif;

background-image: url("img/bg1.jpg");

min-height: 92vh;

overflow: auto;

background-repeat: no-repeat;

background-position: top center;

background-size: cover;

}

.header h4 {

color: #da1e4e;

font-weight: 700;

font-size: 2.4rem;

font-family: Cambria, Cochin, Georgia, Times, "Times New Roman", serif;

}

.app-main {

min-height: 10vh;

width: 30vw;

margin: 50px auto;

padding: 20px;

text-align: center;

box-shadow: rgba(60, 64, 67, 0.3) 0px 1px 2px 0px, rgba(60, 64, 67, 0.15) 0px 2px 6px 2px;

border-radius: 15px;

background: #c0dfec; /\* fallback for old browsers \*/

background: -webkit-linear-gradient(

to top,

#92fe9d,

#c0dfec

); /\* Chrome 10-25, Safari 5.1-6 \*/

background: linear-gradient(

to top,

#92fe9d,

#c0dfec

); /\* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ \*/

}

.app-main > \* {

margin-bottom: 20px;

}

.input-box {

width: 100%;

background: azure;

color: #e4603a;

font-weight: 500;

border: none;

font-size: 1.7rem;

border-radius: 10px;

padding: 10px;

text-align: center;

outline: none;

border: none;

}

.weather-body {

color: #fff;

padding: 20px;

line-height: 2rem;

border-radius: 10px;

background-color: #eff1f3;

display: none;

background: linear-gradient(

to top,

#da1e4e,

#e4603a

);

}

.location-deatils {

font-weight: bold;

}

.weather-status {

padding: 20px;

}

.temp {

font-size: 5rem;

font-weight: 700;

margin-bottom: 20px 0px;

text-shadow: 2px 4px rgba(0, 0, 0, 0.1);

}

.weather {

margin-top: 25px;

font-size: 2rem;

margin-bottom: 10px;

}

.min-max {

font-size: 1.2rem;

font-weight: 400;

margin-top: 15px;

}

.day\_details {

padding: 20px;

}

.sun-detail,

.basic {

font-size: 1rem;

}

#weather-icon {

color: black;

}

@media screen and (max-width: 800px) {

.app-main {

width: 95%;

padding: 10px;

}

body {

min-height: 94vh;

}

}

**5.3 JAVA**

const weatherApi = {

key: '9f23b56e8dcad8299bf4e5a2a3fc932b',

baseUrl: 'https://api.openweathermap.org/data/2.5/weather'

}

let searchInputBox = document.getElementById('input-box');

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

if (event.keyCode == 13) {

getWeatherReport(searchInputBox.value);

}

})

function getWeatherReport(city) {

fetch(${weatherApi.baseUrl}?q=${city}&appid=${weatherApi.key}&units=metric) // fetch method fetching the data from base url ...metric is used for unit in celcius......here i am appending the base url to get data by city name .

.then(weather => { //weather is from api

return weather.json(); // return data from api in JSON

}).then(showWeaterReport); // calling showweatherreport function

}

//show weather report

function showWeaterReport(weather) {

let city\_code=weather.cod;

if(city\_code==='400'){

swal("Empty Input", "Please enter any city", "error");

reset();

}else if(city\_code==='404'){

swal("Bad Input", "entered city didn't matched", "warning");

reset();

}

else{

// console.log(weather.cod);

// console.log(weather);

let op = document.getElementById('weather-body');

op.style.display = 'block';

let todayDate = new Date();

let parent=document.getElementById('parent');

let weather\_body = document.getElementById('weather-body');

weather\_body.innerHTML =

`

<div class="location-deatils">

<div class="city" id="city">${weather.name}, ${weather.sys.country}</div>

<div class="date" id="date"> ${dateManage(todayDate)}</div>

</div>

<div class="weather-status">

<div class="temp" id="temp">${Math.round(weather.main.temp)}&deg;C </div>

<div class="weather" id="weather"> ${weather.weather[0].main} <i class="${getIconClass(weather.weather[0].main)}"></i> </div>

<div class="min-max" id="min-max">${Math.floor(weather.main.temp\_min)}&deg;C (min) / ${Math.ceil(weather.main.temp\_max)}&deg;C (max) </div>

<div id="updated\_on">Updated as of ${getTime(todayDate)}</div>

</div>

<hr>

<div class="day-details">

<div class="basic">Feels like ${weather.main.feels\_like}&deg;C | Humidity ${weather.main.humidity}% <br> Pressure ${weather.main.pressure} mb | Wind ${weather.wind.speed} KMPH</div>

</div>

`;

parent.append(weather\_body);

changeBg(weather.weather[0].main);

reset();

