

FETCHING WEATHER INFO

**A Project Report Submitted in partial fulfilment of the requirements for
the award of the degree of**

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Sishir Bohara

18221A05B9

Rupali Adhikari

18221A05C0

Devram Yadav

18221A05B8

Under the Supervision of

Dr. B S N Murthy

M.Tech, Ph.D.

Associate Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE
(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with 'A' Grade)

ODALAREVU – 533210

2018-22

BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE
(AUTONOMOUS)

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with 'A' Grade)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project work entitled **“FETCHING WEATHER INFORMATION”** is being submitted for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, at **BVC Engineering College, Odalarevu**, is a bonafide work done by **Sishir Bohara (18221A05B9)**, **Rupali Adhikari (18221A05C0)**, **Devram Yadav (18221A05B9)** under the academic year 2021-22 and it has been found suitable for acceptance according to the requirement of University. The results embodied in this thesis have not been submitted to any other University Institute for the award of any degree.

Project Guide

Dr. B S N Murthy

M.Tech. Ph.D

Associate Professor

Head of the Department

Dr. G Jena

M.Tech. Ph.D

Professor

External Examiner

ACKNOWLEDGEMENT

We express our deep sense of gratitude and respect towards our guide **Dr. B S N Murthy Associate Professor** of CSE Department, BVC Engineering College, Odalarevu for his excellent guidance right from selection of the Project work. Her constant encouragement and support have been the cause of our success in completing this thesis in the college.

We are thankful To **Dr. GUNAMANI JENA, Professor and Head of the Department of CSE** for facilitating, Training, providing us the counselling and support to carry out the project work.

We are also thankful to **Dr. Chandramouli V S A, Principal**, BVC Engineering College, Odalarevu, for his support during and till the completion of the project.

We are thankful to all the **teaching and non-teaching staff** of Computer Science & Engineering Department, **Management**, BVC Engineering College, Odalarevu, and our friends for their direct and indirect help provided to us in completing the project.

Project Associates

Sishir Bohara (18221A05B9)

Rupali Adhikari (18221A05C0)

Devram Yadav (18221A05B8)

ABSTRACT

The idea of the project is to implement the web page design for fetching the current weather information of the City. For this web page design we use the concept of Hypertext markup language, Cascading Style Sheet and Javascript.

In this web page, API plays an important role. API is the Application Programming Interface which is used to communicate the client site and server site. We use the OpenWeatherMap API for this web page.

HTML is the standard markup language for Web pages. It describes the structure of the Web page.

CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on screen. It can control the layout of multiple web pages all at once.

JavaScript is the world's most popular programming language. It is the programming language of the Web. JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

TABLE OF CONTENT

1. Introduction

1.1 Introduction

1.2 Objective

1.3 Significance of the Project

2. Literature Review

3. Web Page Requirement

3.1 Functional Requirements

3.2 Nonfunctional Requirements

3.3 Modules and their Functionalities

4. Web Page Design

4.1 Introduction to HTML

4.2 Introduction to CSS

4.3 Introduction to Javascript

4.4 API: Open Weather-Map API

5. Coding

5.1 Main Program

5.1.1 HTML script to design webpage for fetching weather info

5.1.2 CSS script to style the webpage

5.1.3 JavaScript program to show the webpage functioning dynamically

6. Testing

7. Output Screens

8. Conclusion

9. Reference

1. INTRODUCTION

1.1 Introduction

With the evolving Era of Technology, the society and world has come this far with the different marvellous works of Science and Technology. The use of technology can be in any field like : Education, Health, Transport, Banking and many more which we come across on our daily life. The mobile phone, Laptop, TV etc are the workshop science which contains very small or huge part of Computer Science.

In this Project, we will see one of the creation of weather information webpage which is small to think but has a very important play in today's time. The title of our project is "Fetching Weather Information" where we see the weather of different Cities in current time around the world. Webpages were first formed by Tim Berner's Lee in the way of paving a path to introduce more interesting fact about it. Webpages can be defined as the hypertext document provided by a website and displayed to a user in a web browser.

