



## Assignment

### INTEGRATION OF API'S IN WEB PAGE

**Name : Salman | Roll no: 2k17/IT/79 |**

**Class: BS-IT Part IV (Morning)**

**Semester: 8th | Assignment by:**

**Dr Kamran Taj Pathan**

# Assignment

**Topic:** Make a web services that is going to utilize any of the services of Profile (LinkedIn), Location (Google Maps), Time, Weather etc.

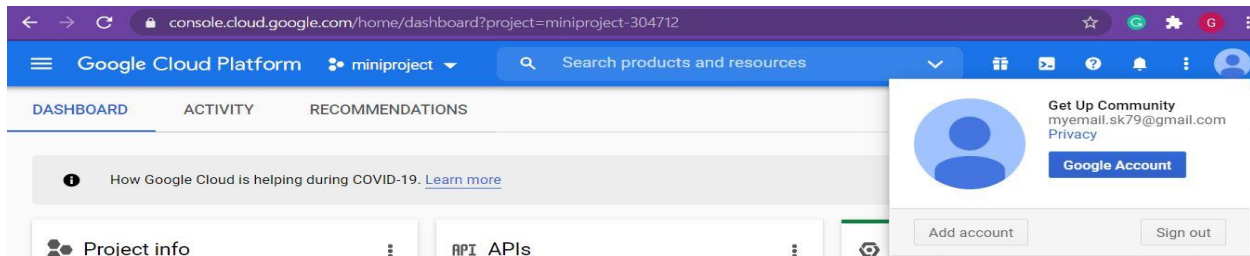
**I Used 3 API's which is (Google maps API, Open Weather API, Google Sign-in API).**

1. In this project I create a web page using html, JavaScript and css. Then integrate some different API's which collect current data and show on my web page.

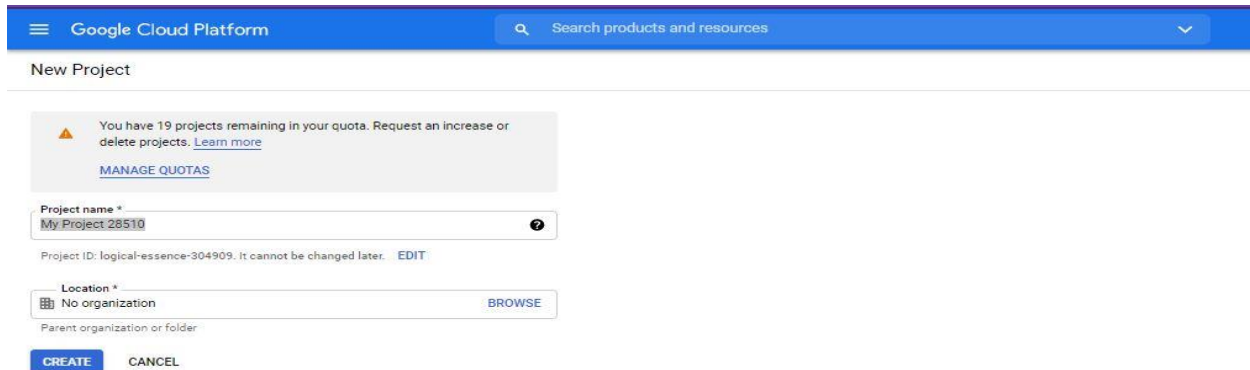
Firstly I used Google Maps API. I go to Google cloud platform and there I sign-in from my account. Below are some steps for integrating Google maps API on your website.

