

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology by

Harsh Gupta
(Reg. No. 37120029)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119, TAMILNADU
DECEMBER 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with 'A' grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 3151 / 3152 / 3154 / 3155 Fax: 044 - 2450 2344
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **HARSH GUPTA (REG NO.37120029)** who underwent the professional training in **“WEATHER FORECAST USING WEB DEVELOPMENT”** under our supervision from AUGUST 2020 to JANUARY 2021.

Internal Guide

Mr. P.SARAVANAN, M.E.,(Ph.D).,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination heldon_____

Internal Examiner

External Examiner

DECLARATION

I, **HARSH GUPTA (Reg. No 37120029)** hereby declare that the Professional Training Report on “**WEATHER FORECAST USING WEB DEVELOPMENT**” done by me under the guidance **Mr. P. SARAVANAN,.M.E.,(Ph.D)** at Sathyabama University is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

The satisfaction and elation that accompany the successful completion of any task would be incomplete without the mention of the people who have made it possible. It is our great privilege to express our gratitude and respect to all those who have guided me during the course of my Professional Training.

I convey my thanks to **Dr.T.SASIKALA,M.E.,Ph.D.**, Dean, School of computing and **Dr. R. SUBHASHINI,M.E.,Ph.D.**, Head of the Department, Department of Information Technology for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Mr.P.SARAVANAN.,M.E.,(Ph.D)** for his valuable guidance ,suggestions and constant Encouragement which paved my way to successful completion of my project.

I convey my thanks to the management and all the teaching staffs and non teaching staffs of the Information Technology department who were helpful in many ways for the completion of the project.

Also, I thank my **Parents** for supporting me in the completion of the Professional Training.

TRAINING CERTIFICATE

AKON ONLINE SERVICES PRIVATE LIMITED

Quality Never Goes out of style....
GST No 27AATCA65048125
UDYAM-MH-33-0007682
ceo@akonzon.com

Bungalow no 6, Shrushti
Complex Hiranandani
Estate CB Road Patlipada
Thane West (400607)
Maharashtra India

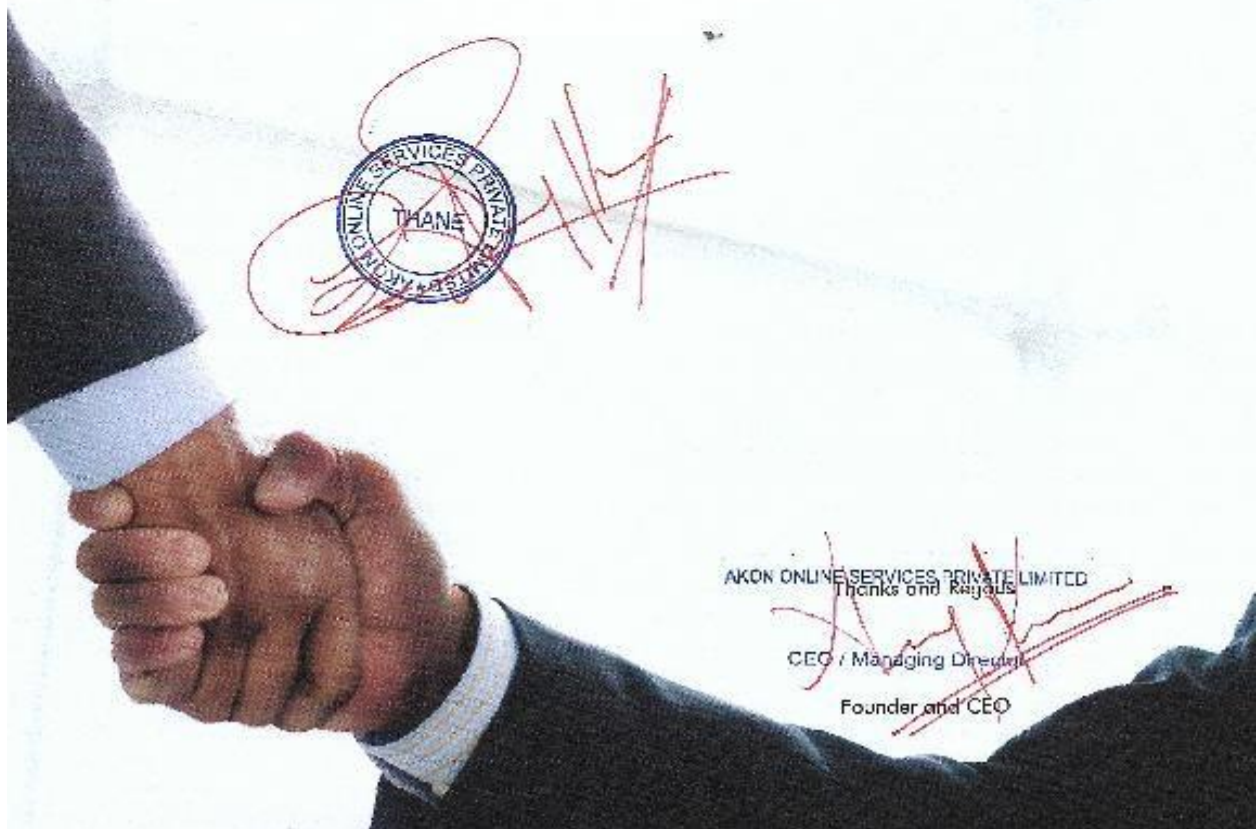
Date : 26-09-2020

TO WHOM IT MAY CONCERN

This is to certify that **MR. HARSH GUPTA** has successfully completed his internship during his tenure from 25-07-2020 to 25-09-2020 in our organisation as **DATA ANALYST**.

During his tenure we found him to be punctual, hardworking, inquisitive and dedicated to the project allotted to him. He has duly completed the project and assignments assign to him to our satisfaction.

We wish him all the best for his future endeavours.



ABSTRACT

Weather forecast is always needed by a person on a daily basis, for example you would already be aware of a rainstorm hours before it would arrive. So , to display weather forecast in my website , I would connect server data ie an API (An application program interface (**API**) is a set of routines, protocols, and tools for building software applications. Basically, an **API** specifies how software components should interact. Additionally, **APIs** are used when programming graphical user interface (GUI) components.) through which I would accessing its data and connect it with my website though my API key. I would use open weather API for this project through which I will access all its real time weather data and display it on my web page using java script ,css , and HTML.