Here we use several programming skills which consist of Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript (JS). The platforms we have used in this project are Github, Virtual Studio, Netlify. The below explanation of the project is done in a elaborated way so its uses can be discovered vividly.

1.2 Objective

The main objective of this project is to fetch the current weather information of numbers of cities. We used Open Map API to create this website which shows us the following in our webpage:

- Temperature in degree Celsius, Fahrenheit and

Kelvin

- Condition of the climate: Cloud, Humidity

1.3 Significance of Project

Users can use the website explicitly using the internet to access it over any kind of device. Users can input their cities as well as the name of cities which weather report they want currently. It helps to check if the difference in climate of different cities are similar or not. Looking at the cloudy information one can predict the rainfall in future too as a use of this website.

2. LITERATURE REVIEW

Weather fetching has been one of most challenging difficulties around the world because of both its practical value in popular scope for scientific study and meteorology .weather is continuous ,dynamic, multi-dimensional chaotic process. The various methodology viz. Static decomposition model, Exponential smoothing models, ARIMA model.

In this project, we will be building a weather application. This application will show the temperature of a location. To fetch weather information we will need an API. An [API\(Application Programming Interface\)](#) is a function that allows applications to interact and share data using various components and microservices.

For this project, we will be using WeatherBit API for fetching weather data. WeatherBit API provides a fast and elegant way to fetch weather data. We use this API to fetch the weather report such as_(Temperature, humidity, wind speed and descriptions of weather of any place in world. You need to just type the name of the places and will get the current weather details.

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.

Here are some examples for that format:

- ☐ Built-in API request by city ID
- ☐ You can make an API call by city ID. List of city ID 'city.list.json
- ☐ We recommend to call API by city ID to get unambiguous result for your city.

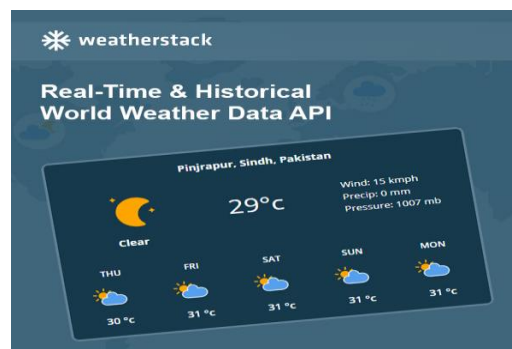
Call by: `api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}`

❑ ***Built-in API request by ZIP code***

NOTE: if country is not specified then the search works for NEPAL as a default.

API call : `api.openweathermap.org/data/2.5/weather?zip={zip code},{country code}&appid={API key}`

After completion of some format the result is showing like this figure:



3. WEBPAGE REQUIREMENT

Functional requirements define a function of a system or its component. A function described as a set of inputs, the behavior, and outputs. It can be documented in various ways. The most common ones are written descriptions in documents, and use cases. Functional requirements is what a system is supposed to accomplish. It may be

- Calculations
- Technical details
- Data manipulation
- Data processing

A typical functional requirement will contain a unique name and number, a brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, and to track the requirement through the development.

Software Requirements:

- **Operating System:** Windows/Linux/Mac
- **Programming Language:** Javascript
- **Scripting Language:** HTML, CSS

- IDE: Visual Studio Code

Hardware requirements

- RAM: 512MB or higher
- Processor: Intel Pentium 4 or higher/AMD Althon Xp or higher (2.4GHz)

3.2 Non-functional requirements

A non- functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Nonfunctional requirements are called qualities of a system, they are:

1. Performance Requirements

- Response times
- Processing time (How long is acceptable to perform key functions are export/import data?)
- Query and reporting times

2. Interface Requirements

- Good interface (How easier is the interface for the user to interact without any prior knowledge)

3. Documentation Requirements

- Easily Understandable documentation

4. Portability Requirements

- Highly portable (Since this can be used even in mobile phones)

5. Quality Requirements

- Quality of communication

6. Reliability Requirements

- How reliable is this project when compared to written communication?