**Steps:**

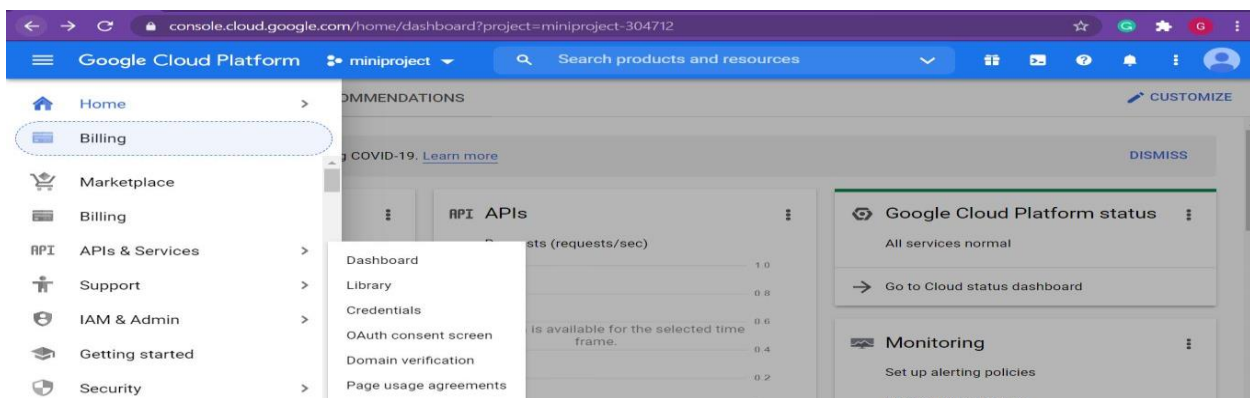
**1. Sign-in in Google Cloud Platform**

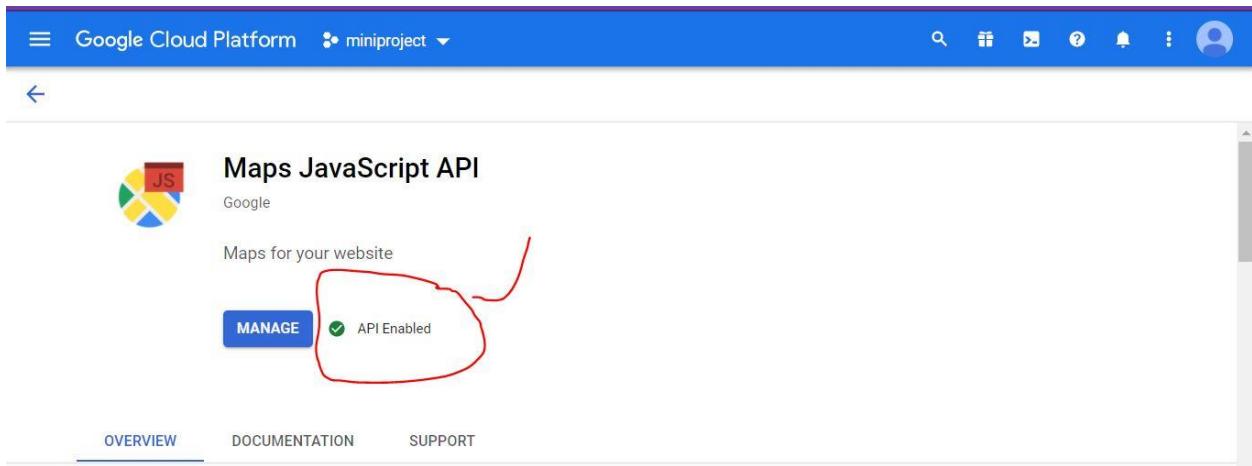


**2. Create new project with suitable name.**

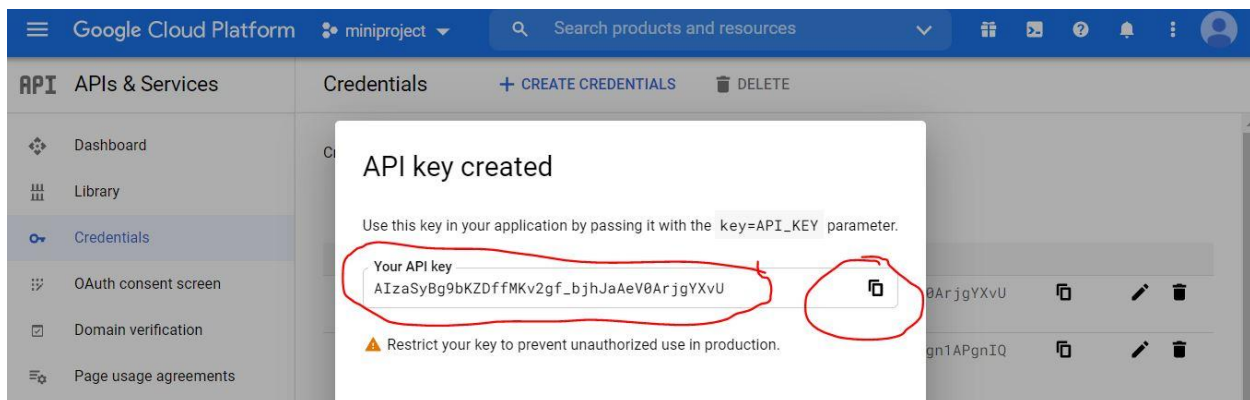
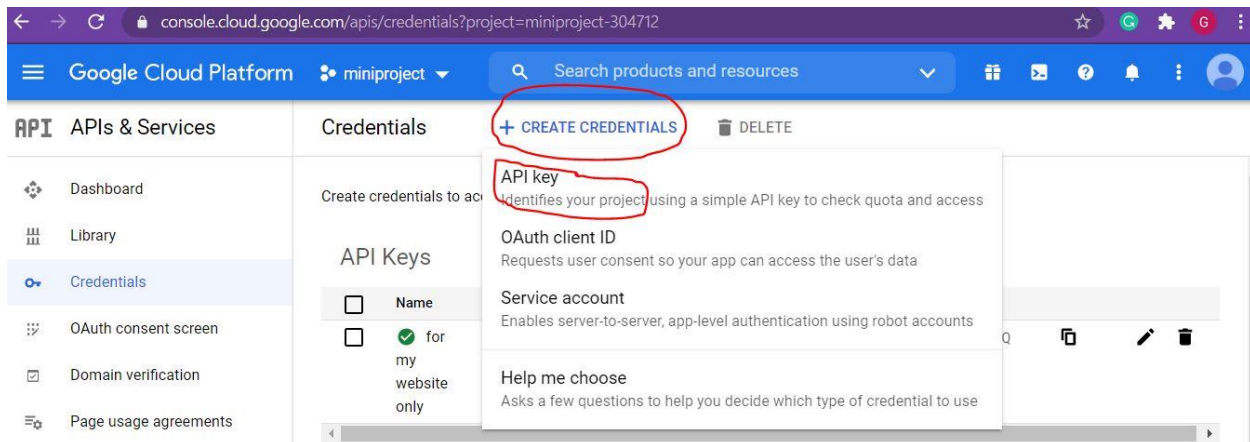


**3. Go to menu then "API's and Service" then "Library" and search for (Google maps API JavaScript) and Enable (Google maps API JavaScript).**





4. Then again go to menu click on “API’s and Service” then” Credentials” create Credentials” Click on “API KEY” create API key And Copy This key And paste in the script Tag link of Google API.



## 5. Put API key in script tag in the Google map API Link.



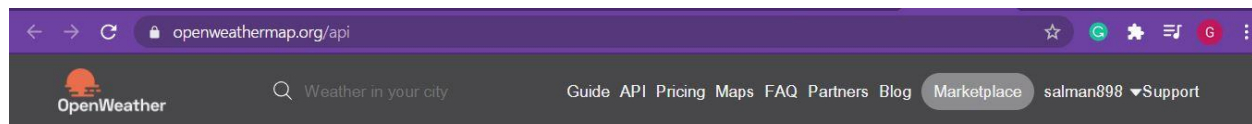
```
File Edit Selection View Go Run Terminal Help
index.html - WeatherApp-main - Visual Studio Code

EXPLORER
  OPEN EDITORS
    X index.html
    JS script.js
    # style.css
  WEATHERAPP-MAIN
    > icons
    index.html
    readme.md

index.html X
  4 <!-- google map script tag -->
  5 <script>
  6   src="https://maps.googleapis.com/maps/api/js?key=Iza5yCvq33M0RmCw3iaVRotNMxAFIOjGvEaUriscallback=initMap&libraries=&v=weekly"
  7   defer
  8 </script>
```

Used another weather API add:

1. Go to (Openweathermap.org) site then sign-up here and then generate API Key And copy this key and put in your code javascript section.



### Current & Forecast weather data collection

#### Select Current weather API.

<div><h4>Current Weather Data</h4><p>API doc <a href="#">Subscribe</a></p><ul style="list-style-type: none"><li>• Access current weather data for any location including over 200,000 cities</li><li>• We collect and process weather data from different sources such as global and local weather models, satellites, radars and vast network of weather stations</li><li>• JSON, XML, and HTML formats</li><li>• Available for both Free and paid subscriptions</li></ul></div>	<div><h4>Hourly Forecast 4 days</h4><p>API doc <a href="#">Subscribe</a></p><ul style="list-style-type: none"><li>• Hourly forecast is available for 4 days</li><li>• Forecast weather data for 96 timestamps</li><li>• Higher geographic accuracy</li><li>• JSON and XML formats</li><li>• Available for Developer, Professional and Enterprise accounts</li></ul></div>	<div><h4>One Call API</h4><p>API doc <a href="#">Subscribe</a></p><ul style="list-style-type: none"><li>• Make one API call and get current, forecast and historical weather data</li><li>• <b>Minute forecast</b> for 1 hour</li><li>• <b>Hourly forecast</b> for 48 hours</li><li>• <b>Daily forecast</b> for 7 days</li><li>• <b>Historical data</b> for 5 previous days</li><li>• <b>National weather alerts</b></li><li>• JSON format</li><li>• Available for both Free and paid subscriptions</li></ul></div>
---	---	---

Generate and copy API KEY.

<div><h4>API call</h4><pre>api.openweathermap.org/data/2.5/weather?q={city name}&amp;appid={API key}</pre><pre>api.openweathermap.org/data/2.5/weather?q={city name},{state code}&amp;appid={API key}</pre></div>	<div><h4>Weather f</h4><p>JS( XM List Mir API Other feal For Uni ...</p></div>
---	--

## Weather API key Copy.

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

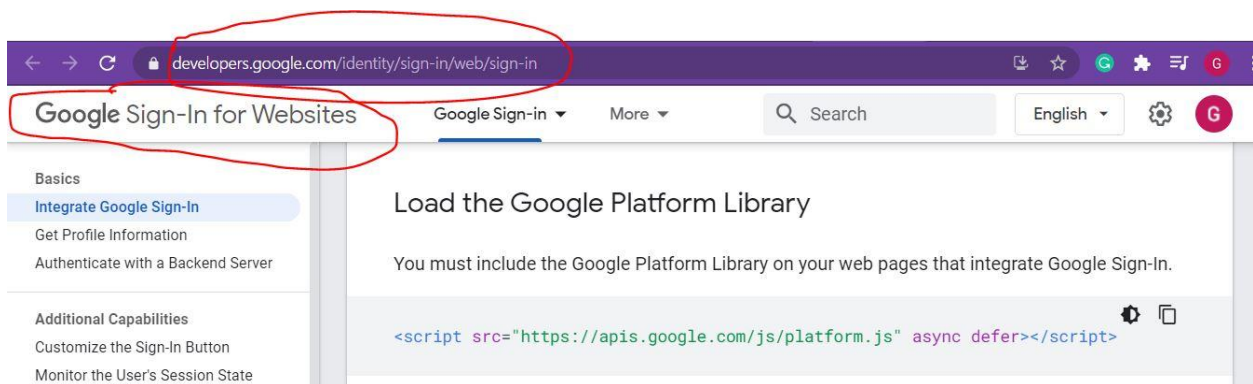
Key	Name	Create key
2a155e01b71eb2afe4c5a1494cf26cc1	Default	<div>API key name</div> <div>Generate</div>

## Paste API key in JavaScript Tag in the Link.

```
const getWeather=async (city)=>
{
  try{
    const response= await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=dab3af44de7d24ae7ff86549334e45bd`,
    {mode: 'cors'}
  );
```

## 3<sup>rd</sup> API which I used in the page is “Google Sign-In”:

Go to (<https://developers.google.com/identity/sign-in/web/sign-in>) and copy the Link for google Sign-in function.