TABLE OF CONTENTS

CHAPTERNO.	TITLE	PAGE.NO
1.	INTRODUCTION	
	1.1 OUTLINE OF THE PROJECT	2
	1.2 LITERATUE REVIEW	2
	1.3 PROBLEM STATEMENT	3
	1.4 OBJECTIVES	3
2	AIM & SCOPE	
	2.1 REQUIREMENTS	4
	2.1.1 HARDWARE REQUIREMENTS	4
	2.1.2 SOFTWARE REQUIREMENTS	4
	2.2 ROLE OF BRACKETTS	4
	2.2.1 LANGUAGES USED	5
	2.2.1 HTML	5
	2.2.2 CSS	5
	2.2.3 JAVA SCRIPT	6
	2.3 OPEN WEATHER API	6
	2.3.1 CURRENT WEATHER DATA	7
	2.3.2 FORECAST	7
	2.3.3 SEARCHING	7
	2.3.4 MAPS	7
3	METHODS & MATERIAL USED	8
	3.1 DESIGNING OF TEXT EDITOR	8
	3.2CONNECTION OF API	11
4	RESULT & DISCUSSION	14

5	CONCLUSION AND FUTURE WORK	17
	5.1SUMMARY	17
	5.2FUTURE WORK	17
	APPENDIX	
	A.SOURCE CODE	18
	B.SCREENSHOTS	24
6.	REFERENCES	26

LIST OF FIGURES

FIGURE NAME	PAGE NO
Outline of the Webpage	8
Styling of the Webpage	9
API connection through Java script	10
Open weather API	11
Sign up page	11
API key	12
Weather application	14
Chennai	15
Delhi	15
Opening a new file	24
Editing option	24
View option	25
Debug	25

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANSION

HTML

Hyper Text Mark Up Language

CSS

Cascading Style Sheet

API

Application Program Interface

ECMA

European Computer
Manufacturers Association

CHAPTER 1

INTRODUCTION

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delimited by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

API

In computer programming, an **application programming interface (API)** is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication

among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

An API may be for a web-based system, operating system, database system, computer hardware, or software library.

An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Windows API and ASPI are examples of different forms of APIs. Documentation for the API usually is provided to facilitate usage and implementation.

OUTLINE OF THE PROJECT

Throughout the human history, people were keen to know about the weather, its parameters and its impacts on their daily lives. By the virtue the technological advancement, in this era, the information about the weather lies in our hands (through mobile phones or websites). We can now make ourselves aware of not only our location's temperature, humidity etc. but also any part of the world. In this project, we will learn how to make a weather application using pure JavaScript. We will also be familiarizing ourselves with the JSON extraction during this process using open weather API.

As a web developer, grabbing data from API's is something you are going to do often. Fetching weather data is a perfect way to get your feet wet. In this project we are going to use the browsers built in fetch with JavaScript to grab data from Open Weather Map's API.

LITERATURE REVIEW

JavaScript (*/ˈdʒɑːvəˌskɪpt/*), often abbreviated as **JS**, is a high-level, interpreted scripting language that conforms to the ECMA Script specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

PROBLEM STATEMENT:

To make a web page using HTML, CSS & JAVA SCRIPT and connect the web page application to a API from open weather API and display the data.

OBJECTIVES

We need to first make the outline of the project using HTML and CSS to create the front end of the web page .

CHAPTER 2

AIM & SCOPE OF WEB DEVELOPMENT

REQUIREMENTS:

HARDWARE REQUIREMENTS:

1. Any processor with minimum GPU memory

SOFTWARE REQUIREMENTS:

1. BRAKETTS EDITOR
2. OPEN WEATHER API

1. ROLE OF BRAKETTS

Brackets is a source code editor with a primary focus on web development. Created by Adobe Systems, it is free and open-source software licensed under the MIT License, and is currently maintained on GitHub by Adobe and other open-source developers. It is written in JavaScript, HTML and CSS. Brackets is cross-platform, available for macOS, Windows, and most Linux distributions. The main purpose of brackets is its live HTML, CSS and JavaScript editing functionality.

On November 4, 2014, Adobe announced the first (1.0) release of Brackets. The update introduced new features such as custom shortcut key combinations and more accurate JavaScript hinting. Brackets has a major focus on development in JavaScript, CSS and HTML. With release of version 1.0, Adobe announced a feature that extracts design information from a PSD file for convenience of coding in CSS. As of June 28, 2016, the feature is officially discontinued, due to low usage. However, Extract is still available via Photoshop and Dreamweaver, both of which are part of their paid service, Adobe Creative Cloud. The latest version release of Brackets is 1.14.

LANGUAGES USED

- **HTML**
- **CSS**
- **JAVA SCRIPT**

HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.

JAVA SCRIPT

JavaScript often abbreviated as **JS**, is a high-level, interpreted scripting language that conforms to the ECMA Script specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web

applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

2. OPEN WEATHER API

Open Weather Map is an online service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications. For data sources, it utilizes meteorological broadcast services, raw data from airport weather stations, raw data from radar stations, and raw data from other official weather stations. All data is processed by Open Weather Map in a way that it attempts to provide accurate online weather forecast data and weather maps, such as those for clouds or precipitation. Beyond that, the service is focused on the social aspect by involving weather station owners in connecting to the service and thereby increasing weather data accuracy. The ideology is inspired by Open Street Map and Wikipedia that make information free and available for everybody. It uses Open Street Map for display of weather maps.

Open Weather Map provides an API with JSON, XML and HTML endpoints and a limited free usage tier. Making more than 60 calls per minute requires a paid subscription starting at USD 40 per month.

Access to historical data requires a subscription starting at US\$150 per month.

Users can request current weather information, extended forecasts and graphical maps (showing cloud cover, wind speed, pressure and precipitation).

Current weather data

Current data is refreshed every ten minutes; it can be searched by city or by geographic coordinates on Earth.

Forecasts

Weather forecasts can be searched by city or by coordinates. Three- hourly forecasts are available for up to 5 days, while daily forecasts are available for up to 16 days.

Searching

The Open Weather Map geocoding system allows users to select cities by name, country, zip-code or geographic coordinates. It is possible to search by part of city name. To make searching result more accurate city name and country should be divided by comma.

Maps

Open Weather Map service provides lots of weather maps including Precipitation, Clouds, Pressure, Temperature, Wind and many others. Maps can be connected to mobile applications and web sites. Weather maps can be connected as layers to the wide range of maps including Direct tiles, WMS, Open Layers, Leaflet, Google maps, and Yandex maps.

CHAPTER 3

METHODS& MATERIAL USED

MATERIAL USED:

- Bracketts editor
- API key
- Google Chrome

DESIGNING OF TEXT EDITOR:

The project will stick to the basic functionalities expected of a simple text editor – which includes the ability to – write something on the notepad, save it and open and modify it whenever required. For the purpose of this tutorial we will design the braketts editor with html and css code to create the front end part of the web page.