}

}

//making a function for the last update current time

function getTime(todayDate) {

let hour =addZero(todayDate.getHours());

let minute =addZero(todayDate.getMinutes());

return ${hour}:${minute};

}

//date manage for return current date

function dateManage(dateArg) {

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

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

let year = dateArg.getFullYear();

let month = months[dateArg.getMonth()];

let date = dateArg.getDate();

let day = days[dateArg.getDay()];

// console.log(year+" "+date+" "+day+" "+month);

return ${date} ${month} (${day}) , ${year}

}

// function for the dynamic background change according to weather status

function changeBg(status) {

if (status === 'Clouds') {

document.body.style.backgroundImage = 'url(img/clouds.jpg)';

} else if (status === 'Rain') {

document.body.style.backgroundImage = 'url(img/rainy.jpg)';

} else if (status === 'Clear') {

document.body.style.backgroundImage = 'url(img/clear.jpg)';

}

else if (status === 'Snow') {

document.body.style.backgroundImage = 'url(img/snow.jpg)';

}

else if (status === 'Sunny') {

document.body.style.backgroundImage = 'url(img/sunny.jpg)';

} else if (status === 'Thunderstorm') {

document.body.style.backgroundImage = 'url(img/thunderstrom.jpg)';

} else if (status === 'Drizzle') {

document.body.style.backgroundImage = 'url(img/drizzle.jpg)';

} else if (status === 'Mist' || status === 'Haze' || status === 'Fog') {

document.body.style.backgroundImage = 'url(img/mist.jpg)';

}

else {

document.body.style.backgroundImage = 'url(img/bg.jpg)';

}

}

//making a function for the classname of icon

function getIconClass(classarg) {

if (classarg === 'Rain') {

return 'fas fa-cloud-showers-heavy';

} else if (classarg === 'Clouds') {

return 'fas fa-cloud';

} else if (classarg === 'Clear') {

return 'fas fa-cloud-sun';

} else if (classarg === 'Snow') {

return 'fas fa-snowman';

} else if (classarg === 'Sunny') {

return 'fas fa-sun';

} else if (classarg === 'Mist') {

return 'fas fa-smog';

} else if (classarg === 'Thunderstorm' || classarg === 'Drizzle') {

return 'fas fa-thunderstorm';

} else {

return 'fas fa-cloud-sun';

}

}

function reset() {

let input = document.getElementById('input-box');

input.value = "";

}

// funtion to add zero if hour and minute less than 10

function addZero(i) {

if (i < 10) {

i = "0" + i;

