

WEATHER-VERSE: AN API DRIVEN WEATHER PREDICTION WEBSITE

MINI PROJECT REPORT

Submitted by

JAGAN S	(113222031053)
----------------	-----------------------

in partial fulfillment for the award of the degree

of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE ENGINEERING**



VELAMMAL ENGINEERING COLLEGE, CHENNAI-66
(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

**VELAMMAL ENGINEERING COLLEGE,
CHENNAI – 66**



VELAMMAL
ENGINEERING COLLEGE

The Wheel of Knowledge rolls on!

BONAFIDE CERTIFICATE

Certified that this mini project report “**WEATHER-VERSE: AN API DRIVEN WEATHER PREDICTION WEBSITE**” is the bonafide work of **JAGAN S (113222031053)** who carried out at the project under my supervision.

Signature

Dr. B. MURUGESHWARI
PROFESSOR & HEAD

Department of Computer Science
and Engineering.

Velammal Engineering College ,
Ambattur, Red hills Road,
Chennai – 600 066

Signature

MS. K. C. AARTHI
ASSISTANT PROFESSOR(G-III)
SUPERVISOR

Department of Computer Science and
Engineering.

Velammal Engineering College,
Ambattur, Red hills Road,
Chennai – 600 066.

CERTIFICATE OF EVALUATION

COLLEGE NAME : VELAMMAL ENGINEERING COLLEGE
BRANCH : COMPUTER SCIENCE AND ENGINEERING
SEMESTER : IV

S.N O	Name of the students who has done the project	Title of the Project	Name of the supervisor with designation
1	JAGAN S	WEATHER-VERSE: AN API DRIVEN WEATHER PREDICTION WEBSITE	MS. K.C. AARTHI ASSISTANT PROFESSOR(G-III)

This report of project work submitted by the above students in partial fulfillment for the award of Bachelor of Computer Science and Engineering Degree in Anna University was evaluated and confirmed to be reports of the work done by the above students and then assessed.

Submitted for Internal Evaluation held on.....

Internal Examiner

External Examiner

PROGRAM OUTCOMES

PO No.	Graduate Attribute	Program Outcomes (POs)
PO1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.
PO2	Problem analysis	Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex Problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling to complex engineering activities, with an understanding of the limitations.
PO6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and Sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics	Apply ethical principles and common to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication	Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Mapping of Program outcomes with the Mini-Project

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

PROGRAM SPECIFIC OUTCOMES

PSO No.	Program Specific Outcomes
PSO1	To analyse, design and develop computing solutions by applying foundational concepts of computer science and engineering.
PSO2	To apply software engineering principles and practices for developing quality software for scientific and business applications.
PSO3	To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems.

PSO1	PSO2	PSO3

Signature of Guide

ACKNOWLEDGEMENT

I wish to acknowledge with thanks, the significant contribution given by the management of our college **Chairman, Dr. M. V. Muthuramalingam**, and our **Chief Executive Officer Thiru. M.V.M. Velmurugan**, for their extensive support.

I would like to thank **Dr. S. Satish Kumar, Principal** of Velammal Engineering College, for giving me this opportunity to do this project.

I express my gratitude to our effective **Head of the Department, Dr. B. Murugeswari**, for her moral support and valuable innovative suggestions, constructive interaction, constant encouragement, and unending help that have enabled me to complete the project.

I express my thanks to our Project Coordinators, **Dr. S. Rajalakshmi, Dr. P. S. Smitha and Dr. P. Pritto Paul**, Department of Computer Science and Engineering for their invaluable guidance in the shaping of this project.

I express my sincere gratitude to my **Internal Guide, Mrs. Aarthi, Assistant Professor**, Department of Computer Science and Engineering for her guidance, without her this project would not have been possible.

I am grateful to the entire staff members of the Department of Computer Science and Engineering for providing the necessary facilities and carrying out the project. I would especially like to thank my parents for providing me with the unique opportunity to work and for their encouragement and support at all levels. Finally, my heartfelt thanks to **The Almighty** for guiding me throughout my life.