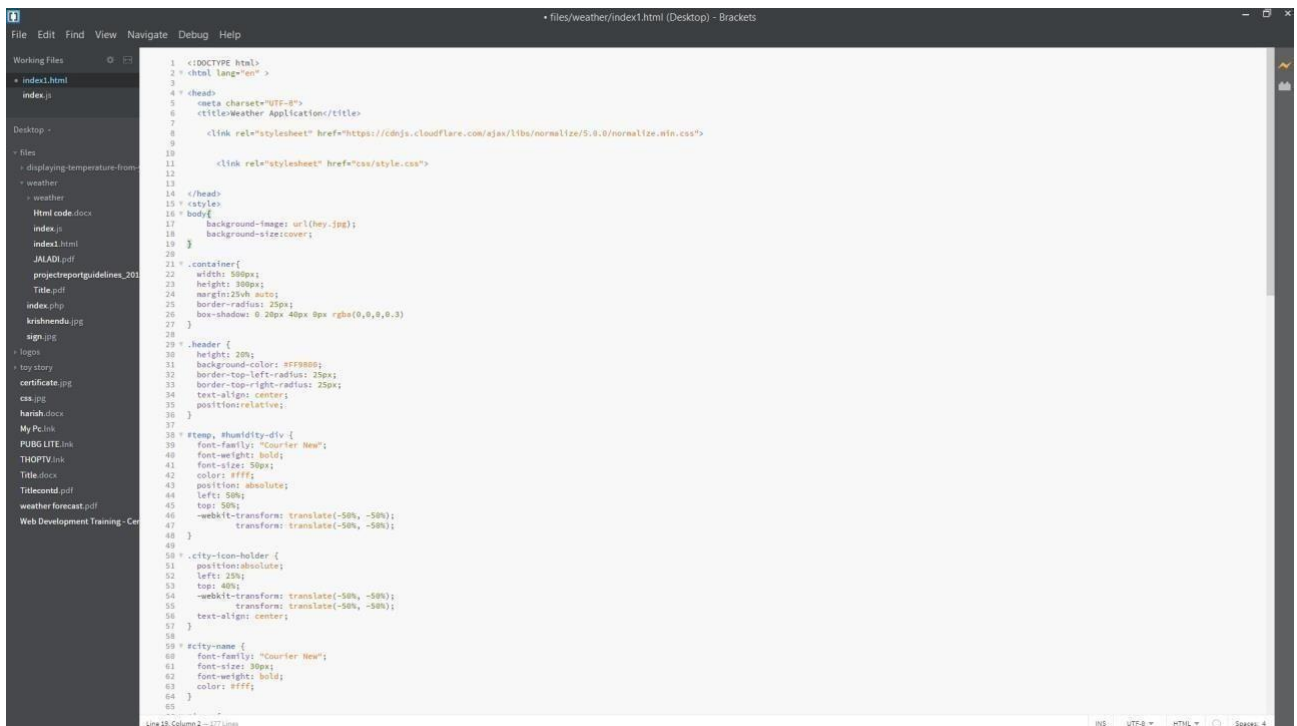


Fig. 3.1 OUTLINE OF THE WEB PAGE

At this point we've already looked at CSS fundamentals, how to style text, and how to style and manipulate the boxes that your content sits inside. Now it's time to look at how to place your boxes in the right place in relation to the viewport, and one another. We have covered the necessary prerequisites so you can now dive deep into CSS layout, looking at different display settings, traditional layout methods involving float and positioning, and new fangled layout tools like flexbox.

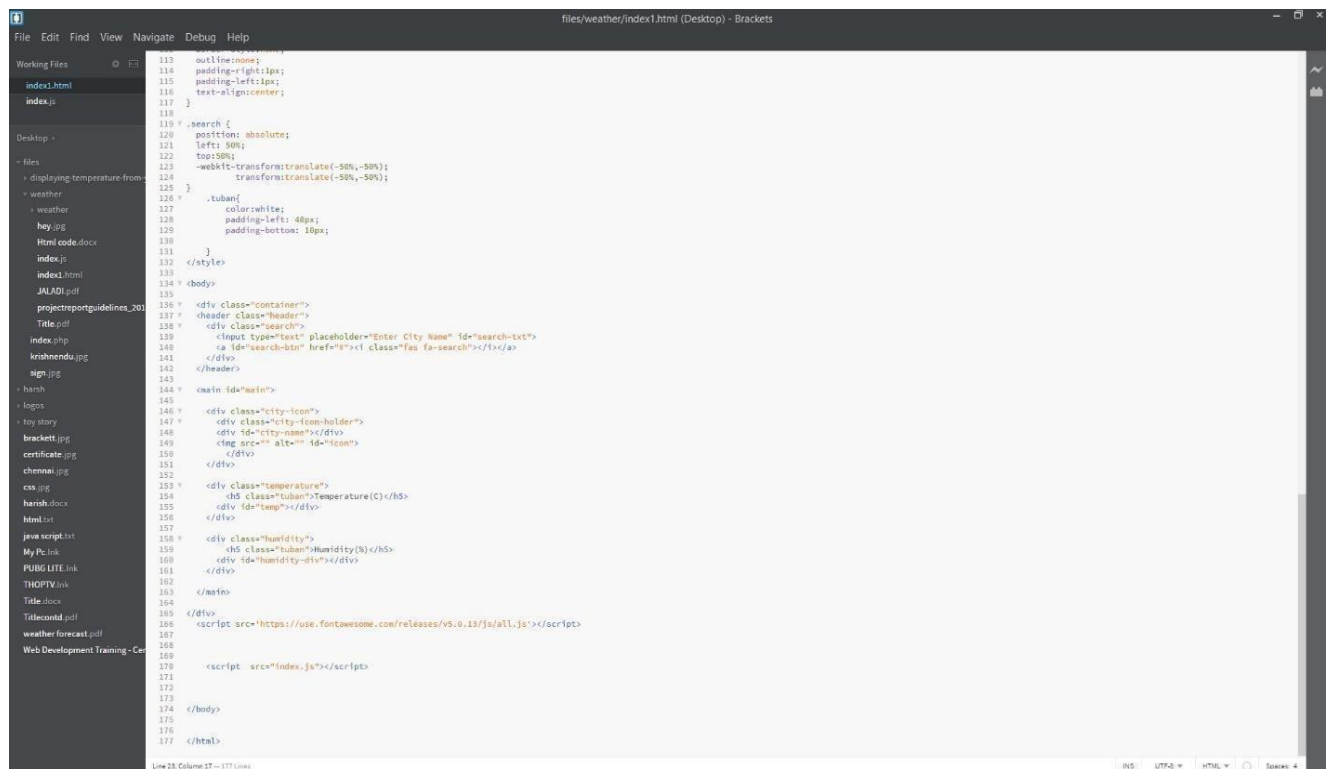


FIG.3.2 STYLING OF THE WEB PAGE

This module looks at styling boxes, one of the fundamental steps towards laying out a web page. In this module we recap the box model, then look at controlling box layouts by setting margins, borders, and padding, custom background colors, images and other features, and fancy features such as drop shadows and filters on boxes.