## Paste LINKs in your Script Tag section.

```
<!-- google sign-in script tag -->
<script src="https://apis.google.com/js/platform.js" async defer></script>
<meta name="google-signin-client_id" content="107362585837-tviaam6oc08r8gguidh99cj7r51685k5.apps.googleusercontent.com">
</head>
```



## Project Code

### index.html Page Code:

```
<!DOCTYPE html>
<html>
  <title>Weather app</title>
  <head>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- google map scrip tag -->
    <script
      src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCvq33M0RmCwJiaVRotNMxAFI
OjGvEaUrI&callback=initMap&libraries=&v=weekly"
      defer
    ></script>
    <!-- google sign-in script tag -->
    <script src="https://apis.google.com/js/platform.js" async defer></script>
    <meta name="google-signin-client_id" content="107362585837-
tviaam6oc08r8gguidh99cj7r5l685k5.apps.googleusercontent.com">
  </head>

  <body>
    <!-- from here google maps api start -->
    <h1>GOOGLE MAP API</h1>
    <hr>
    <br/>
    <div id="map1"></div>

    <!-- from here google maps api End -->
    <h1>Weather API</h1>
    <hr>
    <form id="search-form">
      <input type="search"
        placeholder="Enter City Name"
        id="search-input"
        required
        autocomplete="off"
      />
      <br>
    </br>
    <button id="search-button">Search</button>

  </form>

  <main id="app-container">
    <div id="location">
      <p>-----</p>
    </div>
    <div id="temp">
```

```

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

    <div id="climate">
        <p>-----</p>
    </div>

</main>
<!-- from here Google sign -->
<h1>Sign-in Api</h1>
<hr>
<br/>
<div class="g-signin2" data-onsuccess="onSignIn"></div>

<script src="script.js">

</script>
</body>

</html>

```

## **script.js Page Code:**

```

// from here maps javascript start
let map;
function initMap() {
    map = new google.maps.Map(document.getElementById("map1"), {
        center: { lat: 25.3960, lng: 68.3578 },
        zoom: 8,
        mapTypeId: "terrain"
    });
    const marker = new google.maps.Marker({
        position: { lat: 25.3960, lng: 68.3578 },
        map: map,
        label: "A",
        title: " New Delhi",
        draggable: false,
        animation: google.maps.Animation.DROP,
        icon: "map.png"
    });

    const infoWindow = new google.maps.InfoWindow({
        content: "<p>This is an info window</p>"
    });
    infoWindow.open(map, marker);
}
// from here maps javascript End

```

```

// weather api javascript
let loc =document.getElementById("location");
let tempicon=document.getElementById("temp-icon");
let tempvalue=document.getElementById("temp-value");
let climate =document.getElementById("climate");
let iconfile;
const searchInput=document.getElementById("search-input");
const searchButton=document.getElementById("search-button");

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

e.preventDefault();
getWeather(searchInput.value);
searchInput.value='';

});

const getWeather=async (city)=>
{
    try{

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

            {mode: 'cors'}
        );

        const weatherData= await response.json();
        console.log(weatherData);
        const{name}=weatherData;
        const{feels_like}=weatherData.main;
        const{id,main}=weatherData.weather[0];
        loc.textContent=name;
        climate.textContent=main;
        tempvalue.textContent=Math.round(feels_like-273);
        if(id<300 && id>200)
        {
            tempicon.src="./icons/thunderstorm.svg"
        }
        else if(id<400 && id>300)
        {
            tempicon.src="./icons/cloud-solid.svg"
        }
        else if(id<600&& id>500)
        {
            tempicon.src="./icons/rain.svg"
        }
    }
}

```



```

        else if(id<700 && id>600)
        {
            tempicon.src="./icons/snow.svg"
        }
        else if(id<800 && id>700)
        {
            tempicon.src="./icons/clouds.svg"
        }
        else if(id==800)
        {
            tempicon.src="./icons/clouds-and-sun.svg"
        }
    }
    catch(error)
    {
        alert('city not found');
    }
};

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

let long;
let lat;

if(navigator.geolocation)
{

    navigator.geolocation.getCurrentPosition((position)=>
    {
        long=position.coords.longitude;
        lat=position.coords.latitude;
        const proxy="https://cors-anywhere.herokuapp.com/";

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

        fetch(api).then((response)=>{

            return response.json();

        })

        .then (data =>
        {

            const{name}=data;
            const{feels_like}=data.main;
            const{id,main}=data.weather[0];
            loc.textContent=name;
            climate.textContent=main;

```

```

        tempvalue.textContent=Math.round(feels_like-273);
        if(id<300 && id>200)
        {
            tempicon.src="./icons/thunderstorm.svg"
        }
        else if(id<400 && id>300)
        {
            tempicon.src="./icons/cloud-solid.svg"
        }
        else if(id<600&& id>500)
        {
            tempicon.src="./icons/rain.svg"
        }
        else if(id<700 && id>600)
        {
            tempicon.src="./icons/snow.svg"
        }
        else if(id<800 && id>700)
        {
            tempicon.src="./icons/clouds.svg"
        }
        else if(id==800)
        {
            tempicon.src="./icons/clouds-and-sun.svg"
        }
        console.log(data);
    })
}
    })
})

```

### style.css Page Code:

```

*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}
body{
    height: 100px;
    background-color: #263859;
}

#map1 {
    height: 400%;
    width: 90%;
    margin: auto;
}
h1 {
    text-align: center;

```

```
}
#search-form
{
    display: flex;
    justify-content: center;
    align-items: center;
    margin-top: 100px;
}

#temp
{
    text-align: center;
}

#temp #temp-unit
{
    font-family: -apple-
system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open
Sans', 'Helvetica Neue', sans-serif;
    font-size: 1.5rem;
    color: #ffffff;
}
#temp #temp-value
{
    font-family: -apple-
system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open
Sans', 'Helvetica Neue', sans-serif;
    font-size: 2rem;
    color: #ffffff;
}
#climate
{
    font-family: -apple-
system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open
Sans', 'Helvetica Neue', sans-serif;
    font-size: 2.5rem;
    color: #ffffff;
    text-align: center;
}
#location
{
    font-family: -apple-
system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open
Sans', 'Helvetica Neue', sans-serif;
    font-size: 2.5rem;
    color: #ffffff;
    text-align: center;
}
```

```
#temp-icon
```

```
{  
  margin: 1rem;  
  width: 40%;  
  height: auto;  
}
```

```
main{
```

```
  padding: 20px;  
  margin-top: 10px;  
  background-color: #6b778d;  
  border-radius: 20px;  
  width: 360px;  
  height: 360px;  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
  box-shadow: 10px 10px 20px 10px rgba(112, 102, 102, 0.2);
```

```
}
```

```
#search-button
```

```
{  
  color: #494949;  
  text-transform: uppercase;  
  text-decoration: none;  
  background: white;  
  border-radius: 5px;  
  margin-left: 5px;  
  padding: 11px;  
  border: 2px solid #494949;  
  display: inline-block;  
  transition: all 0.4 ease 0s;
```

```
}
```

```
#search-button:hover{
```

```
  color: #ffffff;  
  background: #ff6768;  
  border-color: #ff6768;  
  transition: all 0.4s ease 0s;
```

```
}
```

```
#search-input
{
    width: 260px;
    overflow: hidden;
    font-size: 20px;
    margin: 8px 0;
    padding: 8px 0;
    border-bottom: 3px solid #ff6768;
    color: white;
    outline: none;
    background: none;
}
input
{
    border: none;
}
/* google sign-in css */

.g-signin2{
    margin-left: 50%;
}
/* .g-signin2 .onSignIn{
    height: 100%;
    width: 50%;
} */
```