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
	ABSTRACT	
1	INTRODUCTION	
	1.1 PURPOSE OF THE PROJECT	9
	1.2 SCOPE OF THE PROJECT	9
2.	DOMAIN INTRODUCTION	
	2.1 WEB DEVELOPMENT	9
	2.2 APPLICATION PROGRAMMING INTERFACE(API)	10
3.	SYSTEM ANALYSIS	
	3.1 EXISTING SYSTEM	11
	3.2 PROPOSED SYSTEM	11
4.	SYSTEM SPECIFICATION	
	4.1 SOFTWARE SPECIFICATION	12
	4.2 HARDWARE SPECIFICATION	12
5.	SYSTEM DESIGN	
	5.1 USE CASE DIAGRAM	13
6.	CONCLUSION	14
	APPENDIX 1	14
	SNAPSHOT	22

ABSTRACT

Weather-Verse is a web-based platform designed to provide accurate and reliable weather predictions to users worldwide. The website aims to enhance user experience by offering accurate and timely weather data in an easily accessible format. The platform leverages the power of Application Programming Interface (API) to gather real-time and historical weather data, enabling users to access forecasts more efficiently. The website utilizes a weather API to retrieve current weather conditions such as temperature, humidity, wind speed, and pressure levels for specific locations. The developed website is can fetch weather all over the world and gives accurate prediction by utilizing the API. Furthermore, the project emphasizes data accuracy and reliability by regularly updating weather information based on the latest data from the API provider. This ensures that users receive up-to-date and precise weather forecasts for their chosen locations. It also provides the details like minimum and maximum temperature of the particular day and time. The website's responsive design ensures compatibility across various devices, including desktop computers, tablets, and smartphones, enabling users to access weather information on the go. Accessibility features such as screen reader compatibility and color contrast options enhance the inclusivity of the website, catering to users with diverse needs. Weather-Verse is a comprehensive platform for users to access reliable weather information effortlessly, aiding in making informed decisions related to their daily activities. The integration of API allows seamless data retrieval and ensures that the website is up-to-date which has recent weather forecasts.

1. INTRODUCTION

1.1 PURPOSE OF THE PROJECT

The emergence of technology and the increasing accessibility of weather APIs have made the creation of a weather prediction website not only possible but also quite advantageous. The main purpose of developing Weather-Verse is to provide users with reliable and user-friendly platform for accessing and viewing accurate weather information. Access to accurate and trustworthy weather information is crucial for individuals, businesses, and organizations in the fast-paced world of today. Current weather forecasts and statistics are essential for organizing outdoor activities, booking trips, maximizing agricultural techniques, and becoming ready for weather-related emergencies. The emergence of technology and the increasing accessibility of weather APIs have made the creation of a weather prediction website not only possible but also quite advantageous.

1.2 SCOPE OF THE PROJECT

The scope of Weather-Verse is to produce accurate weather information across the globe which considers humidity, pressure minimum and maximum temperatures. The intend of this project is to give accurate prediction of weather conditions using series of algorithms applied through API. In order to provide the user with high performance in weather prediction this project is used.

2. DOMAIN OF THE PROJECT

2.1 WEB DEVELOPMENT

Web development gives the wide range of activities related to creating web and non-web applications and maintaining it. It contributes to the key components of the website which interacts with the user. The development is divided into,

- (i) Front-end development
- (ii) Backend development

(i) Front-End Development

The main focus of front-end development is User Interface (UI) and user experience (UX) of the website. Front end tools like HTML, CSS, Javascript is used so users can view and interact with the website. Some of the front-end tools are,

(i) HTML (Hyper Text Markup Language):

The primary language used in web development is HTML. HTML is a programming language used by front-end developers to define the content, layout, headers, paragraphs, pictures, links, and other components that make up a web page. The structure and order of material on a web page are largely determined by HTML.

(ii) CSS (Cascading Style Sheets):

CSS is used to style and visually enhance the appearance of HTML elements. Front-end developers use CSS to define colors, fonts, spacing, alignment, borders, backgrounds, and other visual aspects of web pages. CSS allows developers to create responsive designs that adapt to different screen sizes and devices.

(iii) Javascript:

JavaScript is a computer language used to give web pages dynamic behaviour and interactivity. JavaScript is used by front-end developers to generate interactive features including animations, user-triggered actions, dropdown menus, pop-up modals, sliders, and form validations. Angular, Vue.js, and React are a few JavaScript frameworks and modules that make it easier to construct complicated front-end features.

2.2 APPLICATION PROGRAMMING INTERFACE (API):

An API, or application programming interface, acts as a link between various platforms, apps, and software systems so they may interact and communicate with one another. APIs provide developers the ability to access and utilize features, data, or services offered by other programs or systems by defining a set of guidelines, conventions, and instruments for creating and integrating software components. The following are the main features and attributes of APIs:

(i) Interoperability: APIs facilitate interoperability by standardizing communication between software components or systems, regardless of their underlying technologies, programming languages, or architectures. This interoperability enables seamless integration and interaction between diverse systems, applications, or services.