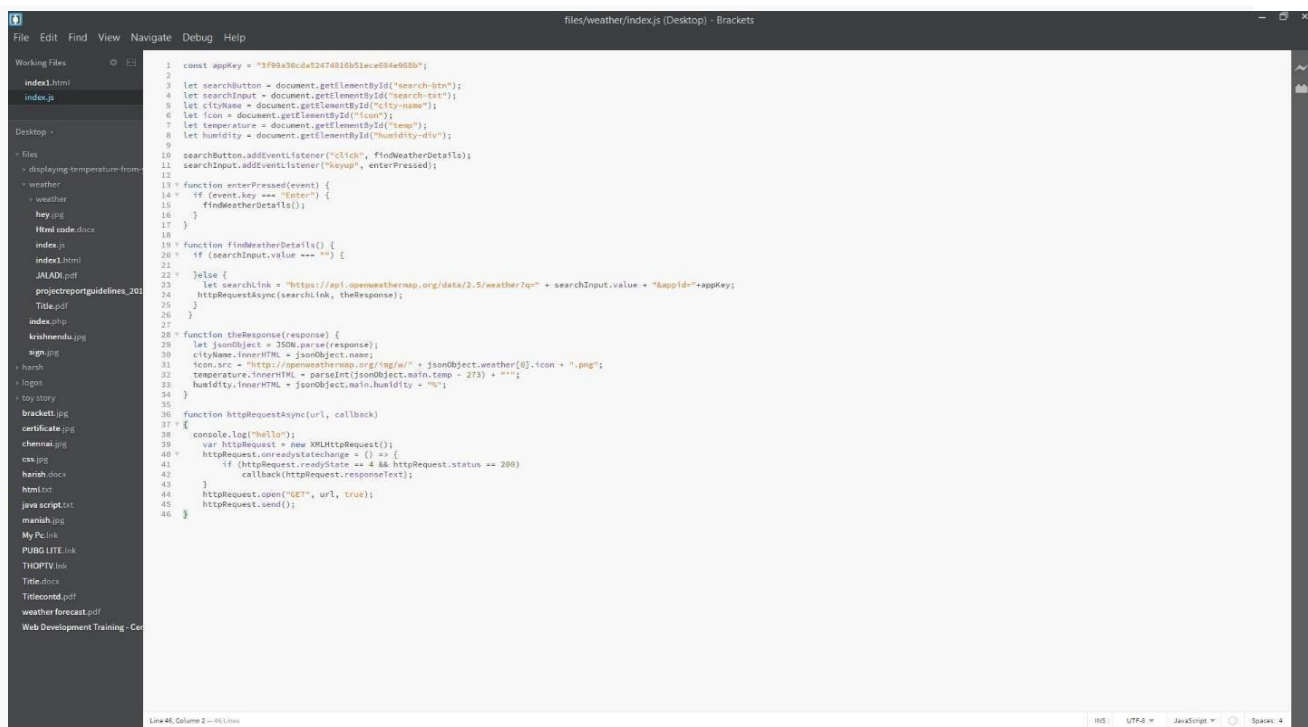


FIG.3.3 API CONNECTION THROUGH JAVA SCRIPT

API keys aren't as secure as authentication tokens but they identify the application or project that's calling an API. They are generated on the project making the call, and you can restrict their use to an environment such as an IP address range, or an Android or iOS app.

By identifying the calling project, you can use API keys to associate usage information with that project. API keys allow the Extensible Service Proxy (ESP) to reject calls from projects that haven't been granted access or enabled in the API.

1. **Weather data is open:** current weather, forecasts, maps with precipitations, wind, clouds, data from weather stations are available through APIs, maps and other products.
2. **Coverage is global:** weather data is available for any geographic location.
3. **Weather model:** own model of weather forecast calculation, WRF model for regions + global models.
4. **Advanced technologies for a competitive price:** due to Big Data technology costs of production and support are cheap, a price for a user is affordable.

CONNECTION OF API:

Since we use open weather API ,we need to sign up on open weather map website ie, www.openweathermap.org/api

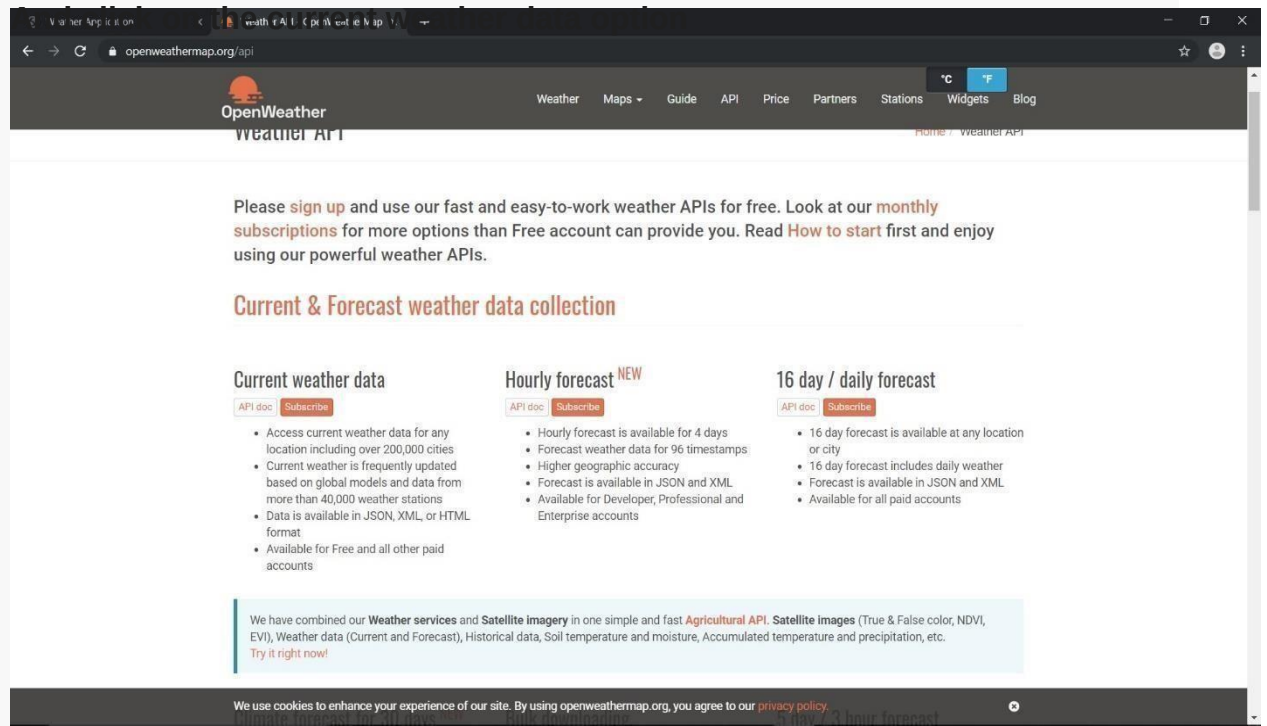


Fig3.4 OPEN WEATHER API WEBSITE

After opening the website enter your sign up details to log in.