4. WEBPAGE DESIGN

Designing has always been one of the top priorities on creating anything either related to IT or non-IT. The project we created consists of different scripting languages which we have used for designing our webpage so that it is well formed and looks attractive to the user and gives the user an appropriate searching interface with interaction. The web designing sometimes is also known as web scraping in few areas of webpage or website development. The following are the scripting languages we used to design our project:

HTML:

- HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.
- Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version is known as [HTML5](#).
- A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes**.

Description:

- HTML (Hyper Text Markup Language) is a language for specifying how text and graphics appear on a web page.
- When you visit a web site (e.g., www.google.com) your web browser retrieves the HTML web page and renders it.

- The HTML page is actually stored on the computer that is hosting the web site and the page is sent to your browser.
- HTML code is stored in a simple text file that has either a .htm or a .html filename extension (e.g., restaurant.html).
- A webpage is a document written in HTML and can be viewed on any web browser.
- It is contained within the web server, which can be accessed by entering the URL for that web page, and once it is loaded, it appears on the user's web browser.
- Each webpage is linked with a unique URL, hence two pages cannot have the same URL.
- Webpage is a part of a website; it means a website contains different web pages. Such as google.com is a website, and the page currently you are accessing is the webpage.
- The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.
- HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

Syntax of HTML:

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>
```

```
</head>
<body>

<h1>Heading1</h1>

<h2>Heading2</h2>

<h3>Heading3</h3>

<h4>Heading4</h4>

<h5>Heading5</h5>

<h6>Heading6</h6>

<p>paragraph.</p>

</body>
</html>
```

Output:

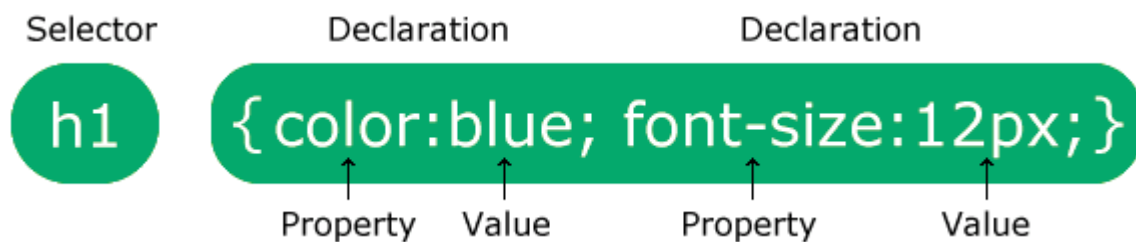
Heading 1
Heading 2
Heading 3
Heading 4
Heading 5
Heading 6
Paragraph

CSS:

- CSS stands for Cascading Style Sheets.
-
- External stylesheets are stored in CSS files
- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements should be displayed.
- This tutorial will teach you CSS from basic to advanced.

- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
-
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
-
- External stylesheets are stored in CSS files.
-
- CSS removed the style formatting from the HTML page!

Syntax:



Example:

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

There are three different ways you can use to insert CSS definitions in your web page. These are:

1. **Inline Style**
2. **Embedded Style Sheet**
3. **External Style Sheet**

Inline CSS: Styles sheets that are of type inline refer to information related to the style being functional to the existing HTML element. Using an inline approach, rather than defining your style once, you have to write the style in every HTML element you use to design your web page. It can be more precisely called an *inline style* rather than the *inline style sheet*. It uses the style attribute within that HTML element.

Syntax:

```
<HTML ELEMENT style="properties: value"> .... </HTML ELEMENT>
```

Example:

```
<p style="color: blue">The text gets the effect of inline style.</p>
```

Embedded Style Sheet : Embedded Style Sheets is a style sheet where designers can embed information of the style sheet in an HTML document by making use of the <style> element. This embedding of style sheet info within <style> ... </style> tags are done within head section of HTML.

The syntax for embedded style sheets has no such exception. Simply you have to place the style sheet code between the <head>.....</head> tags where <style> </style> is nested within head element:

Example:

```
<!DOCTYPE html>  
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Embedded Style Sheets Example</title>
<style>
h1 {
border-bottom: 1px solid #DDDDDD;
color: #069;
font-family: Helvetica, Arial;
font-size: 25px;
font-weight: normal;
line-height: 34px;
margin-bottom: 10px;
outline: 0 none;
padding-bottom: 3px;
padding-top: 0;
text-decoration: none;
}

hr {
background-color: #069;
border: 0 none;
clear: both;
color: #D4D4D4;
height: 1px;
}

.sublines {
background-color: #DAF4FE;
padding: 5px;
border: 1px solid #09C;
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
}

.infotext {
font-size: 10pt;
background-color: #F2F2F2;
padding: 5px;
}
</style>
</head>

<body>
<h1><span class="headlines">Welcome to w3schools.in</span><br>
</h1>
<div class="sublines"> This is an example page using CSS.
<br> The example is really simple,
<br> and doesn't even look good,
<br> but it shows the technique. </div>
<br>
```

```

<table border="0" cellpadding="3" cellspacing="1">
<tr>
<td class="sublines"> As you can see:
<br>
</td>
<td class="sublines">The styles even work on tables.</td>
</tr>
</table>

<hr>

<div class="infotext">Example from w3schools.in</div>
<hr>
</body>
</html>

```

External Style Sheet : This type of style sheet gets a separate file in which designers can state every CSS styles that seem relevant for your web site.

Then this has to be linked with the external style sheet from your HTML page. You have to follow some specific steps to make this conceptual style sheet implementable.

Steps to create External Style Sheets:

1. Build the Style Sheet by typing the CSS code in a plain text file (using text editor, usually), and then save the with as a `.css` extension.
2. You have to link the Style Sheet with the HTML document by using an HTML link element.

Example:

```

<head>

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

```

JavaScript

- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.

- JavaScript is easy to learn.
- This tutorial will teach you JavaScript from basic to advanced.
- JavaScript is a lightweight, interpreted programming language.
- It is designed for creating network-centric applications.
- It is complimentary to and integrated with Java.
- JavaScript is very easy to implement because it is integrated with HTML.
- It is open and cross-platform.
- JavaScript can be implemented using JavaScript statements that are placed within the `<script> </script>` HTML tags in a web page.
- You can place the `<script>` tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the `<head>` tags.
- The `<script>` tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows:

`<script>`

JavaScript code

</script>

The script tag takes two important attributes:

- **Language:** This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type:** This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript syntax will look as follows:

```
<script language="javascript" type="text/javascript">  
    JavaScript code  
</script>
```

Example:

```
<html>  
<body>  
<script language="javascript" type="text/javascript">  
<!--  
    document.write ("Hello World!")
```