(ii) Functionality Extension: APIs allow developers to extend the functionality of their applications by integrating external APIs or third-party services. For example, integrating a

payment gateway API enables processing online payments, while integrating a mapping API enables displaying interactive maps and geolocation features within an application.

(iii) Data Access and Services: APIs provide access to data, functionalities, or services offered by an application or platform. This could include retrieving data from a database, accessing cloud services, interacting with web services, sending and receiving messages, processing requests, or performing specific operations.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Google weather locates and displays current weather forecast, hourly and it can even has 7 day forecast which predict the weather for next 7 days. Though it has useful features, it lacks in performance and limited features. Generally, the models which is been created will have more error which impacts user satisfaction and results in wrong updation of weather information. Weather.com uses lot of PNG and JPG files that are considered outdated and heavy for modern web use.

DRAWBACKS

- Limited Data Sources
- Outdated Interface
- Lack of Customization
- Poor performance
- Limited Features

3.2 PROPOSED SYSTEM

The proposed system resolves the drawbacks of existing system. The system developed shows pressure and wind conditions for the given particular area. It uses an API from OpenWeatherMap which provides global weather database. It provides minute-by-minute hyperlocal precipitation forecast. The convolutional machine learning model uses meteorological broadcast services and data form airport weather stations, weather satellites, on-ground radar stations, remote sensing satellites, METAR and automated weather stations. Furthermore, the proposed system is provided with the advanced features where user can see pressure around the area.

The website is linked through API from OpenWeatherMap organization which gives the accurate information on climatic and weather conditions. OpenWeatherMap team analysed the behaviour of metrics and found Mean Absolute Error(MAE) as 0.5 and the data provided by them has Root Mean Square (RMSE) less than 2 degrees. The reliability of prediction is nearly 90% to 100% only and inaccuracy is about only 1%. This shows that the prediction of weather is more accurate and give high performance.

4. SYSTEM SPECIFICATION

4.1 SOFTWARE SPECIFICATION:

- HTML5
- CSS3
- Javascript ES6
- Visual Studio code

Visual Studio Code: Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft. It is designed to be lightweight, fast, and customizable, making it a popular choice for developers across a variety of programming languages and platforms. One of the key features of VS Code is its support for extensions, which allows developers to add functionality and customize the editor to suit their needs. There are thousands of extensions available, ranging from language-specific syntax highlighting and autocomplete to tools for debugging, testing, and version control. In addition to its extensibility, VS Code also includes a range of built-in features to support efficient coding, such as IntelliSense, which provides context-aware code completion and suggestion, and a built-in terminal for running commands and scripts. VS Code also includes support for debugging and profiling, with integrated tools for debugging Node.js, Python, and other languages, as well as the ability to attach to remote processes for debugging and profiling in a distributed environment.

4.2 HARDWARE SPECIFICATION:

Hardware environment refers to the physical components that makeup a computer system, including the processor, memory, storage, input/output device. Environment can have a significant impact on the performance and capabilities of a computer system and is an important consideration when choosing a system for a specific application.

- **Processor:** Intel Core i7 or AMD Ryzen
- **Memory:** 16GB of RAM
- **Storage:** Solid-state drives (SSDs)