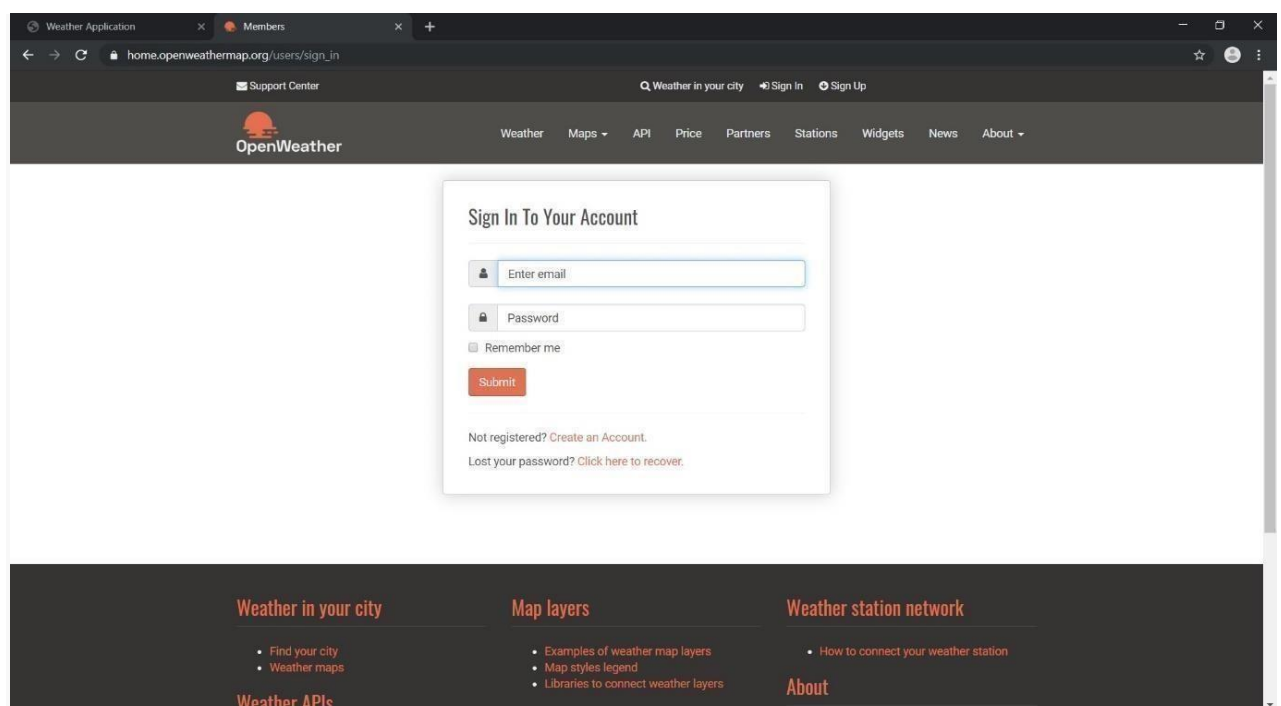


Fig 3.5 SIGN UP PAGE

Api recommend making calls to the API no more than one time every 10 minutes for one location (city / coordinates / zip-code). This is due to the fact that weather data in our system is updated no more than one time every 10 minutes.

The server name is **api.openweathermap.org**. Please, never use the IP address of the server.

Better call API by city ID instead of a city name, city coordinates or zip code to get a precise result.

Please, mind that Free and Startup accounts have limited service availability. If you do not receive a response from the API, please, wait at least for 10 min and then repeat your request. We also recommend you to store your previous request.

After you log in go to API keys on the tabular column and then you will find your respective API key.

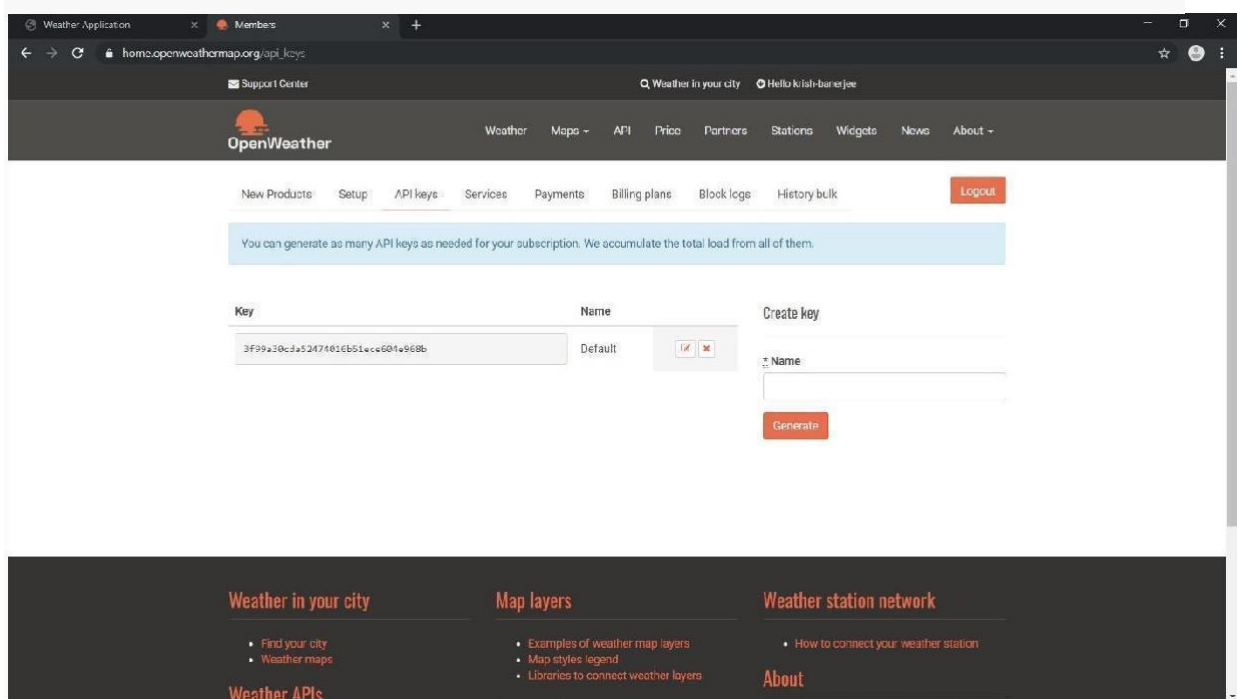


Fig 3.6 API KEY

A confirmation email with your API key and technical instructions will be sent to your email address.

Please note that it takes up to 2 hour to activate your API key. Please use your unique API key {appid=} in each API call to authorize a request from your application and process it appropriately.

Demand for certain products and services varies greatly depending on the weather. For example, users are much more likely to search for information on amusement parks on a hot, sunny day than if it's cold and raining. An amusement park company may want to increase their bids when the weather is nice, but doing so every day would require a lot of manual work. With AdWords scripts, however, it's possible to programmatically fetch weather information and adjust bids in a matter of minutes. This script uses Google Spreadsheets to store the list of campaigns and their associated locations.

A call to the OpenWeatherMap API is made for each location and weather conditions are calculated using some basic rules. If a rule evaluates to true, then a corresponding location bid multiplier is applied to the location targeting for the campaign.

CHAPTER 4

RESULTS & DISCUSSION

Before running the code enter your API key in the java script file in the const key area ,this will use your unique key to access all the weather data you want to use in your website and after running the code you get the front end part of the website.