```
//-->  
</script>  
</body>  
</html>
```

Output:

Hello World!

API: (Application programming interfaces)

- Application programming interfaces, or APIs, simplify software development and innovation by enabling applications to exchange data and functionality easily and securely.
- An application programming interface, or API, enables companies to open up their applications' data and functionality to external third-party developers, business partners, and internal departments within their companies.
- This allows services and products to communicate with each other and leverage each other's data and functionality through a documented interface.
- Developers don't need to know how an API is implemented;
- they simply use the interface to communicate with other products and services.

- API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.
- An API is a set of defined rules that explain how computers or applications communicate with one another.
- APIs sit between an application and the web server, acting as an intermediary layer that processes data transfer between systems.

Common API examples:

- Universal logins
- Third-party payment processing
- Travel booking comparisons
- Google Maps

Types of APIs

- Open APIs
- Partner APIs
- Internal APIs
- Composite APIs

Open Map Weather API:

OpenWeatherMap is an online service, owned by OpenWeather Ltd, that provides global weather data via API, including current

weather data, forecasts, nowcasts and historical weather data for any geographical location. The company provides a minute-by-minute hyperlocal precipitation forecast for any location. The convolutional machine learning model is used to utilise meteorological broadcast services and data from airport weather stations, on-ground radar stations, weather satellites, remote sensing satellites, METAR and automated weather stations. The founders of this API were Denn Ukolov, Olga Ukolova in 2012. We can access this API from Website: openweathermap.org.

The variety of weather APIs provided by Open Weather Map have found a significant popularity among the software developers, which resulted in the growing multitude of repositories on GitHub. The APIs support multiple languages, units of measurement and industry standard data formats like JSON and XML. In 2021, Open Weather Map launched a number of initiatives to support students, researchers and developers across the world.

5.CODING

//html script to design webpage for fetching weather info.

Weather.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>OpenWeatherMap Api</title>
  <link rel="stylesheet" href="weather.css">
  <link rel="shortcut icon" href="favicon.ico" type="ico">
</head>

<body>
  <div class="main">
    <div class="header">
      <h1>OpenWeatherMap API</h1>
      <p>Enter any city name in the input box below to get the
data</p>
    </div>

    <div class="input">
      <input type="text" placeholder="Enter the city"
class="input_text">
      <input type="submit" value="Check" class="submit">
    </div>
  </div>

  <div class="container">
    <div class="card">
      <h1 class="name" id="name"></h1>
      <p class="temp"></p>
      <p class="min-temp"></p>
      <p class="max-temp"></p>
      <p class="humidity"></p>
      <p class="clouds"></p>
      <p class="desc"></p>
    </div>
  </div>
<script src="weather.js"></script>
</body>
</html>
```

//css script to style the webpage.

Weather.css

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

body{
  font-family: "Nunito",sans-serif;
  background: #f8f8f8;
  background-image: url("weather.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
}

.header {
  text-align: center;
}

.input{
  text-align: center;
  margin: 100px auto;
}

input[type="submit"]{
  padding: 15px 30px;
  background: lightskyblue;
  border: none;
  border-radius: 1px;
  font-family: "Nunito",sans-serif;
  font-weight:bold;
  font-size: 20px;
  cursor: pointer;
}

.input input[type="text"]{
  width: 600px;
  height: 55px;
  padding: 5px 10px;
}
```

```
background: lightgray;
border: none;
border-radius: 1px;
font-family: "Nunito", sans-serif;
font-weight: bold;
font-size: 20px;
}

.card{
width: 50%;
background: lightgreen;
height: 55vh;
margin: 50px auto;
border-radius: 2px;
opacity: 0.7;
}