# API's Code ScreenShot's

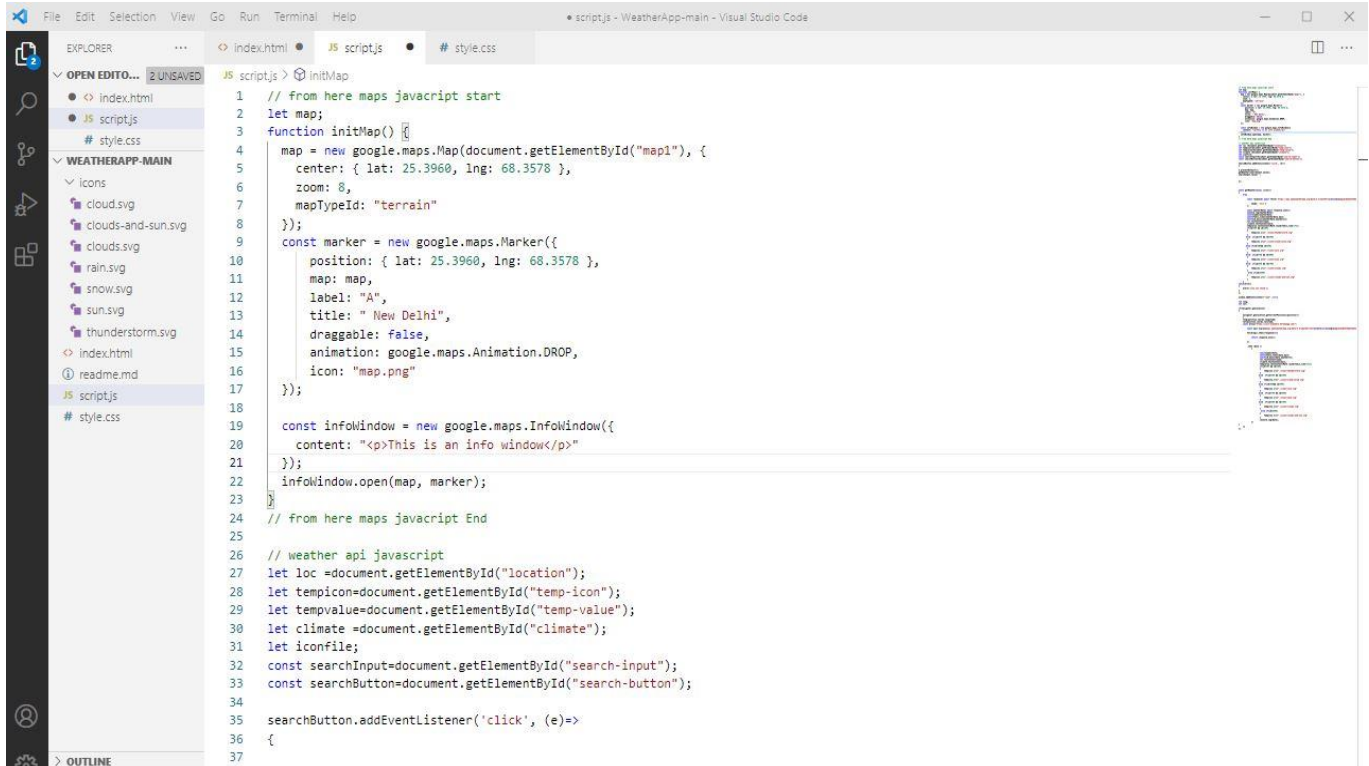
## index.html code Screenshot:

```
1 <!DOCTYPE html>
2 <html>
3   <title>Weather app</title>
4   <head>
5     <link rel="stylesheet" href="style.css">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <!-- google map scrip tag -->
8     <script
9       src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCvq33M0RmCwJiaVRotlMxkAFIOjGvEaUrI&callback=initMap&libraries=&v=weekly"
10      defer
11    ></script>
12    <!-- google sign-in script tag -->
13    <script src="https://apis.google.com/js/platform.js" async defer></script>
14    <meta name="google-signin-client_id" content="107362585837-tviaam6oc08r8gguidh99cj7r51685k5.apps.googleusercontent.com">
15  </head>
16
17  <body>
18    <!-- from here google maps api start -->
19    <h1>GOOGLE MAP API</h1>
20    <hr>
21    <br>
22    <div id="map1"></div>
23
24
25    <!-- from here google maps api End -->
26    <h1>Weather API</h1>
27    <hr>
28    <form id="search-form">
29      <input type="search"
30        placeholder="Enter City Name"
31        id="search-input"
32        required
33        autocomplete="off"
34      />
35      <br>
36      <button id="search-button">Search</button>
37    </form>
38
39    <main id="app-container">
40      <div id="location">
41        <p>-----</p>
42      </div>
43      <div id="temp">
44        
45        <p><span id="temp-value">-----</span> <span id="temp-unit"> &#176c</span></p>
46      </div>
47      <div id="climate">
48        <p>-----</p>
49      </div>
50    </main>
51    <!-- from here Google sign -->
52    <h1>Sign-in Api</h1>
53    <hr>
54    <br>
55    <div class="g-signin2" data-onsuccess="onSignIn"></div>
56
57    <script src="script.js">
58  </script>
59 </body>
60 </html>
```

```
36 </br>
37 <button id="search-button">Search</button>
38 </form>
39
40 <main id="app-container">
41   <div id="location">
42     <p>-----</p>
43   </div>
44   <div id="temp">
45     
46     <p><span id="temp-value">-----</span> <span id="temp-unit"> &#176c</span></p>
47   </div>
48   <div id="climate">
49     <p>-----</p>
50   </div>
51 </main>
52 <!-- from here Google sign -->
53 <h1>Sign-in Api</h1>
54 <hr>
55 <br>
56 <div class="g-signin2" data-onsuccess="onSignIn"></div>
57
58 <script src="script.js">
59 </script>
60 </body>
61 </html>
```