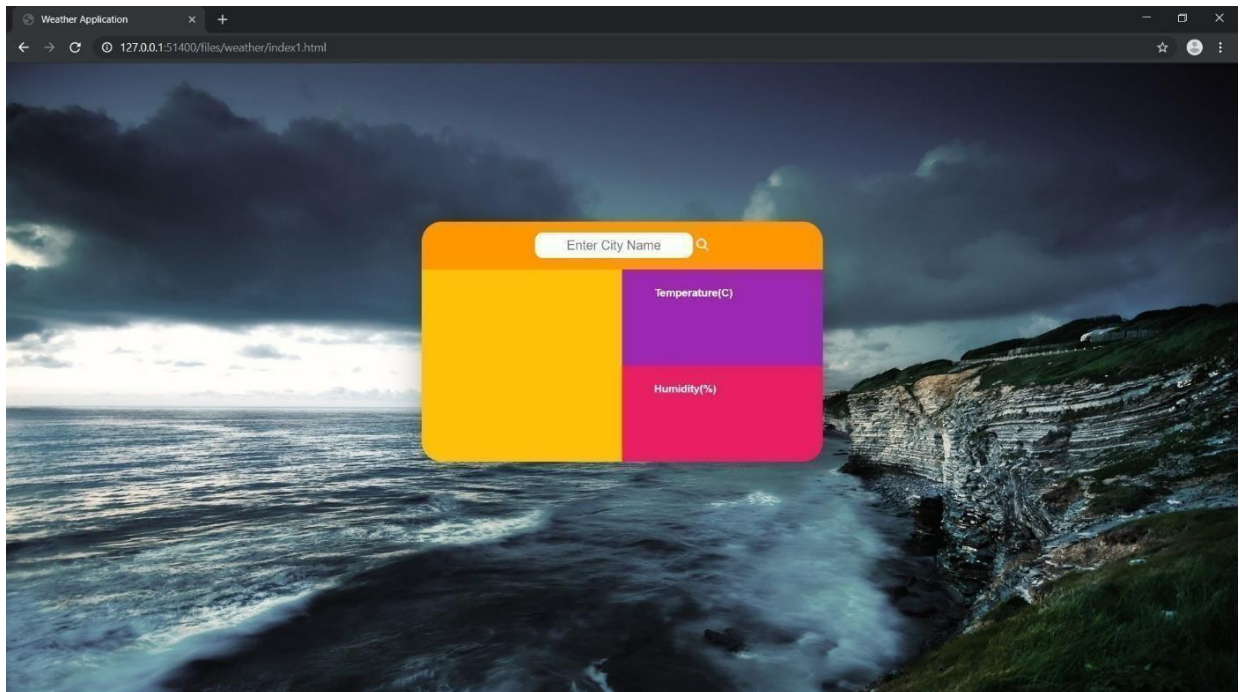


Fig.4.1 WEATHER APPLICATION

Now enter any location you want to see the weather of, For example I want to see the weather of Chennai so, I will enter Chennai on the search tab which will in turn access the weather data from the open weather API and show us the connected data

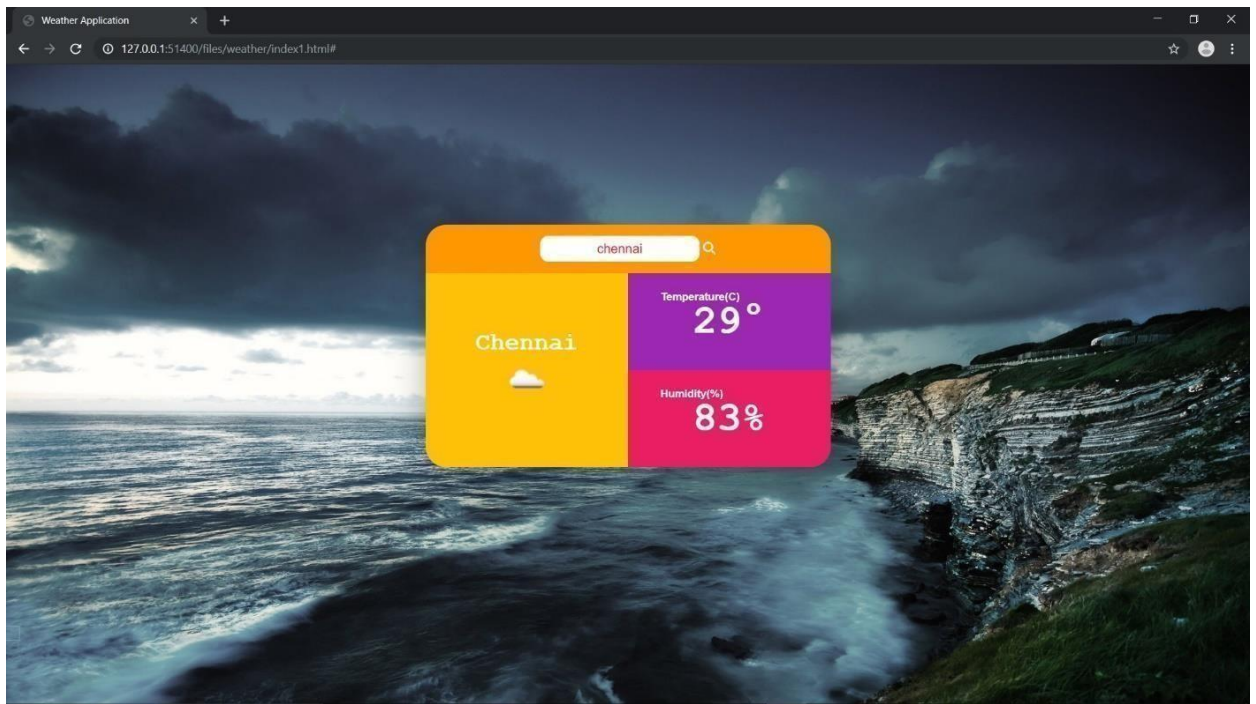


Fig 4.2 CHENNAI

Similarly, lets see the weather of Delhi.