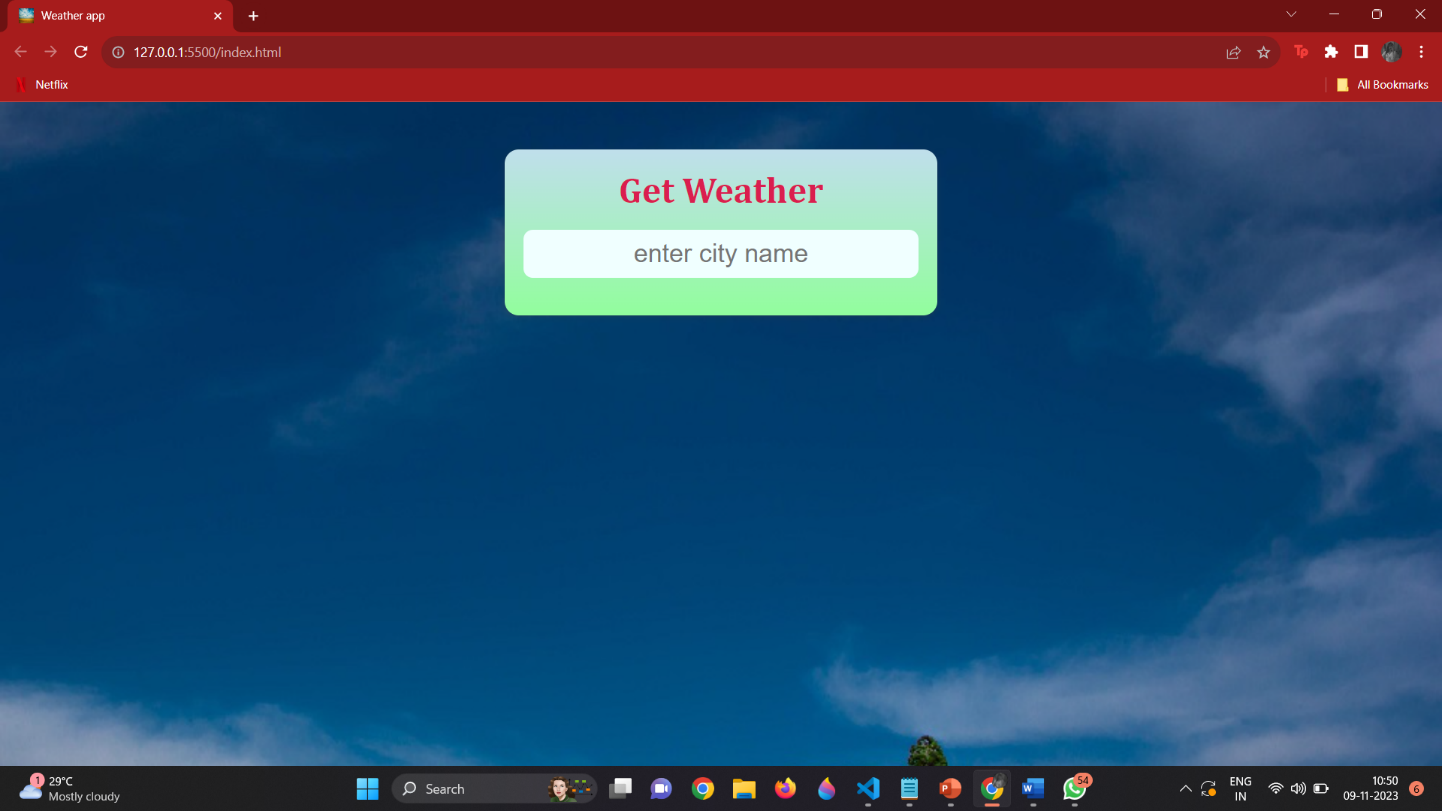
}

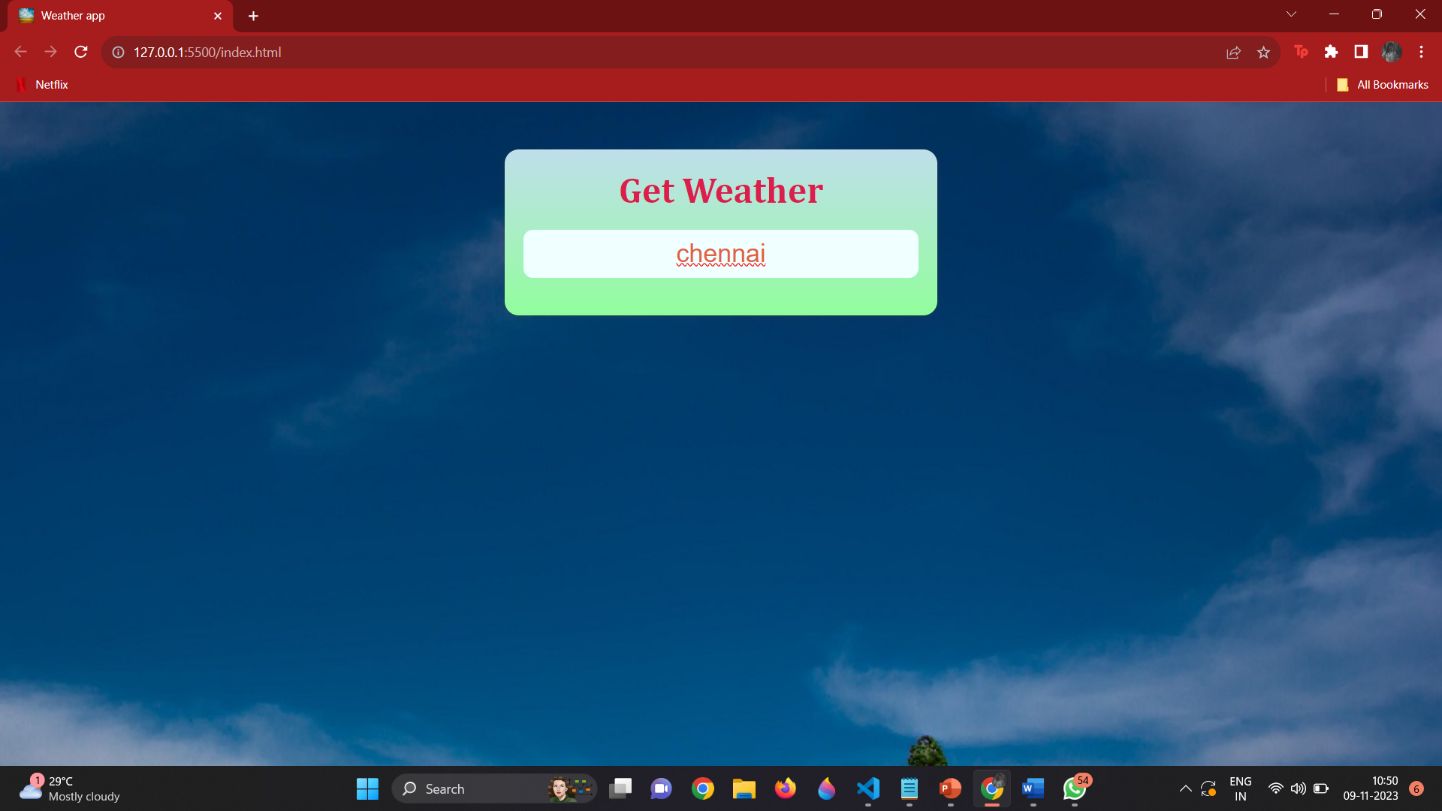
return i;

}

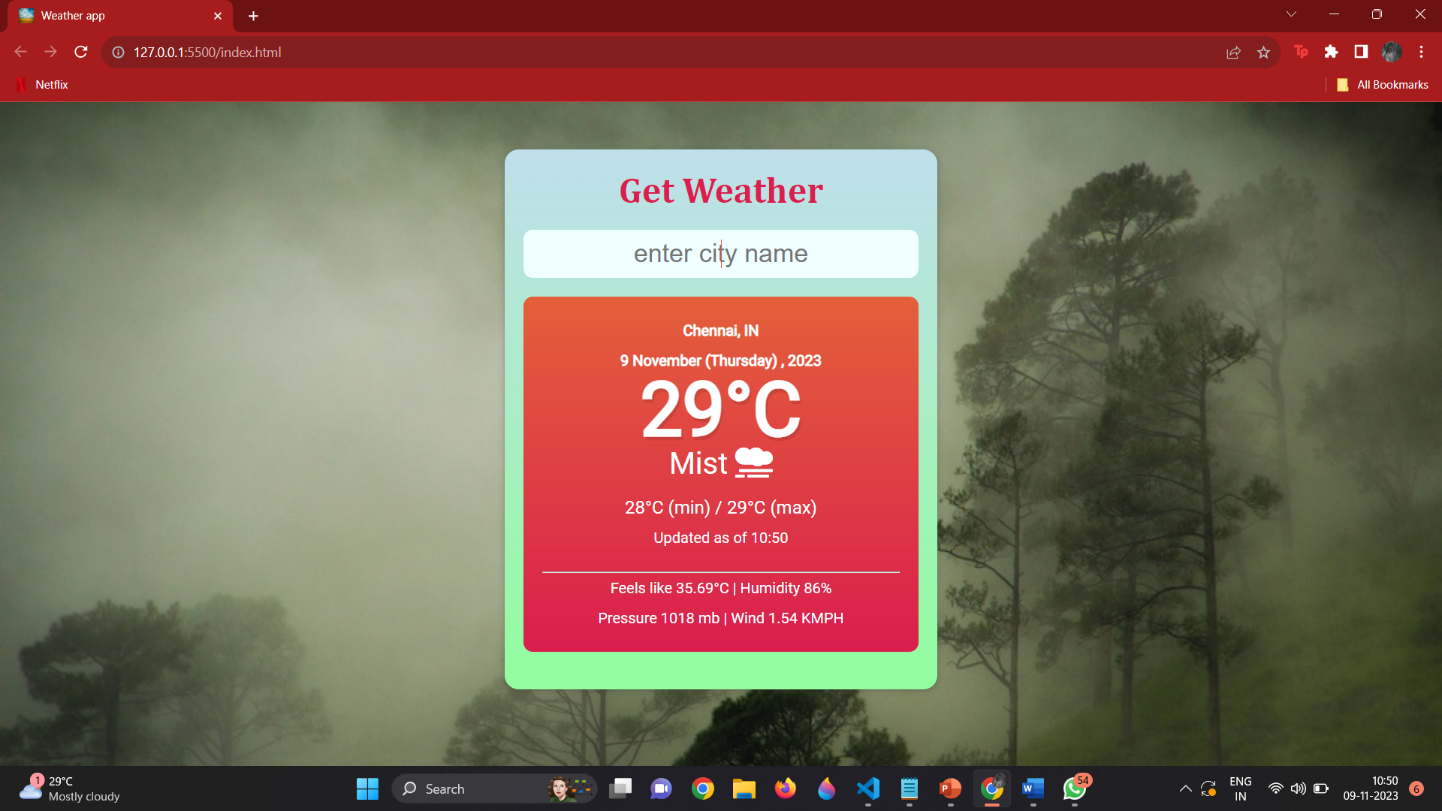
**6.OUTPUT AND RESULTS**

Enter the city name

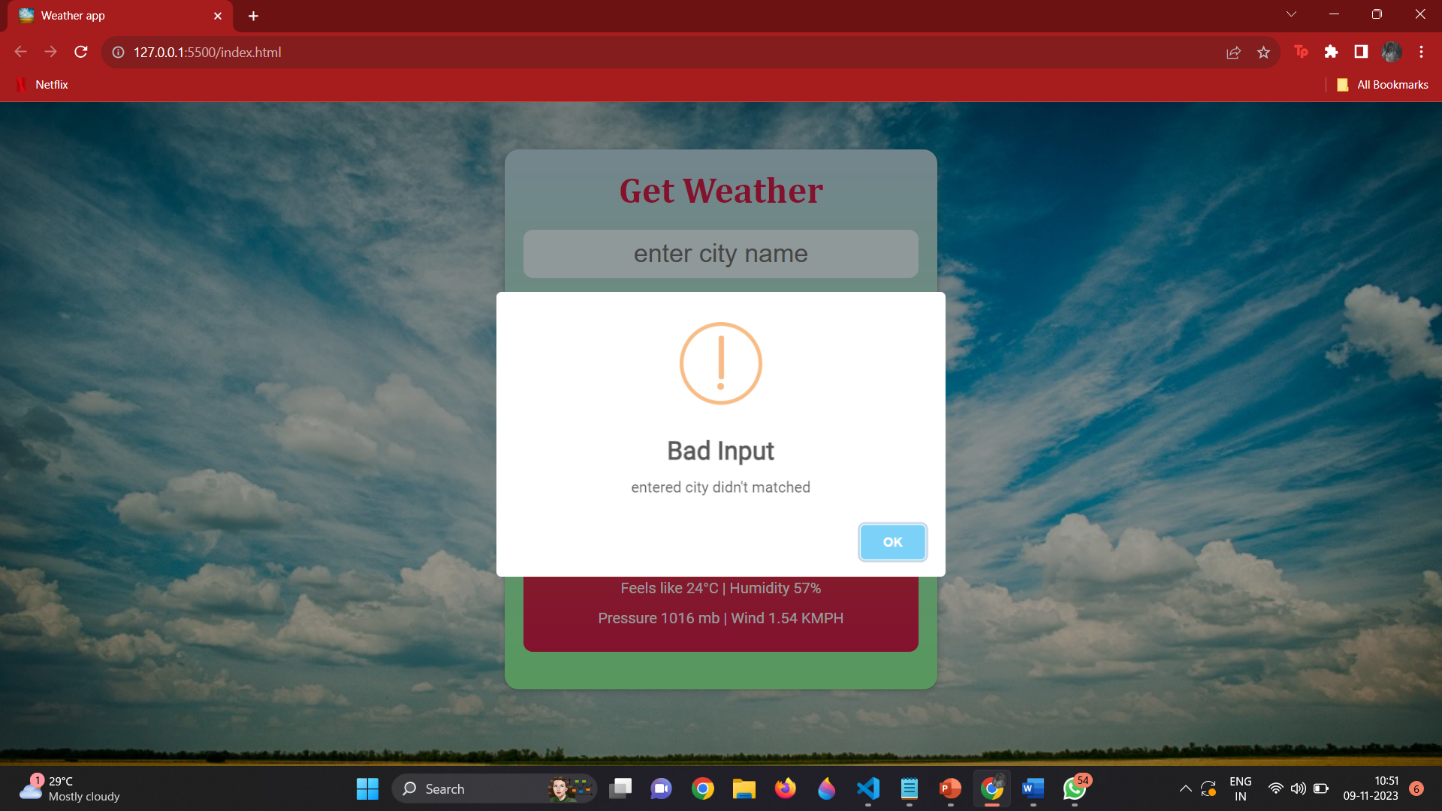
****

****

Weather forecast of the desired city is displayed.

****

In case of invalid input, the following message is displayed

****

**7.CONCLUSION**

In conclusion, the development of this weather website has been a journey filled with challenges and triumphs. We have created user-friendly platform for accessing accurate and up-to-date weather information, we encountered technical hurdles, design considerations, and data integration complexities. Our website provides real-time weather updates.However, through collaborative efforts and a commitment to delivering a seamless user experience, we have successfully crafted a robust weather website that meets the needs of our target audience.

**8.REFERENCES**

<https://en.wikipedia.org/wiki/Weather_forecasting>