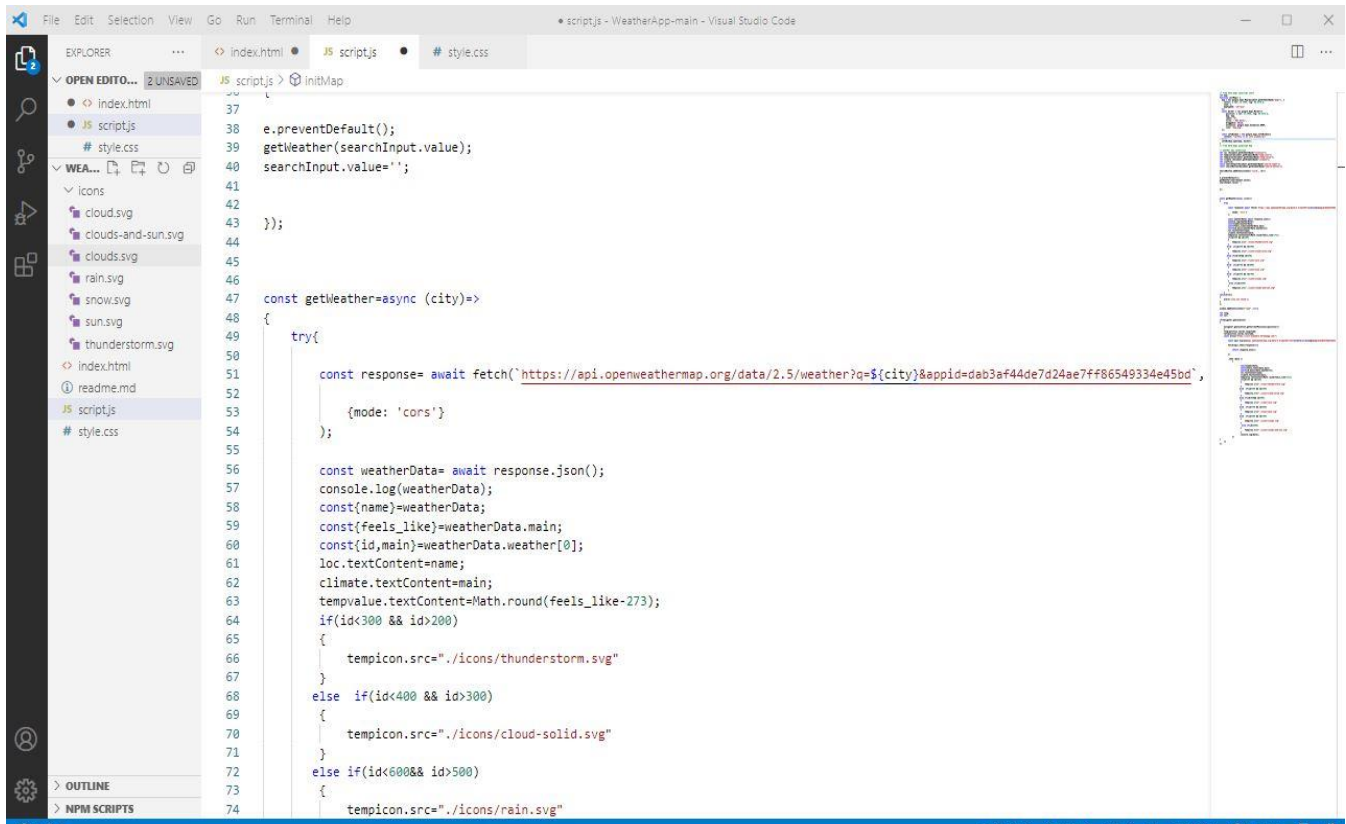


## script.js Code ScreenShot's:



This screenshot shows the initialMap function in script.js. The function initializes a Google Map with a center point at lat: 25.3960, lng: 68.3578, zoom level 8, and terrain map type. It also adds a marker for New Delhi and an info window. The function is followed by a section for the weather API, which defines variables for location, temperature icon, temperature value, climate, and search input/button elements. An event listener is added to the search button.

```
1 // from here maps javascript start
2 let map;
3 function initMap() {
4     map = new google.maps.Map(document.getElementById("map1"), {
5         center: { lat: 25.3960, lng: 68.3578 },
6         zoom: 8,
7         mapTypeId: "terrain"
8     });
9     const marker = new google.maps.Marker({
10         position: { lat: 25.3960, lng: 68.3578 },
11         map: map,
12         label: "A",
13         title: " New Delhi",
14         draggable: false,
15         animation: google.maps.Animation.DROP,
16         icon: "map.png"
17     });
18
19     const infoWindow = new google.maps.InfoWindow({
20         content: "<p>This is an info window</p>"
21     });
22     infoWindow.open(map, marker);
23 }
24 // from here maps javascript End
25
26 // weather api javascript
27 let loc =document.getElementById("location");
28 let tempicon=document.getElementById("temp-icon");
29 let tempvalue=document.getElementById("temp-value");
30 let climate =document.getElementById("climate");
31 let iconFile;
32 const searchInput=document.getElementById("search-input");
33 const searchButton=document.getElementById("search-button");
34
35 searchButton.addEventListener('click', (e)=>
36 {
37
```



This screenshot shows the getWeather function in script.js. The function is an async function that takes a city name as input. It uses fetch to get weather data from the OpenWeatherMap API. The response is parsed as JSON and used to update the UI elements (location, temperature icon, temperature value, climate) based on the weather conditions. The function also includes a try-catch block to handle errors.

```
37
38 e.preventDefault();
39 getWeather(searchInput.value);
40 searchInput.value='';
41
42
43 });
44
45
46
47 const getWeather=async (city)=>
48 {
49     try{
50
51         const response= await fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=dab3af44de7d24ae7f86549334e45bd`,
52             {mode: 'cors'}
53         );
54
55
56         const weatherData= await response.json();
57         console.log(weatherData);
58         const(name)=weatherData.name;
59         const(feels_like)=weatherData.main.feels_like;
60         const(id,main)=weatherData.weather[0];
61         loc.textContent=name;
62         climate.textContent=main;
63         tempvalue.textContent=Math.round(feels_like-273);
64         if(id<300 && id>200)
65         {
66             tempicon.src="./icons/thunderstorm.svg"
67         }
68         else if(id<400 && id>300)
69         {
70             tempicon.src="./icons/cloud-solid.svg"
71         }
72         else if(id<600&& id>500)
73         {
74             tempicon.src="./icons/rain.svg"
75         }
76     }
77     catch{
78
79     }
80 }
```

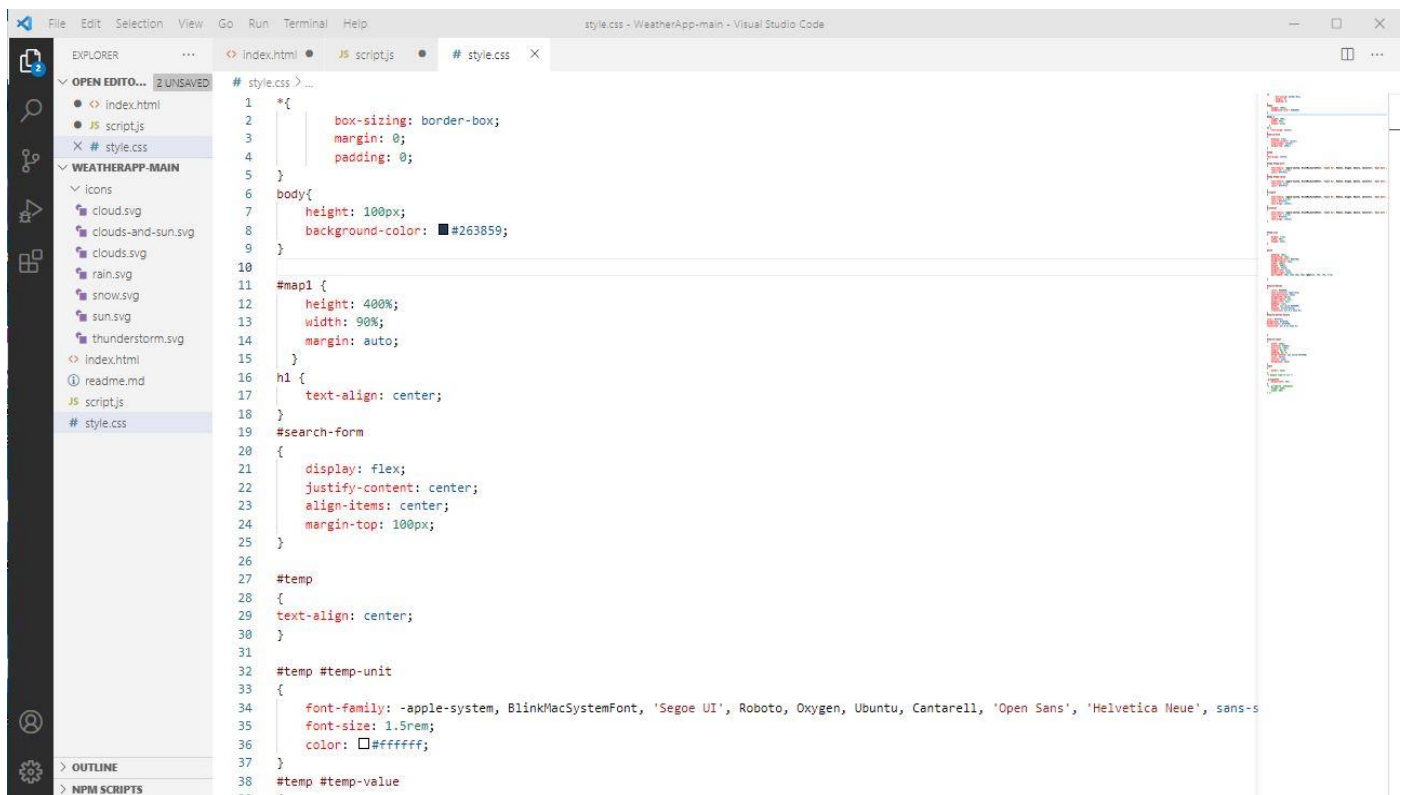
```
File Edit Selection View Go Run Terminal Help • scriptjs - WeatherApp-main - Visual Studio Code
EXPLORER
  OPEN EDITOR... 2 UNSAVED
  • index.html
  • JS scriptjs
  # style.css
  WEATHERAPP-MAIN
    icons
      cloud.svg
      clouds-and-sun.svg
      clouds.svg
      rain.svg
      snow.svg
      sun.svg
      thunderstorm.svg
    index.html
    readme.md
    JS scriptjs
    # style.css
  OUTLINE
  NPM SCRIPTS

index.html • JS scriptjs • # style.css
JS scriptjs > initMap
70     tempicon.src="./icons/cloud-solid.svg"
71   }
72   else if(id<600&& id>500)
73   {
74     tempicon.src="./icons/rain.svg"
75   }
76   else if(id<700 && id>600)
77   {
78     tempicon.src="./icons/snow.svg"
79   }
80   else if(id<800 && id>700)
81   {
82     tempicon.src="./icons/clouds.svg"
83   }
84   else if(id==800)
85   {
86     tempicon.src="./icons/clouds-and-sun.svg"
87   }
88   }
89   catch(error)
90   {
91     alert('city not found');
92   }
93   };
94
95   window.addEventListener("load", ()=>{
96
97     let long;
98     let lat;
99
100    if(navigator.geolocation)
101    {
102
103      navigator.geolocation.getCurrentPosition((position)=>
104      {
105        long=position.coords.longitude;
106        lat=position.coords.latitude;
107        const proxy="https://cors-anywhere.herokuapp.com/";
108
```