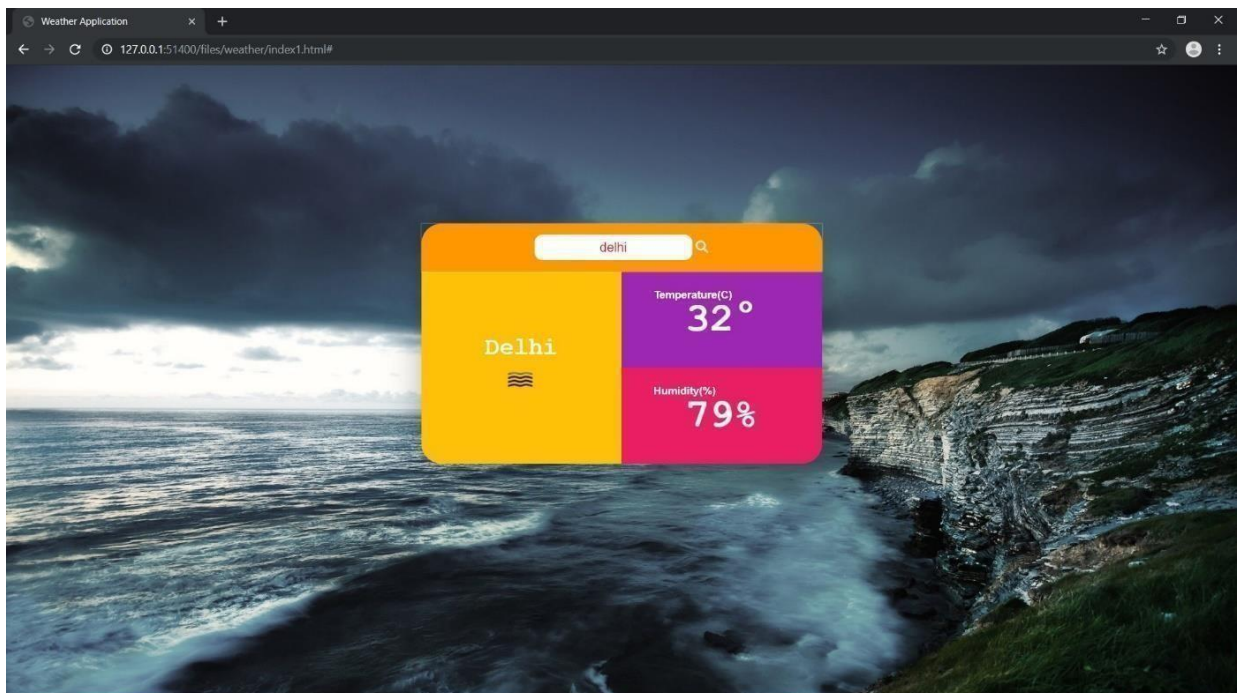


Fig 4.3 DELHI

DISCUSSION

The process of updating Weather API to new .NET Core 2.0 and Entity Framework Core 2.0 has already started. In addition, as soon as EF Core supports spatial data

types, they should be implemented to Weather API. As EnerKey service are sold to other countries weather information should be collected from those countries. Most likely, the first new country to be added is Sweden followed by other Nordic countries after that. There is also need for weather forecasts, so that the customers and Enegia's internal services can react to rapid weather changes. Cold weather will increase electricity consumption and creates spikes to energy consumption. Implementing Weather API to other services can be done quickly using OpenAPI specification (OAS). With generated JSON file programs such as NSwagStudio can generate clients instantly.

Developer should pursue clean and maintainable code. Over engineering six-layer logic when two is more than enough is counter intuitive. Planning good test cases will save developers time in the long run. It is important to priorities testing cases and at least cover every basic case. Writing tests that will not test anything worth testing and are hard to maintain, will take time from the development process. 24 Code reviews are good and inexpensive way to improve quality. Every developer has their unique way of looking challenges and can catch issues that others have missed. Asking for guidance and opinions will often lead to better result than working solo.

The process of predicting weather patterns is a very complicated science. It requires the need to analyze and decode massive data sets gathered from thousands of sensors and weather satellites every day.

Identifying patterns in collected data to predict the future is a very strenuous task. For best results, it also needs to be done in real-time.

But like any kind of forecast, weather forecasting is something of an educated guess. Since we cannot control the weather, the best meteorologists can do, is to use past and present data and patterns to attempt to predict the future.

CHAPTER 5

CONCLUSION & FUTURE WORK

SUMMARY:

The intention of the project is to just make a model of a weather forecast using web development and code it in a simple way so that one can know the weather of a location on his/her website just by accessing a API key. This application would be a part of the development platform of the technology.

FUTURE WORK:

In future projects I'll improve the design of my program by giving stylish themes and fonts.

Also , would compare the API data with a weather predictor which predicts weather with the data of weather in that location in the past 5 years.

CONCLUSION:

Run the program , the google chrome window opens with your designed web page. Then enter the location of the place for which you want to see its weather. Now your web page shows the respective temperature and humidity of that particular area.

Hence, we have successfully created a website which displays the weather of any location entered at that particular time.

APPENDIX

A.SOURCE CODE:

HTML&CSS CODE

```
<!DOCTYPE html>
```

```
<html lang="en" >
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Weather Application</title>
```

```
<linkrel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
```

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

```
</head>
```

```
<style> body{ background-image:
```

```
url(hey.jpg); background-
```

```
size:cover;
```

```
}
```

```
.container{ width: 500px; height: 300px; margin:25vh
```

```
auto; border-radius: 25px; box-shadow: 0 20px 40px
```

```
0px rgba(0,0,0,0.3)
```

```
}
```

```

.header { height: 20%; background-color: #FF9800; border-top-left-radius: 25px;
border-top-right-radius: 25px; text-align: center; position: relative;
}
#temp, #humidity-div {

font-family: "Courier New"; font-weight: bold;
font-size: 50px; color: #fff; position: absolute; left: 50%; top: 50%;

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

}
.city-icon-holder { position: absolute; left: 25%; top:
40%;

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

transform: translate(-50%, -50%); text-align: center;
}
#city-name {

font-family: "Courier New"; font-size: 30px; font-weight:
bold; color: #fff;
}
#icon {

width: 50%;

}
#main{

width: 100%;

height: 100%; position: relative;
}

```

```

.city-icon    { height: 80%; width: 50%;

border-bottom-left-radius: 25px; background-color: #FFC107;
}
.temperature { position: absolute; left: 50%;
top:0%; height: 40%; width: 50%;

background-color: #9C27B0;

}
.humidity { height: 40%; width: 50%; position:absolute; left:50%;
top:40%; border-bottom-right-radius: 25px; background-color:
#E91E63;
}
#search-btn {
width: 40px; height:40px; color: #eee;
}
    #search-txt { color: red; height:30px; border-radius: 10px; border-style:none;
    outline:none; padding-right:1px; padding-left:1px; text-align:center;
}
.search {

position: absolute; left:    50%;
top:50%;

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

.tuban{

color:white; padding-left: 40px; padding-bottom:
10px;
}

</style>

```

```
<body>

<div class="container">

<header class="header">

<div class="search">

<input type="text" placeholder="Enter City Name" id="search-txt">

<a id="search-btn" href="#"><i class="fas fa-search"></i></a>

</div>

</header>
<main id="main">

<div class="city-icon">

<div class="city-icon-holder">

<div id="city-name"></div>

<img src="" alt="" id="icon">

</div>

</div>

<div class="temperature">

<h5 class="tuban">Temperature(C)</h5>

<div id="temp"></div>
```

```
</div>
<div class="humidity">
<h5 class="tuban">Humidity(%)</h5>

<div id="humidity-div"></div>

</div>
</main>
</div>

<script src='https://use.fontawesome.com/releases/v5.0.13/js/all.js'></script>

<script src="index.js"></script>
</body>
</html>
```