5. SYSTEM DESIGN

5.1 USE-CASE DIAGRAM

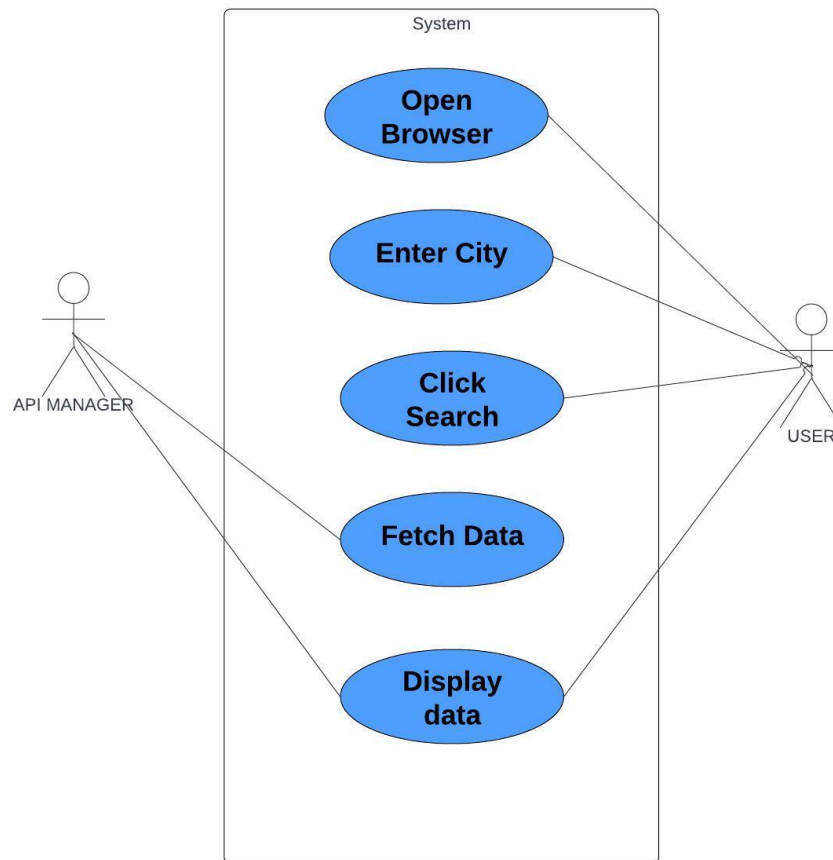


Figure 1 Use-case diagram for Weather-Verse usage

CONCLUSION

The development of Weather-Verse using API represents the significant leap forward in providing users with accurate and easily accessible weather information. By leveraging agile development practices, user feedback mechanisms, and continuous improvement strategies, the proposed system is poised to evolve dynamically, incorporating new technologies, addressing user feedback, and staying ahead of industry trends. The collaboration with meteorological experts, adherence to best practices in API management, and commitment to data accuracy and security further reinforce the system's credibility and reliability.

APPENDIX 1

SOURCE CODE

Index.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+nuLcSKlbnrPcLa0OT92xO1BIsZ+ymDWZCvqsWgccV3gFoRBv0z+8dLJgyA
HlhR35VZc2oM/gI1w=="
    crossorigin="anonymous" referrerpolicy="no-referrer" />
<!-- sweetalert cdn..... -->
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
</body>
</html>

```

Style.css

```

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

```

Script.js

```

//making object of weatherapi
const weatherApi = {
  key: '4eb3703790b356562054106543b748b2',
  baseUrl: 'https://api.openweathermap.org/data/2.5/weather'
}
//anonymous function
//adding event listener key press of enter
let searchInputBox = document.getElementById('input-box');
searchInputBox.addEventListener('keypress', (event) => {
  if (event.keyCode === 13) {
    // console.log(searchInputBox.value);
    getWeatherReport(searchInputBox.value);
  }
})//get waether report
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;
}

```

SNAPSHOT

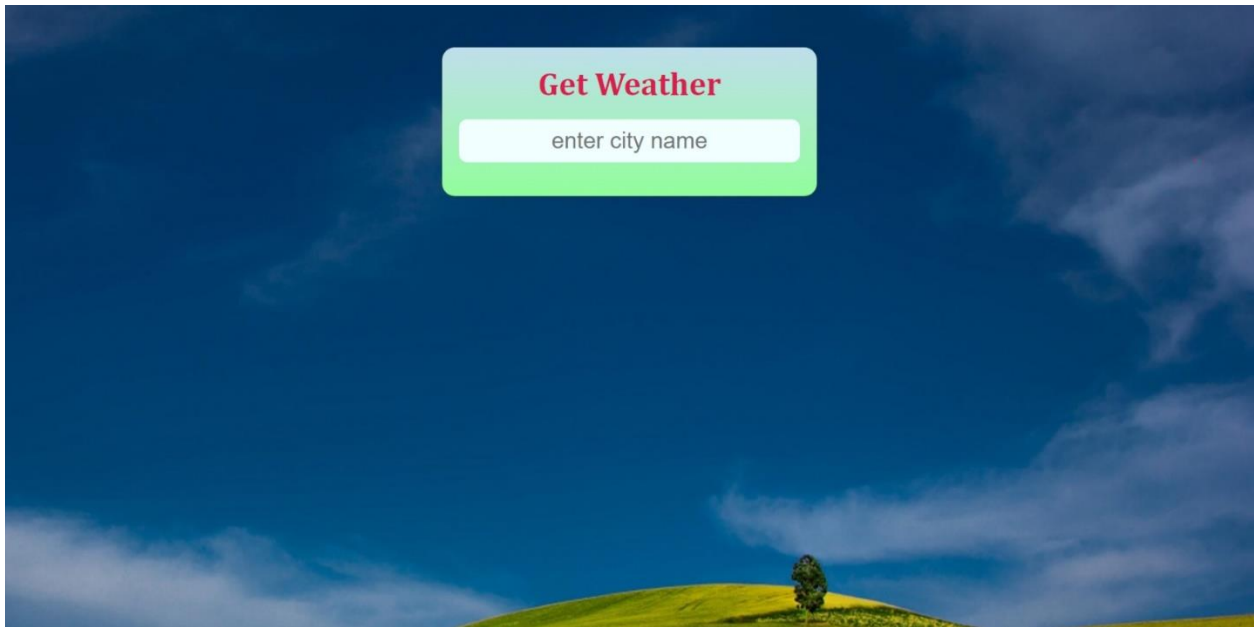


Figure 2 Home page of Weather-Verse

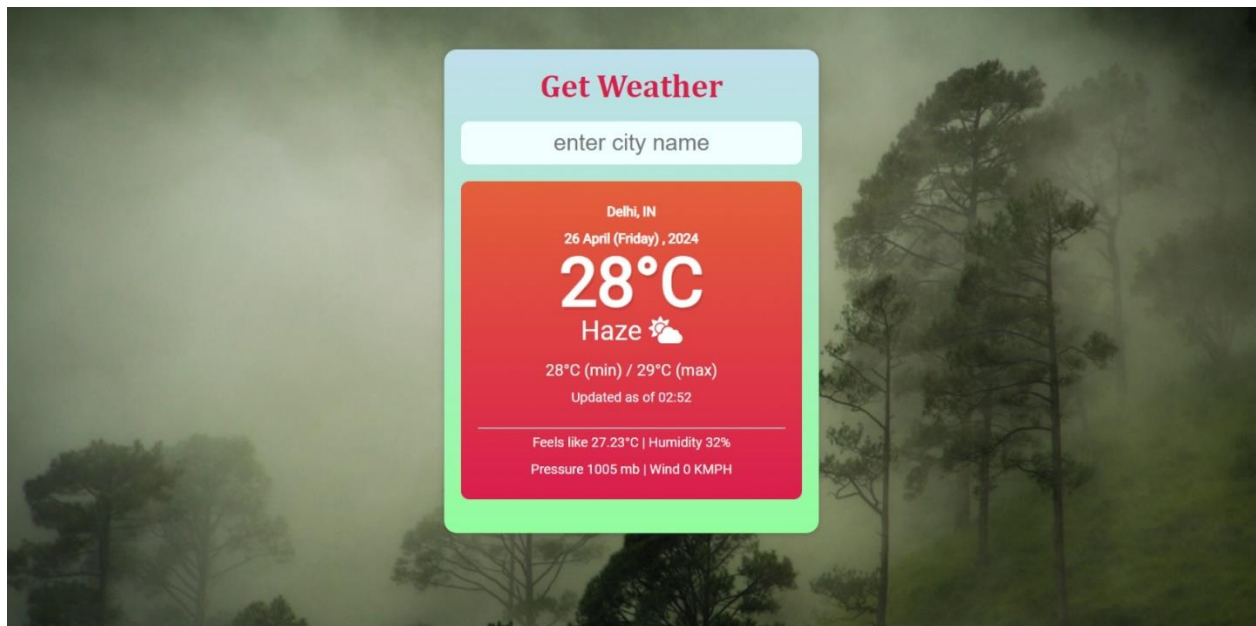


Figure 3 Weather Report Generation

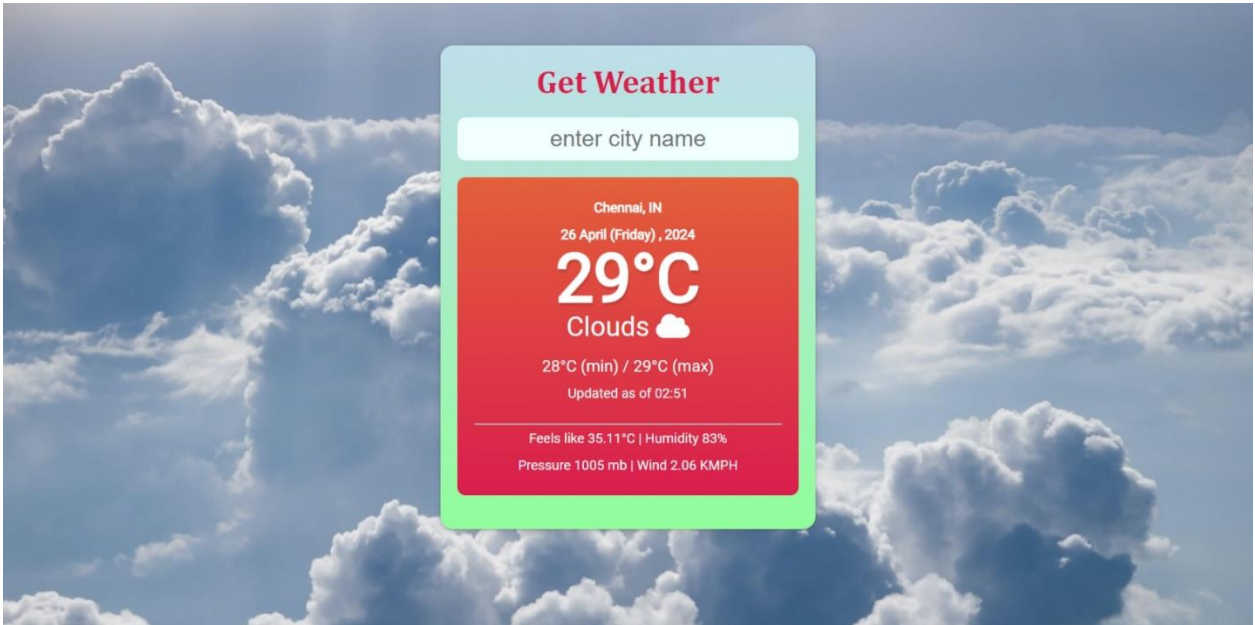


Figure 4 Local Area Weather Report Generation

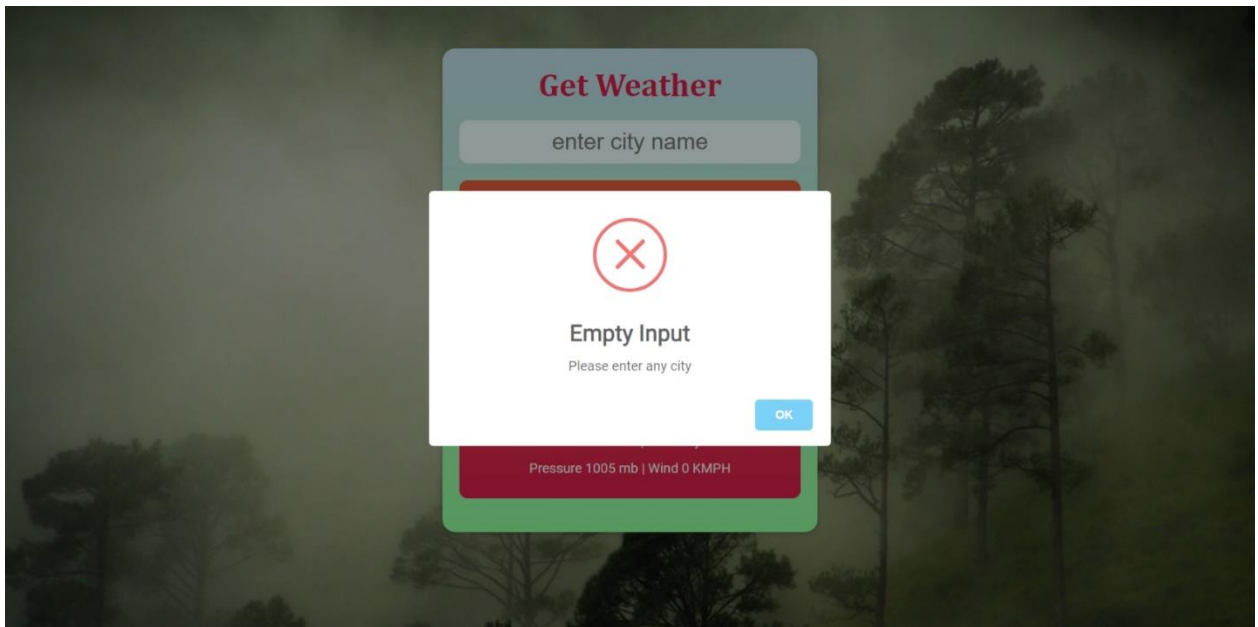


Figure 5 Error case for empty input