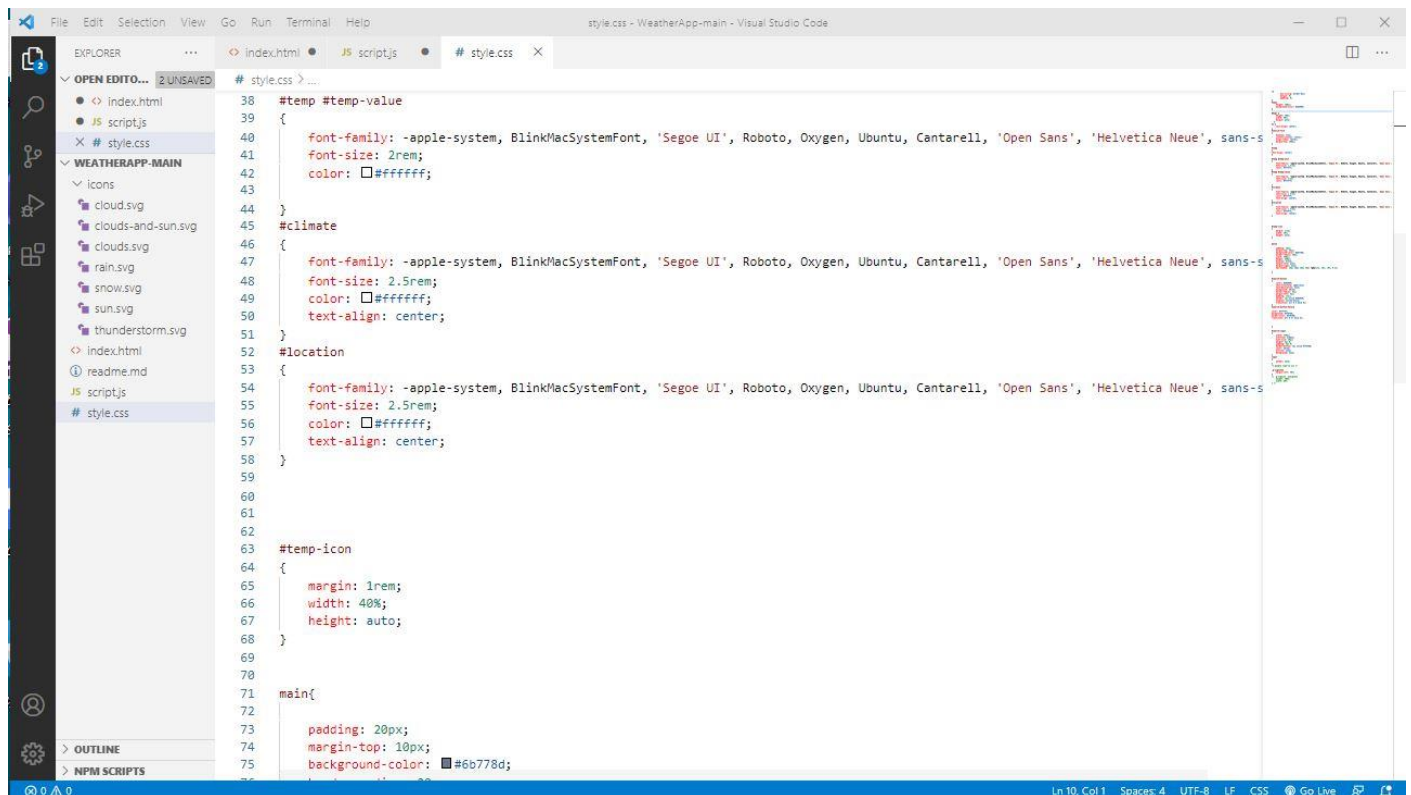
```
File Edit Selection View Go Run Terminal Help • scriptjs - WeatherApp-main - Visual Studio Code
EXPLORER
  OPEN EDITOR... 2 UNSAVED
  • index.html
  • JS scriptjs
  # style.css
  WEATHERAPP-MAIN
    icons
      cloud.svg
      clouds-and-sun.svg
      clouds.svg
      rain.svg
      snow.svg
      sun.svg
      thunderstorm.svg
    index.html
    readme.md
    JS scriptjs
    # style.css
  OUTLINE
  NPM SCRIPTS

index.html • JS scriptjs • # style.css
JS scriptjs > initMap
117     .then (data =>
118     {
119
120       const{name}=data;
121       const{feels_like}=data.main;
122       const{id,main}=data.weather[0];
123       loc.textContent=name;
124       climate.textContent=main;
125       tempvalue.textContent=Math.round(feels_like-273);
126       if(id<300 && id>200)
127       {
128         tempicon.src="./icons/thunderstorm.svg"
129       }
130       else if(id<400 && id>300)
131       {
132         tempicon.src="./icons/cloud-solid.svg"
133       }
134       else if(id<600&& id>500)
135       {
136         tempicon.src="./icons/rain.svg"
137       }
138       else if(id<700 && id>600)
139       {
140         tempicon.src="./icons/snow.svg"
141       }
142       else if(id<800 && id>700)
143       {
144         tempicon.src="./icons/clouds.svg"
145       }
146       else if(id==800)
147       {
148         tempicon.src="./icons/clouds-and-sun.svg"
149       }
150       console.log(data);
151     })
152   })
153 })
154
```