JAVA SCRIPT CODE

```
const appKey = "3f99a30cda52474016b51ece604e968b";

let searchButton = document.getElementById("search-btn"); let searchInput =
document.getElementById("search-txt"); let cityName =
document.getElementById("cityname"); let icon =
document.getElementById("icon");

let temperature = document.getElementById("temp");

let humidity = document.getElementById("humidity-div");

searchButton.addEventListener("click", findWeatherDetails);
searchInput.addEventListener("keyup", enterPressed);
function enterPressed(event) { if (event.key === "Enter") { findWeatherDetails();
```

```

}
}
function findWeatherDetails() { if (searchInput.value === "") {
}else {

    let searchLink = "https://api.openweathermap.org/data/2.5/weather?q=" +
    searchInput.value
    + "&appid="+appKey; httpRequestAsync(searchLink, theResponse);

}
}
function theResponse(response) {

    let jsonObject = JSON.parse(response); cityName.innerHTML = jsonObject.name;
    icon.src = "http://openweathermap.org/img/w/" + jsonObject.weather[0].icon +
    ".png"; temperature.innerHTML = parseInt(jsonObject.main.temp - 273) + "°";
    humidity.innerHTML = jsonObject.main.humidity + "%";
}
function httpRequestAsync(url, callback)
{
console.log("hello");

var httpRequest = new XMLHttpRequest();
httpRequest.onreadystatechange = () => {

if      (httpRequest.readyState == 4      &&      httpRequest.status == 200)
callback(httpRequest.responseText);
}
httpRequest.open("GET", url, true); httpRequest.send();
}

```

B.SCREENSHOTS

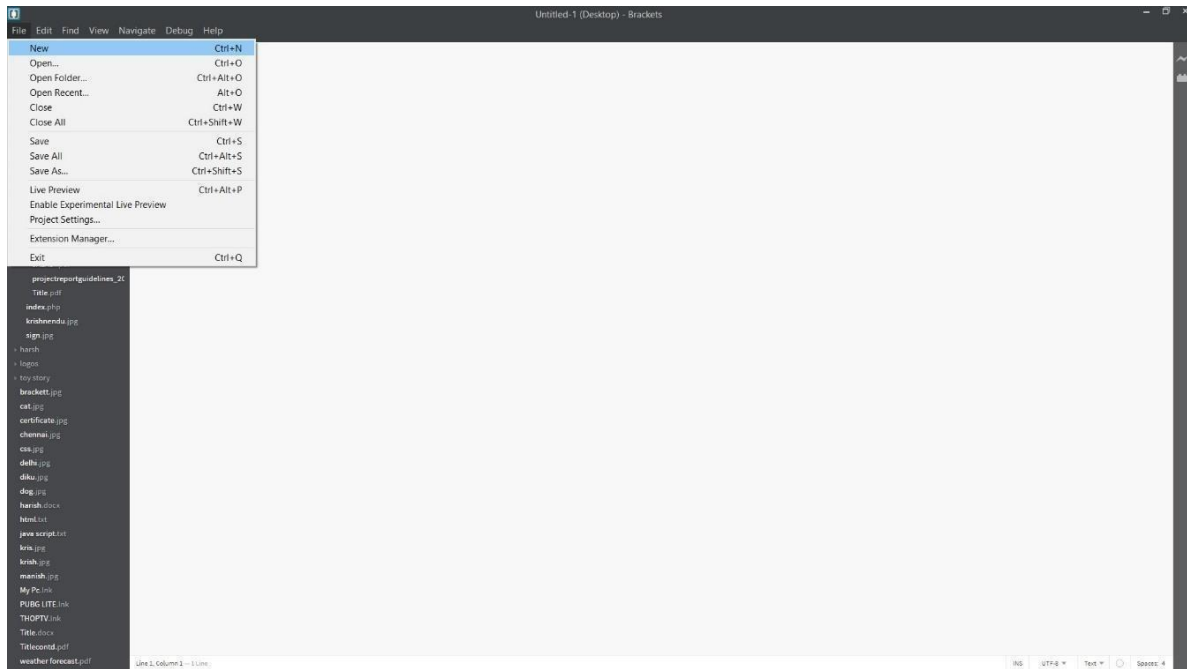


FIG.5.1 OPENING A NEW FILE



FIG.5.2 EDITING OPTION



FIG.5.3 VIEW OPTION

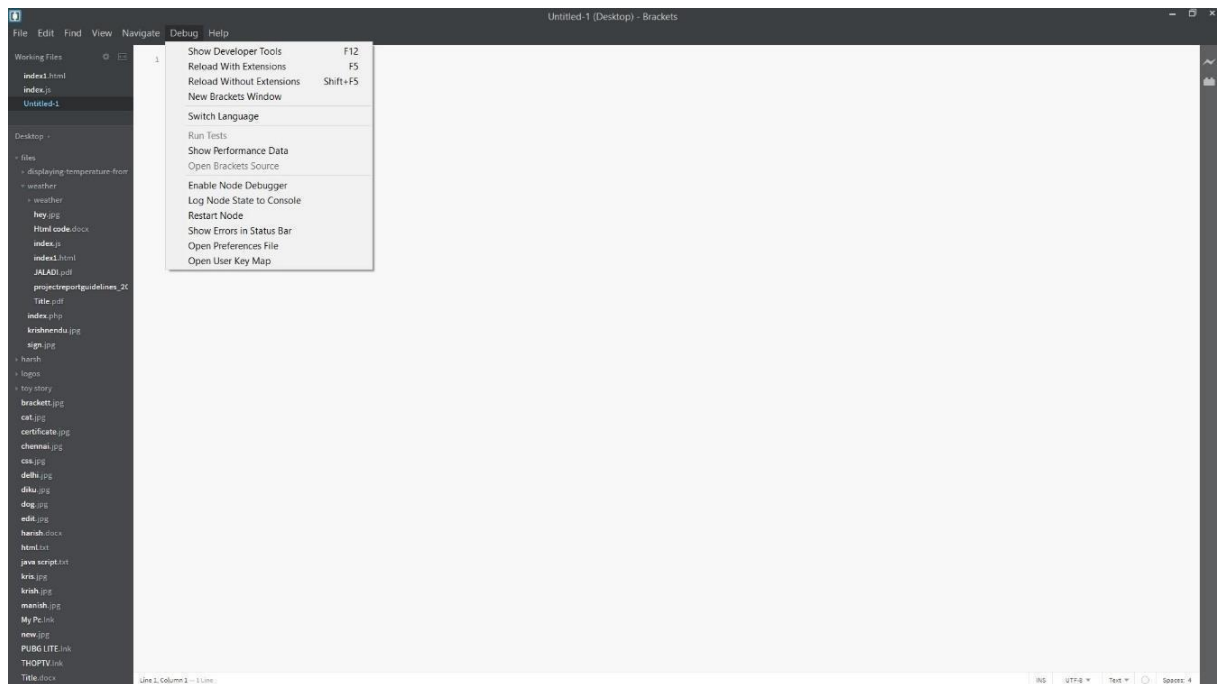


FIG.5.4 DEBUG

REFERENCES

- [1] Pickering, Heydon; Apps For All: Coding Accessible Web Applications(2014), Smashing Magazine GmbH.
- [2] Pickering, Heydon; Inclusive Components(2018): The Book, Inclusive Components
- [3] Pickering, Heydon; Inclusive Design Patterns(2016), Smashing Magazine.
- [4] Sydik, Jeremy; Design Accessible Web Sites(2007), Pragmatic Bookshelf.
- [5] Thatcher, Jim et al; Constructing Accessible Web Sites(2002), Glasshaus.
- [6] Thatcher, Jim et al; Web Accessibility: Web Standards and Regulatory Compliance(2006), friends of ED.