.close{
float: right;
margin-top: -2px;
cursor: pointer;
margin-right: 20px;
}

.card h1{
padding: 5px 0;
text-align: center;
color: darkblue;
}

.card p{
text-align: center;
margin: 40px 0;
font-size: 20px;
}
```

//javascript to show the webpage functioning dynamically.

Weather.js

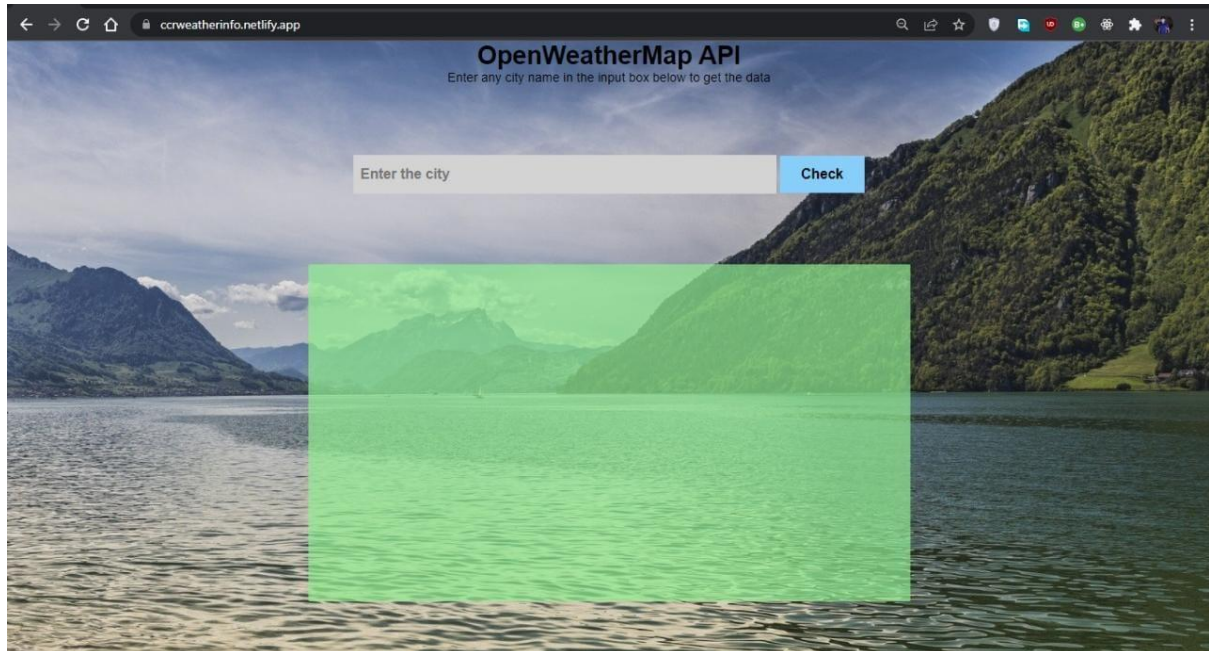
```
var input = document.querySelector('.input_text');
var main = document.querySelector('#name');
var temp = document.querySelector('.temp');
var desc = document.querySelector('.desc');
var min = document.querySelector('.min-temp');
var max = document.querySelector('.max-temp');
var humidity = document.querySelector('.humidity');
var clouds = document.querySelector('.clouds');
var button = document.querySelector('.submit');

button.addEventListener('click', function(name){
  fetch('https://api.openweathermap.org/data/2.5/weather?q='+input.value+'&appid=e4811886ad5213b761877c134bad03b3')
    .then(response => response.json())
    .then(data => {
      var tempValue = data['main']['temp']-273.15;
      var nameValue = data['name'];
      var humidity = data['main']['humidity'];
      var mintemp = data['main']['temp_min']-273.15;
      var maxtemp = data['main']['temp_max']-273.15;
      var descValue = data['weather'][0]['description'];

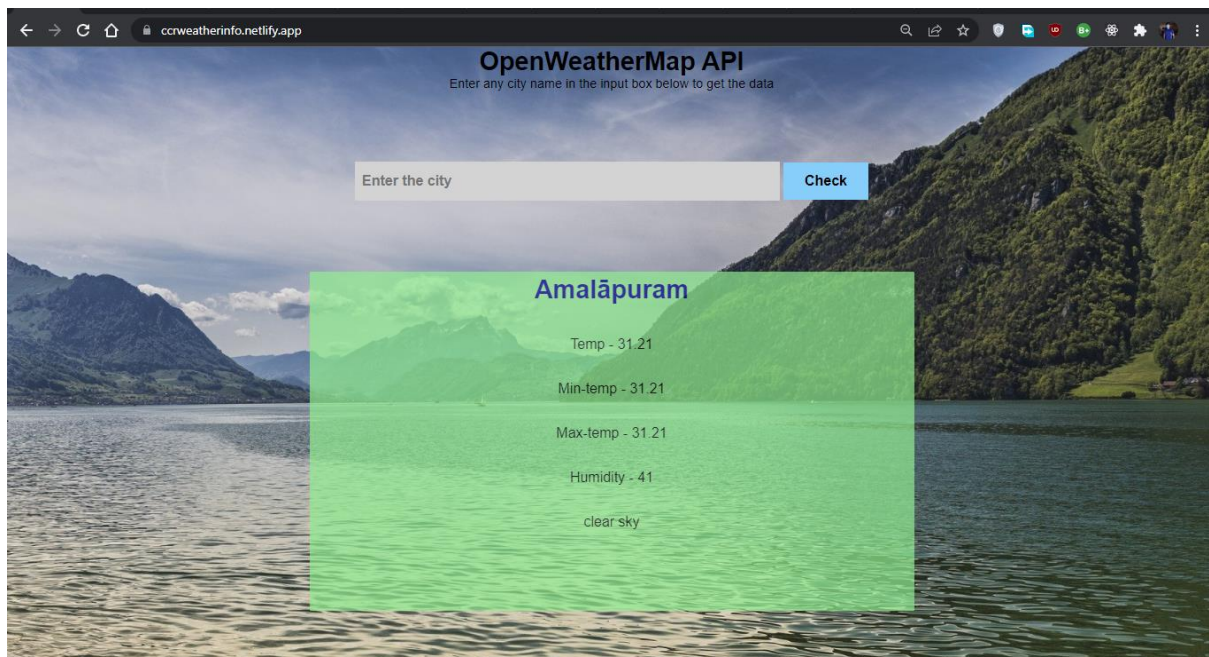
      main.innerHTML = nameValue;
      desc.innerHTML = ""+descValue;
      temp.innerHTML = "Temp - "+tempValue.toFixed(2);
      humidity.innerHTML = "Humidity - "+humidity;
      min.innerHTML = "Min-temp - "+mintemp.toFixed(2);
      max.innerHTML = "Max-temp - "+maxtemp.toFixed(2);
      input.value = "";
    })
    .catch(err => alert("Wrong city name!"));
})
```

6.OUTPUT SCREEN

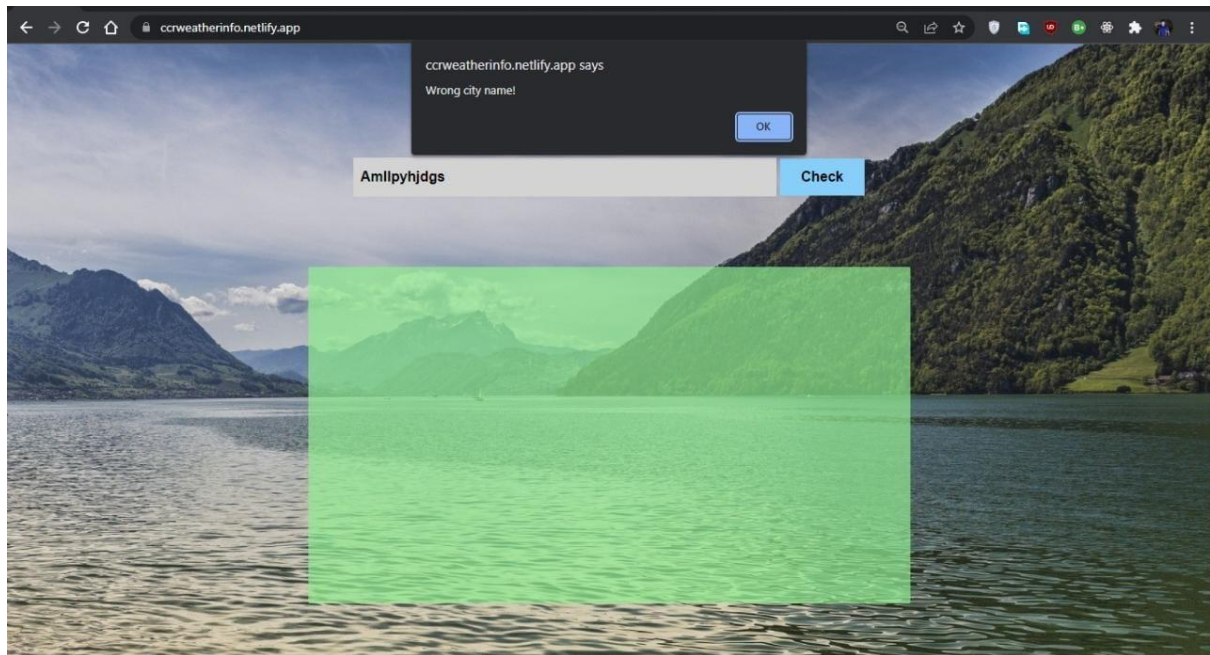
// The output when we don't give any input



// The output when we don't give any input



// The output when we give wrong city name



7.TESTING

The development of software includes a series of productive activities and testing is an important activity of them. This phase is a critical element of software quality assurance and represents the ultimate review of specification, coding and testing.

The main objectives of testing are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Testing can be done in different ways. Some of the types of testing are mentioned below. The main purpose of any type of test is to systematically uncover different classes of errors and do so with a minimum amount of time and effort.

Types of Webpage Testing are;

1. Functionality Testing.
2. Usability Testing.
3. Web UI Testing.
4. Compatibility Testing.
5. Performance Testing.

6. Security Testing

1. **Functionality Testing:** Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing, etc.
2. **Usability Testing:** Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.
 - Ease of learning
 - Navigation
 - Subjective user satisfaction
 - General appearance
3. **Web UI Testing:** In web testing, the server-side interface should be tested. This can be done by verifying that the communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.

The main interfaces are:

- Web server and application server interface
- Application server and Database server interface
- Check if all the interactions between these servers are executed and errors are handled properly. If the database or web server returns an error message for any query by the application server then the application server should catch and display these error messages appropriately to the users.
- Check what happens if the user interrupts any transaction in-between? Check what happens if the connection to the

webserver is reset in between?

4. Compatibility Testing: The compatibility of your website is a very important testing aspect. See which compatibility test to be executed:

- Browser compatibility
- Operating system compatibility
- Mobile browsing
- Printing options

5. Performance Testing : The web application should sustain a heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing

Test application performance at different internet connection speeds.

6. Security Testing : The following are some of the test cases for web security testing:

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages, then try changing URL options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the URL site ID parameter to a different site ID which is not related to the logged-in user. Access should be denied for this user to view other people's stats.
- Try some invalid inputs in input fields like login username, password, input text boxes, etc. Check the system's reaction

to all invalid inputs.

- Web directories and files should not be accessible directly unless they are given the download option.
- Test the CAPTCHA to automate script logins.
- Test if SSL is used for security measures. If used, the proper message should get displayed when users switch from non-secure HTTP:// pages to secure HTTPS:// pages and vice versa.
- All transactions, error messages, and security breach attempts should be logged in log files somewhere on the webserver.

The primary reason for testing the security of a web is to identify potential vulnerabilities and subsequently repair them :

- Network Scanning
- Vulnerability Scanning
- Password Cracking
- Log Review
- Integrity Checkers
- Virus Detection

8.CONCLUSION

Today's Web development, a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. In general, a good page layout has to satisfy the basic elements of a good page design

The work of the project to work on the weather API to perform some tasks valid or invalid. As we can see we get results for valid as well as the invalid inputs. The project first and foremost idea is to make it easier to know about the climatic condition of the current places they are staying to may be further predict the weather or take refuge from any storm or any calamities looking at the cloud and climate.

The website is cost effective as it is free. Web content is a critical business asset worthy of meaningful spend. A webpage helps you to focus resources which the customers care about constantly. Web designing and web development has become one of the important aspect of todays life.

9.REFERENCE

Best Weather APIs

OpenWeatherMap : Best for Weather Forecast

Weatherbit : Best for Weather Forecasr and Alerts

AccuWeather : Best for Weather Conditions, images, Cyclones and More

DarkSky : Best for Forecast and Historical Data

Weather2020 : Best for Long Range Weather Forecast

Tomorrow.io(formerlyClimaCell) : Best for Realtime, Short Term and Hourly Forecast

VisualCrossing : Best for Weather Forecast and Historical Data

AerisWeather : Best for Alerts, Condition, Forecasts