## style.css Code ScreenShot's:



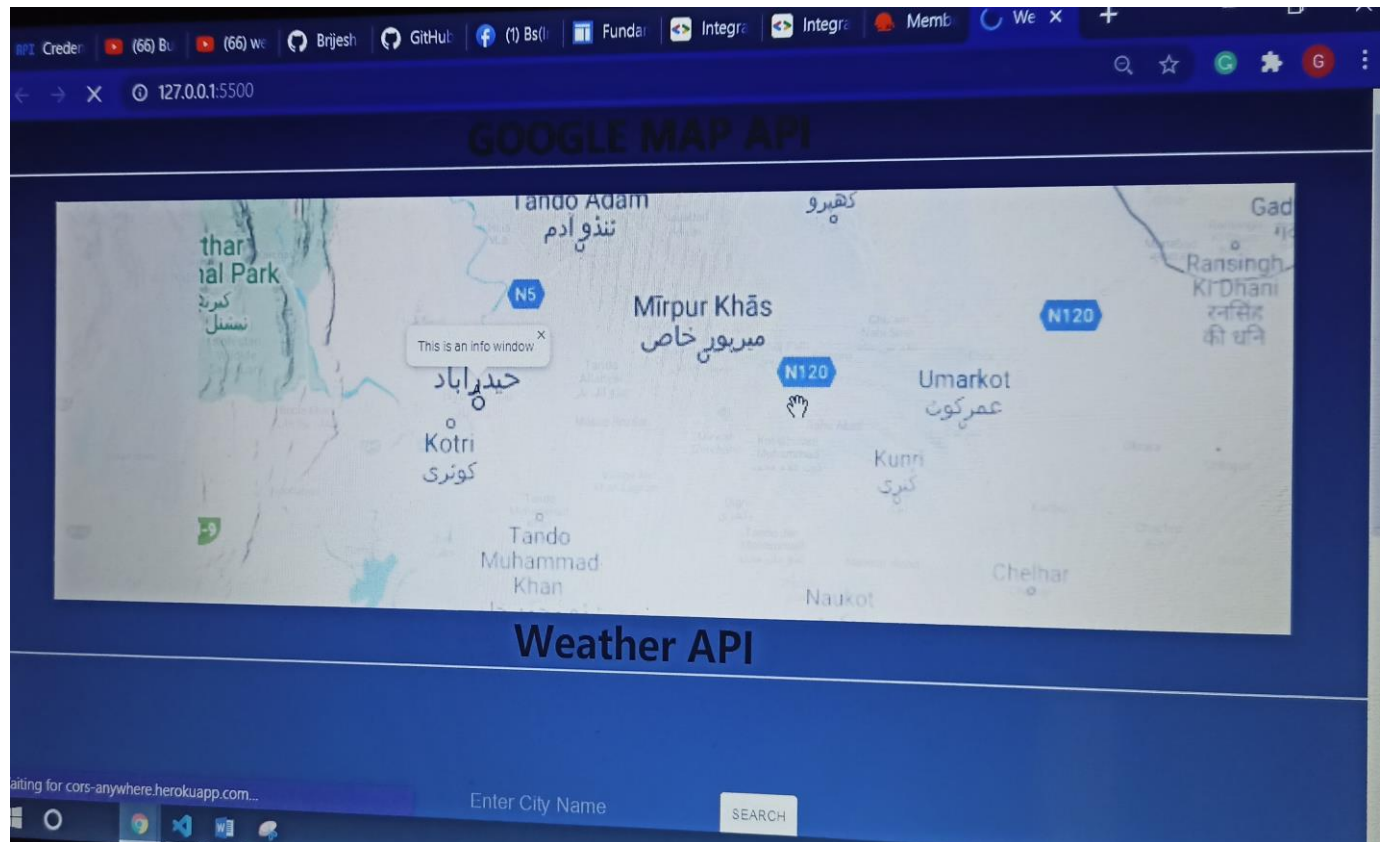
```
# style.css > ...
1  *{
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6  body{
7      height: 100px;
8      background-color: #263859;
9  }
10
11  #map1 {
12      height: 400%;
13      width: 90%;
14      margin: auto;
15  }
16  h1 {
17      text-align: center;
18  }
19  #search-form
20  {
21      display: flex;
22      justify-content: center;
23      align-items: center;
24      margin-top: 100px;
25  }
26
27  #temp
28  {
29      text-align: center;
30  }
31
32  #temp #temp-unit
33  {
34      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
35      font-size: 1.5rem;
36      color: #ffffff;
37  }
38  #temp #temp-value
```



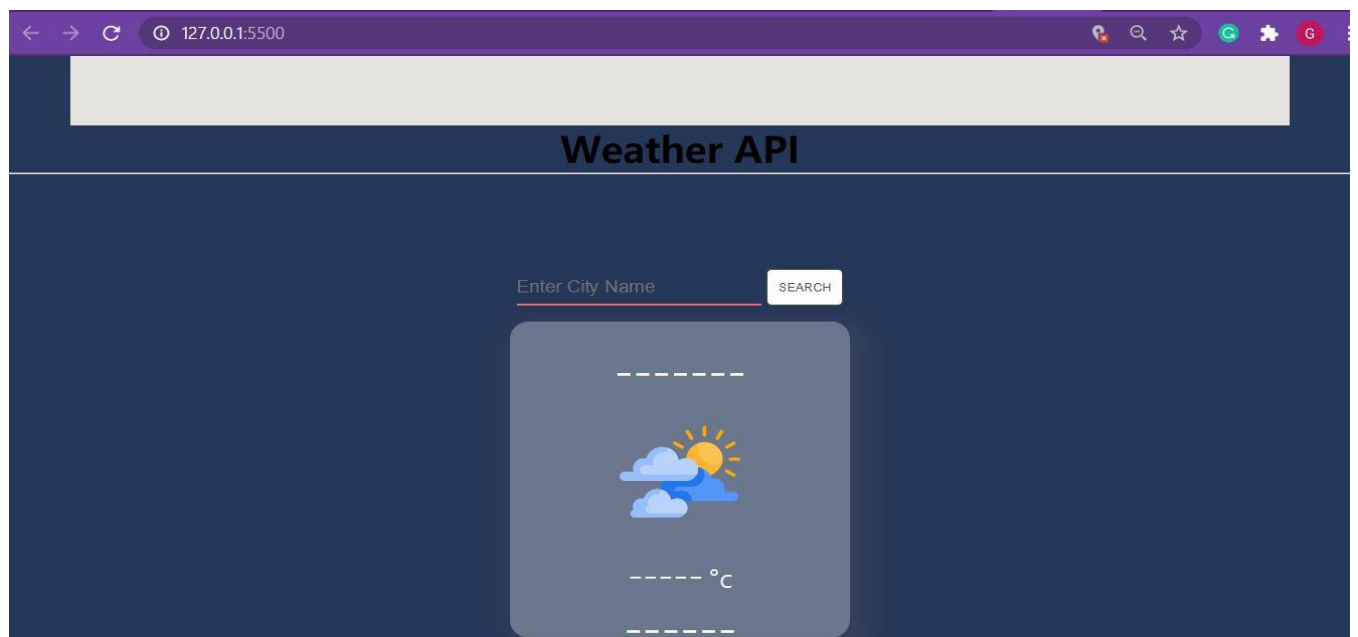
```
38  #temp #temp-value
39  {
40      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
41      font-size: 2rem;
42      color: #ffffff;
43  }
44
45  #climate
46  {
47      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
48      font-size: 2.5rem;
49      color: #ffffff;
50      text-align: center;
51  }
52  #location
53  {
54      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
55      font-size: 2.5rem;
56      color: #ffffff;
57      text-align: center;
58  }
59
60
61
62
63  #temp-icon
64  {
65      margin: 1rem;
66      width: 40%;
67      height: auto;
68  }
69
70
71  main{
72
73      padding: 20px;
74      margin-top: 10px;
75      background-color: #6b778d;
```

## API Project Result or Outputs

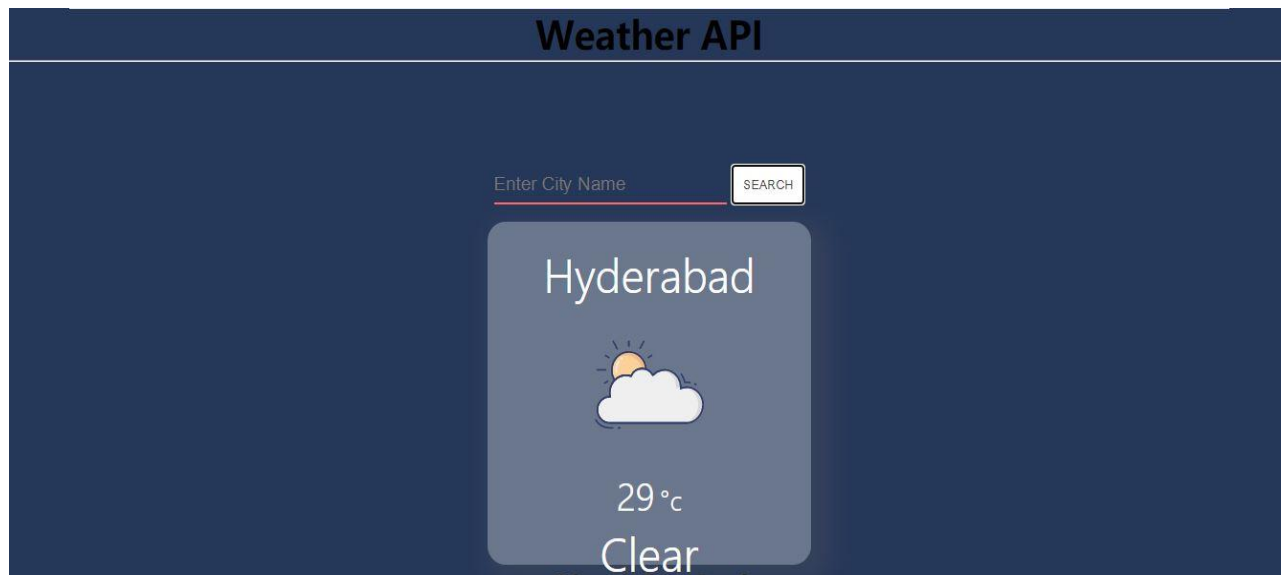
### Google Map API Result on web page:



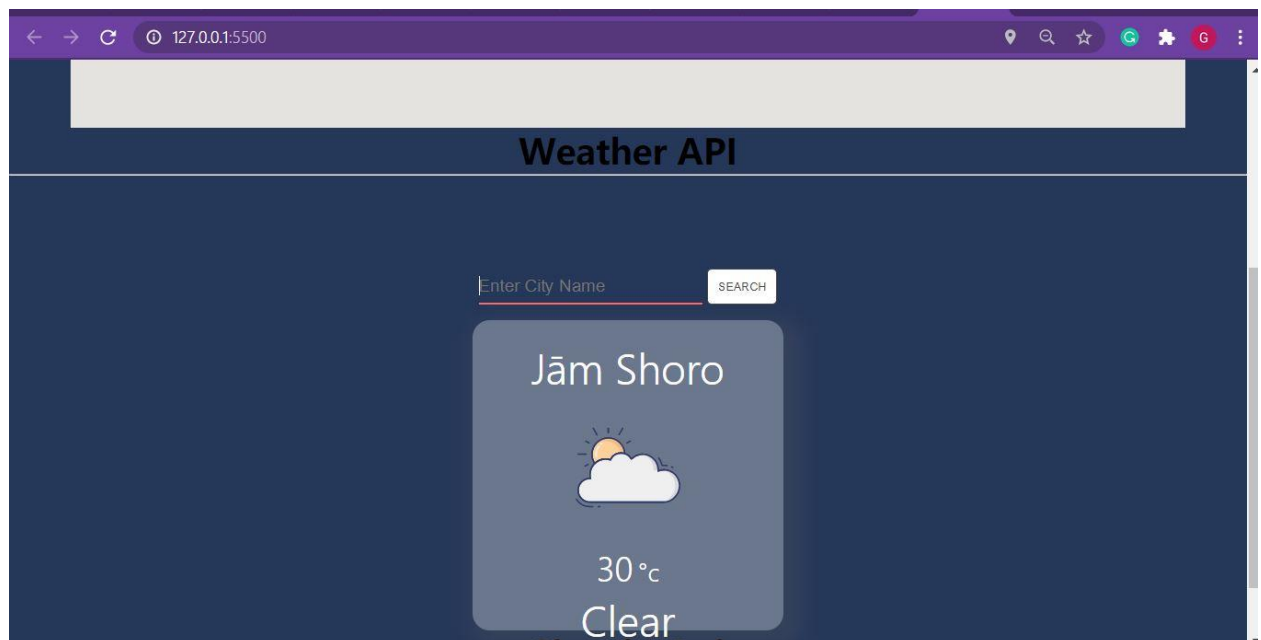
### OpenWeather API Result on web page:



## OpenWeather API with “Hyderabad” Search.

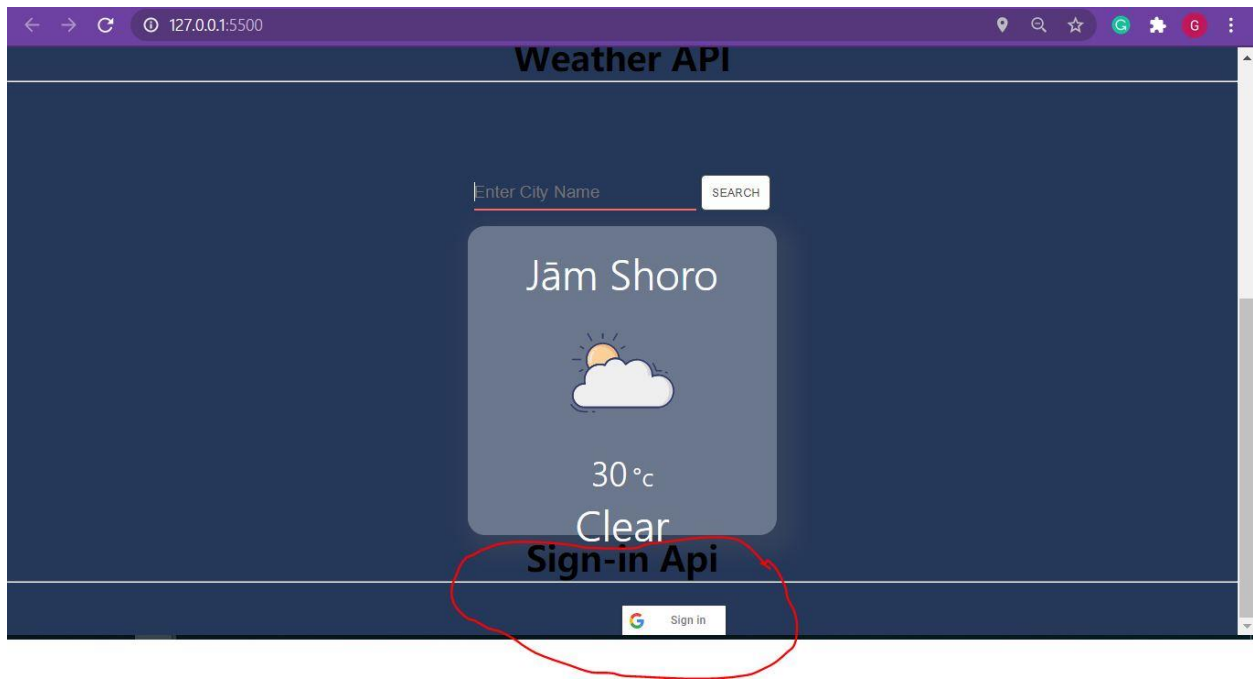


## OpenWeather API with “jamshoro” Search.





## Google Sign-In API Result on web page:



## Click on Google Sign-In API Button Result on web page:

