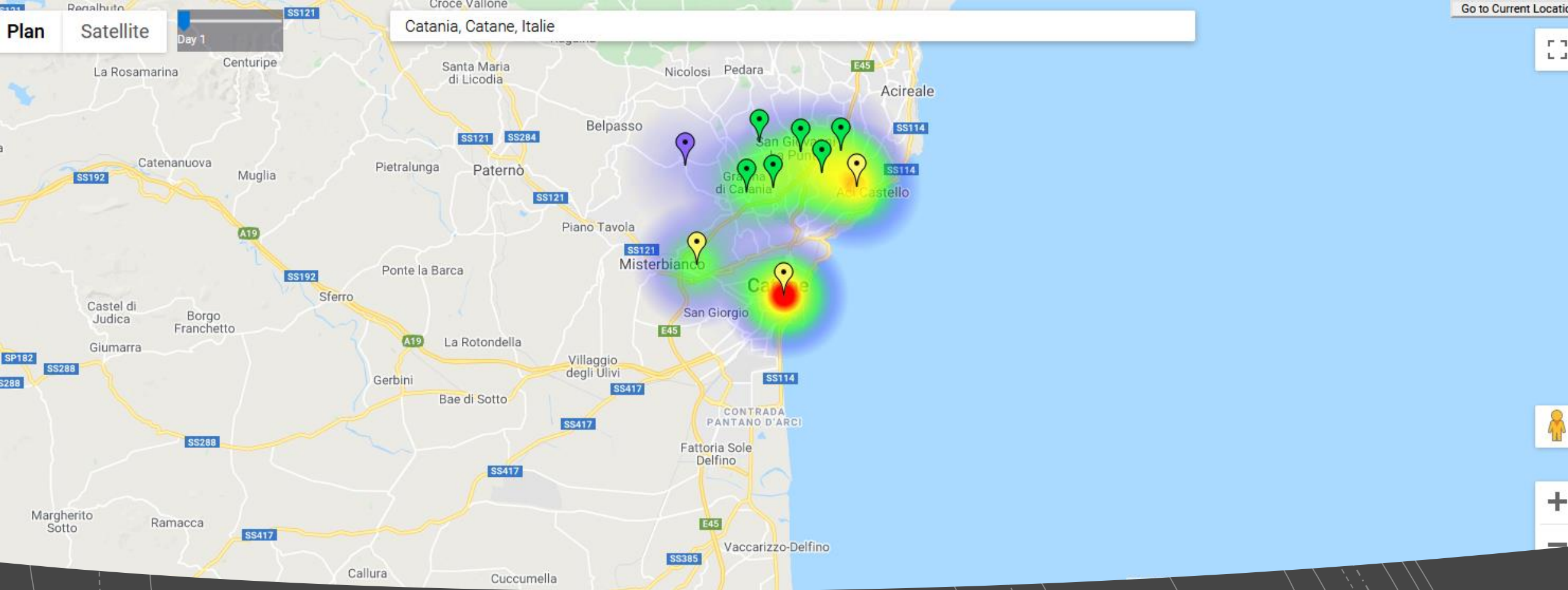




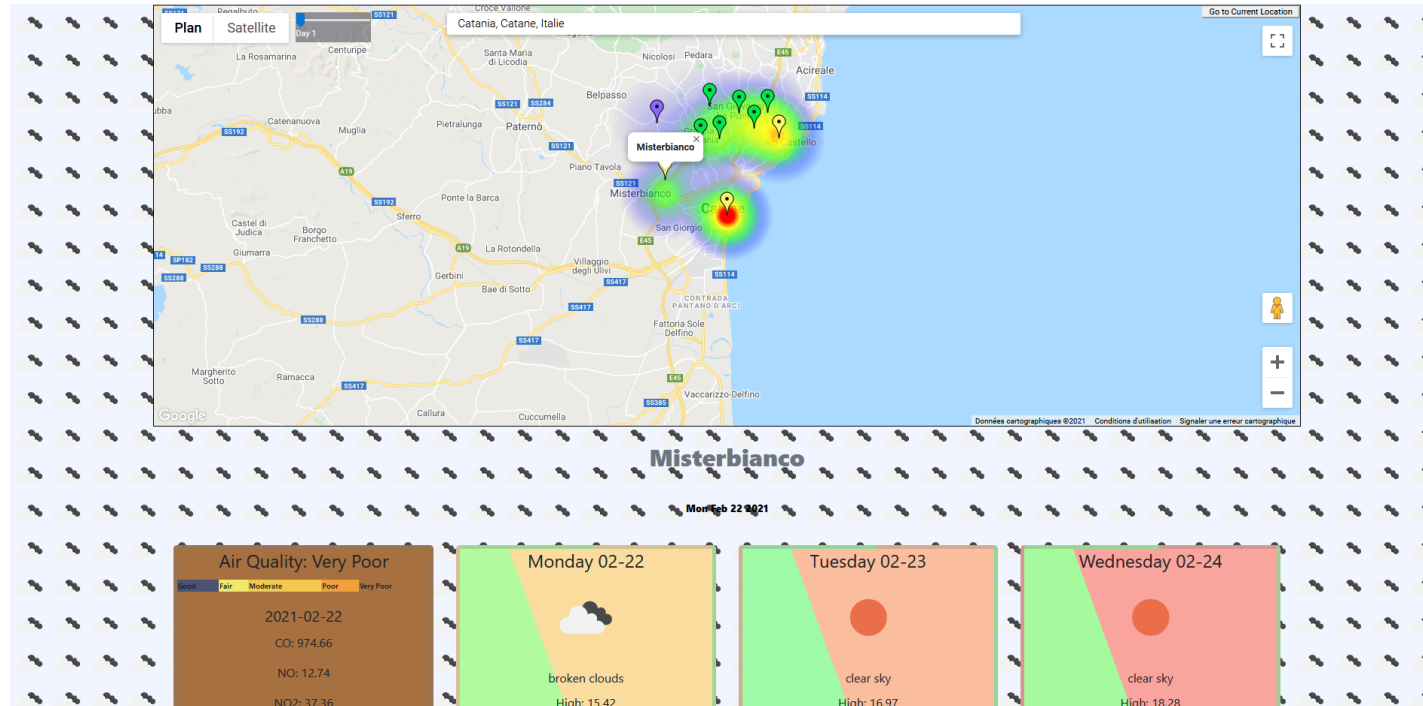
A Node.js web  
application

WeatherVenue code  
revue



## Technologies used

- The web application relies on Google Maps services and Openweathermap.org data



User experience  
using GMap and  
weather cards

- An interactive map that illustrates "surrounding cities" weather forecast. Cities weather can be discovered separately or even side by side (multiple weather cards)



# Hello there,

- I would like to share with you my journey making a simple website during lock-down.
- This is a code overview of a complete solution which is far from being perfect; If you are a beginner in web development, then I think you will learn few things just reviewing my code and approach in general.
- Because I am not ready and comfortable for speaking while live coding, I preferred to prepare this transcript, and rather use a text to speech service. The video presentation will continue like so. Note that you can choose a different language from Youtube live translation service but I don't guaranty the accuracy for languages other than English.
- I am not a web developer nor a keen Node.js developer. I know the way I coded it is not the best from software design perspective, far from that, we can say it is pasta code. Even so, it works as expected !

# Product and technologies

- "WeatherVenue" is a weather website using Google Maps and Openweathermap APIs. It lets people find best places to visit in their entourage; Easily finding warmer winter vacation or cool summer escapes.
- Technologies used in server side:
  - HTML, CSS and JavaScript
  - Node.js: a modern back-end JavaScript runtime environment
  - Express: a back end web application framework for Node.js
  - EJS: Embedded JavaScript templates
  - Redis: An in-memory data structure store
  - Sentry and Google Analytics: Error monitoring and web analytics
  - Other Node.js dependencies like: helmet, express-rate-limit, axios

# Other libraries

- Technologies used in client side:
  1. HTML, CSS and JavaScript
  2. Bootstrap
  3. Google Maps
- Other libraries like
  1. cookies-eu-banner.js,
  2. screenfull.js,
  3. heatmap.js,
  4. dark-mode-switch.js

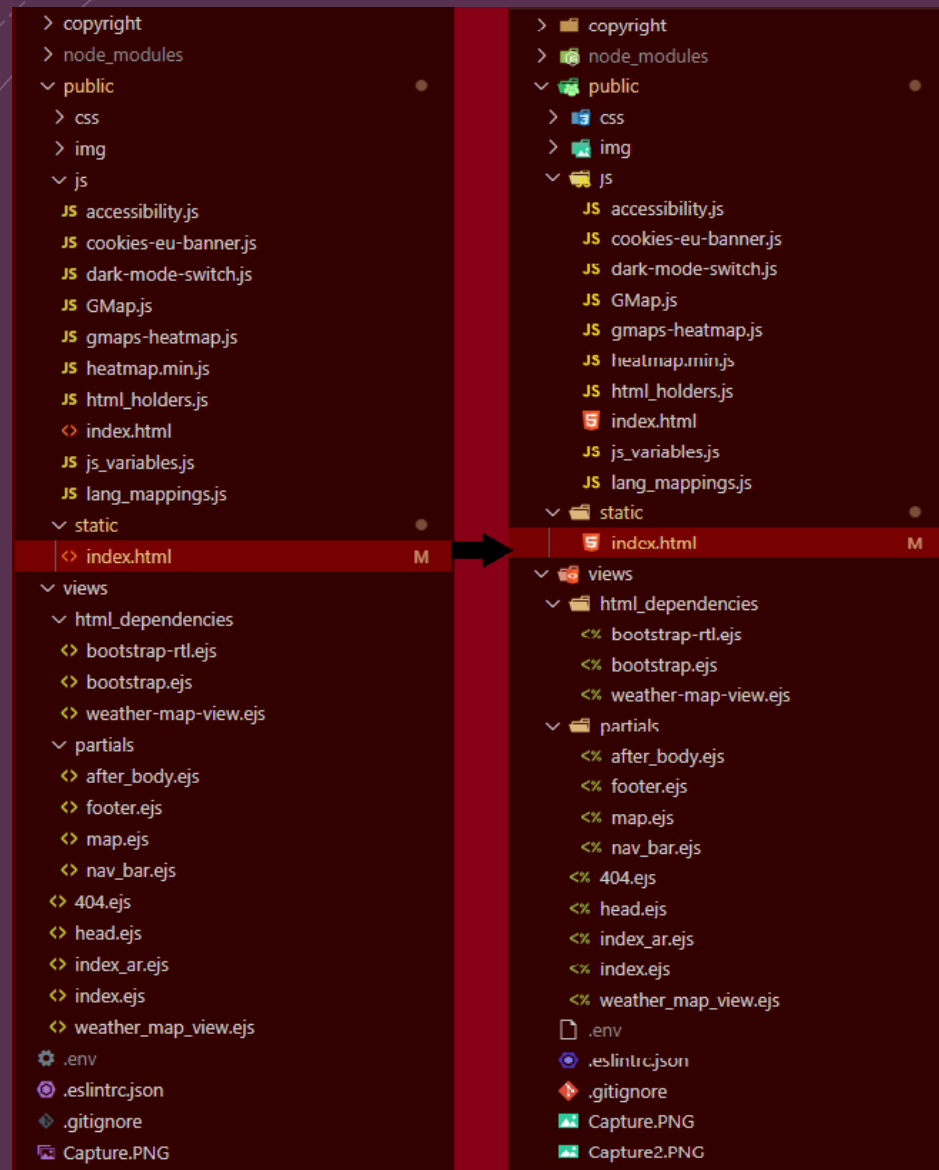
# Plan

- Open accounts for Google Maps API and Openweathermap keys
- Build a one page web application
- Understand then pick canonical example from service providers whenever possible
- Test using browser developer console; Either Chrome or Firefox or others
- Open accounts for Sentry and Google Analytics and setup keys
- Open an account in Heroku then deploy the app

# Setup

- Remember this is a code review, so code is already present and we will not be coding.
- First open Visual Studio Code, then install this plugin: `vscode-icons`. You will then see icons change from this && to this &&
- Also, I installed EJS language support plugin, you will see support of EJS files from this && to this &&.





```

1 <meta charset="UTF-8">
2 <meta name="viewport" content=
  "width=device-width, initial-scale=1">
3 <meta name="Description" content="<%= description %>">
4 <title><%= title %></title>
5 <script src=
  "https://polyfill.io/v3/polyfill.min.js?features=default"
  defer></script>
6 <% if(version=='bootstrap'){ %>
7   <%- include('html_dependencies/bootstrap'); %>
8   <% } else if(version=='bootstrap-rtl'){ %>
9   <%- include('html_dependencies/bootstrap-rtl'); %>
10  <% } %>
11 <script
12 src=
  "https://maps.googleapis.com/maps/api/js?key=AIzaSyBcL91cr-kioveam
  PxcvnQTT40A9asC1TE&callback=initMap&libraries=places&v=weekly"
  centerLocation="<%= centerLocation %>" lang="<%= lang %>"
13 defer></script>
14 <script src="../../js/heatmap.min.js" defer></script>
15 <script src="../../js/gmaps-heatmap.js" defer></script>
16 <link rel="stylesheet" type="text/css" href="../../css/style.css" />
17 <script src="../../js/cookies-eu-banner.js"></script>
18 <script src=
  "https://cdnjs.cloudflare.com/ajax/libs/screenfull.js/5.1.0/screen
  full.min.js"
  integrity=
  "sha512-SGPHIoS+Nsp1NUL5RohNpDs44J1F36tXLN6H3Cw+EUYenEc5zPXWqfw9D+
  xmvR00QYUYewQIJQ6P5yH82Vw6Fg=="
  crossorigin="anonymous"></script>
19 <!-- Global site tag (gtag.js) - Google Analytics -->
20 <script async src=
  "https://www.googletagmanager.com/gtag/js?id=G-59D6066VJ6"
  ></script>
21 <script data-ad-client="ca-pub-4987312092359418" async src=
  "https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"
  ></script>

```



```

1 <meta charset="UTF-8">
2 <meta name="viewport" content=
  "width=device-width, initial-scale=1">
3 <meta name="Description" content="<%= description %>">
4 <title><%= title %></title>
5 <script src=
  "https://polyfill.io/v3/polyfill.min.js?features=default"
  defer></script>
6 <% if(version=='bootstrap'){ %>
7   <%- include('html_dependencies/bootstrap'); %>
8   <% } else if(version=='bootstrap-rtl'){ %>
9   <%- include('html_dependencies/bootstrap-rtl'); %>
10  <% } %>
11 <script
12 src=
  "https://maps.googleapis.com/maps/api/js?key=AIzaSyBcL91cr-kioveam
  PxcvnQTT40A9asC1TE&callback=initMap&libraries=places&v=weekly"
  centerLocation="<%= centerLocation %>" lang="<%= Lang %>"
13 defer></script>
14 <script src="../../js/heatmap.min.js" defer></script>
15 <script src="../../js/gmaps-heatmap.js" defer></script>
16 <link rel="stylesheet" type="text/css" href="../../css/style.css" />
17 <script src="../../js/cookies-eu-banner.js"></script>
18 <script src=
  "https://cdnjs.cloudflare.com/ajax/libs/screenfull.js/5.1.0/screen
  full.min.js"
  integrity=
  "sha512-SGPHIoS+Nsp1NUL5RohNpDs44J1F36tXLN6H3Cw+EUYenEc5zPXWqfw9D+
  xmvR00QYUYewQIJQ6P5yH82Vw6Fg=="
  crossorigin="anonymous"></script>
19 <!-- Global site tag (gtag.js) - Google Analytics -->
20 <script async src=
  "https://www.googletagmanager.com/gtag/js?id=G-59D6066VJ6"
  ></script>
21 <script data-ad-client="ca-pub-4987312092359418" async src=
  "https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"
  ></script>

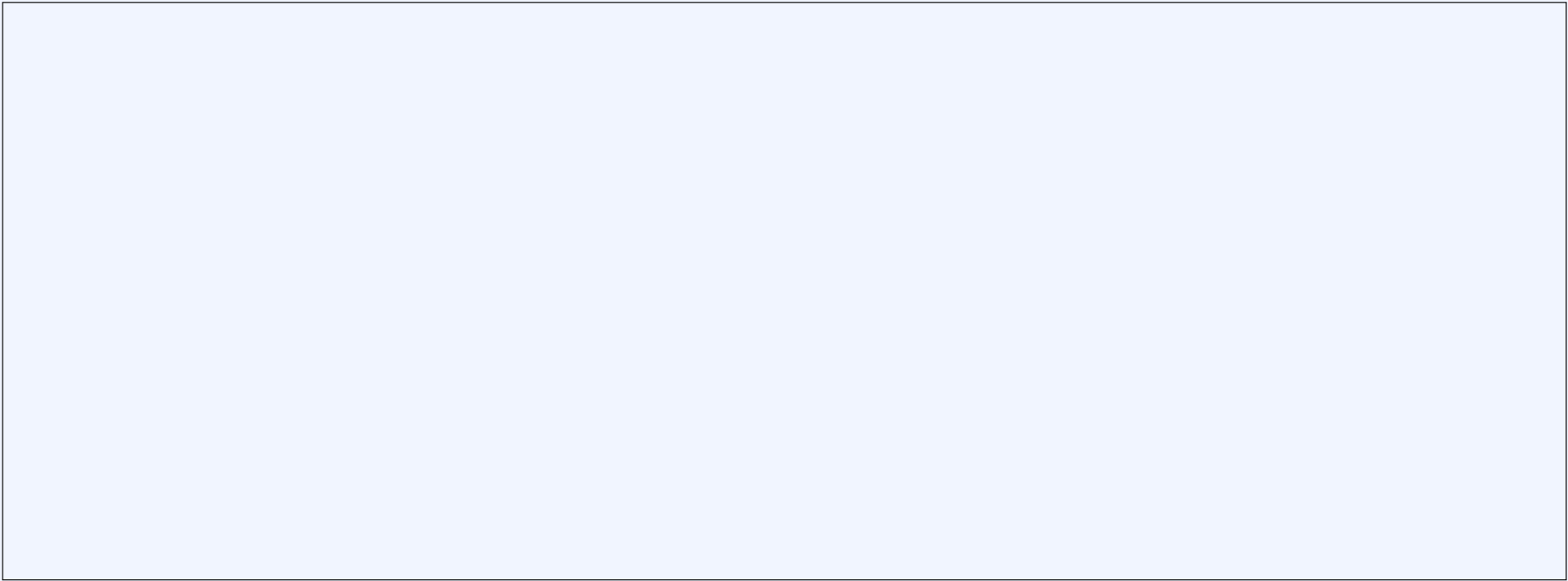
```

## ... Setup

- Install a new Node.js project by running: `npm init`
- install required dependencies by running:
- `npm install @sentry/browser @sentry/node @sentry/tracing axios dotenv ejs express express-rate-limit helmet joi nearby-cities offline-geocoder redis request reverse-geocode serve-favicon umbress`
- Then you will need to install Redis database server locally.

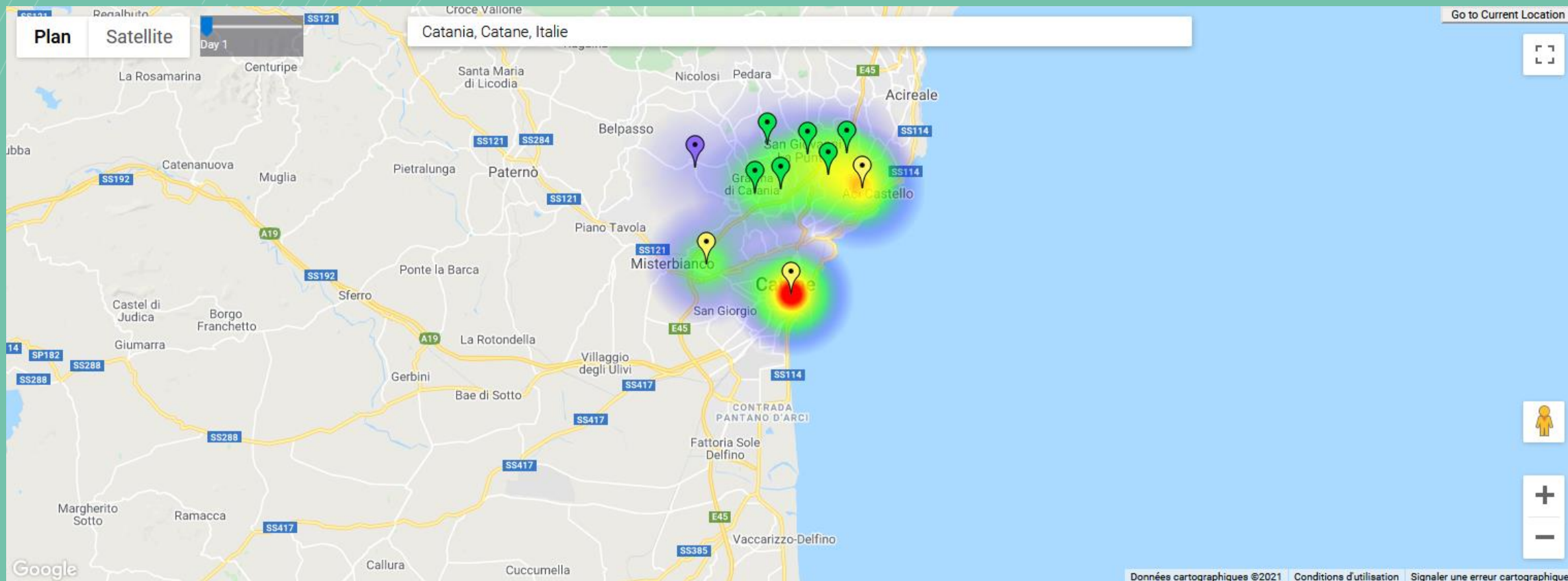
By continuing to visit this site, you accept the use of cookies by Google Analytics for statistical purposes. [Read more](#) [Reject](#) [Accept](#)

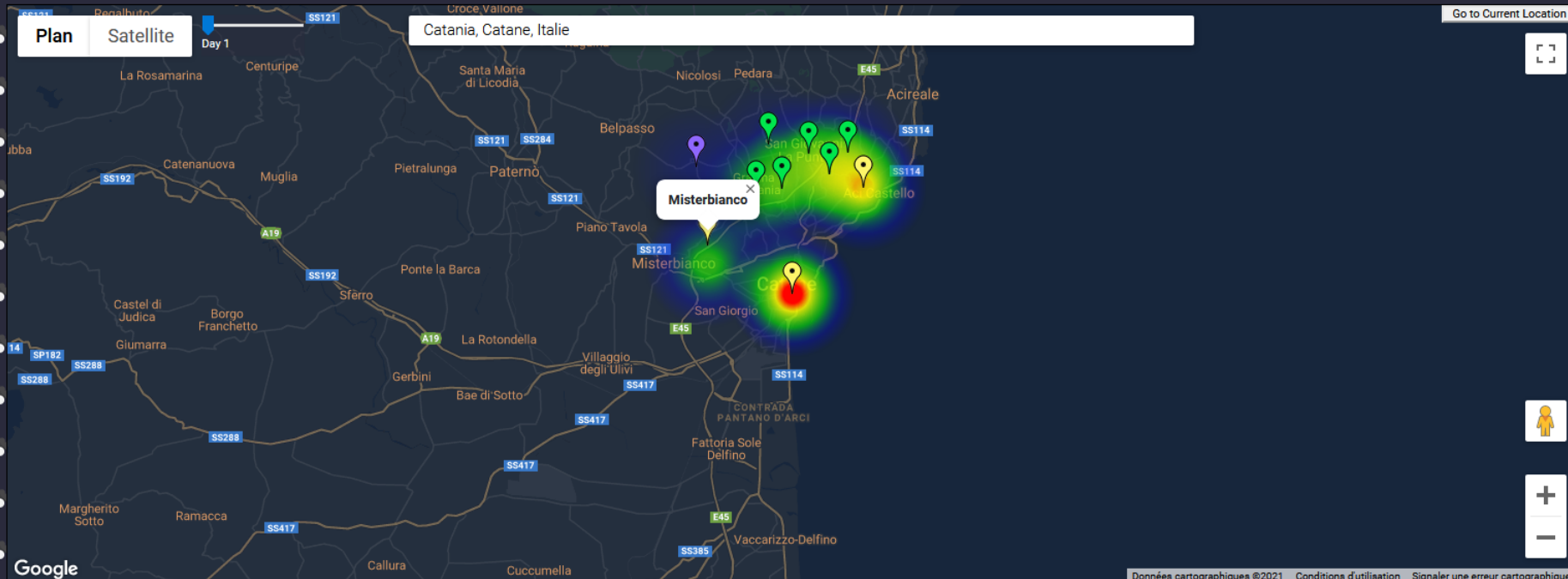
Enter a location



**Search for your city to display next days forecasts.**

**Local pictures of the main city also will appear.**





## Misterbianco

Mon Feb 22 2021

Air Quality: Very Poor

Good Fair Moderate Poor Very Poor

2021-02-22

Monday 02-22



Tuesday 02-23



Wednesday 02-24





EXPLORER

OPEN EDITORS

index.html public:static M

WEATHERVENUE

public

css

img

js

accessibility.js

cookies-eu-banner.js

dark-mode-switch.js

GMap.js

gmaps-heatmap.js

heatmap.min.js

html HOLDERS.js

index.html

js\_variables.js

lang\_mappings.js

static

index.html M

views

html\_dependencies

bootstrap-rtl.ejs

bootstrap.ejs

weather-map-view.ejs

partials

after\_body.ejs

footer.ejs

map.ejs

nav\_bar.ejs

404.ejs

head.ejs

index\_ar.ejs

index.ejs

weather\_map\_view.ejs

TODOTREE: TODOS

Scan mode: workspace and ope...

Nothing found

OUTLINE

TIMELINE

index.html X

public > static > index.html > html

1<!DOCTYPE html>

2<html lang="en">

3

4<head>

5<meta charset="UTF-8">

6<meta name="viewport" content="width=device-width, initial-scale=1">

7<meta name="Description"

8| content="WeatherVenue lets you find best places to visit in your entourage; Easily finding warmer winter vacation or cool summer escapes.">

9<title>Welcome to WeatherVenue</title>

10<script src="https://polyfill.io/v3/polyfill.min.js?features=default" defer></script>

11

12<script src="https://code.jquery.com/jquery-3.5.1.min.js"

13| integrity="sha256-9/aliU8dGd2tb60SsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous" defer></script>

14<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

15| integrity="sha384-JcKb8q3iqJ61gNV9KgB8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">

16<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"

17| integrity="sha384-B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"

18| defer></script>

19

20<script

21| src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBcL91cr-kioveamPxcvnQTT40A9asC1TE&callback=initMap&libraries=places&v=weekly"

22| centerLocation="london" lang="en" defer></script>

23<link rel="stylesheet" type="text/css" href="../css/style.css" />

24<script src="../js/cookies-eu-banner.js"></script>

25<script src="https://cdnjs.cloudflare.com/ajax/libs/screenfull.js/5.1.0/screenfull.min.js"

26| integrity="sha512-SGPHIoS+Nsp1NUL5RohNpDs44JlF36tXLN6H3Cw+EUyenEc5zPXWqfw9D+xmvrR00QYUYewQIJQ6P5yH82Vw6Fg=="

27| crossorigin="anonymous"></script>

28<!-- Global site tag (gtag.js) - Google Analytics -->

29<script async src="https://www.googletagmanager.com/gtag/js?id=G-59D6066VJ6"></script>

30</head>

31

32<body>

33<div id="cookies-eu-banner" style="display: none;">

34| By continuing to visit this site, you accept the use of cookies by Google Analytics for statistical purposes.

35| <a href="https://www.cookieclaw.org/google-analytics-eu-cookie-law/" id="cookies-eu-more">Read more</a>

36| <button id="cookies-eu-reject">Reject</button>

37| <button id="cookies-eu-accept">Accept</button>

38</div>

39<div id="spinner-back"></div>

40<div id="spinner-front">

41| <br>

42</div>

43<nav class="navbar navbar-expand-lg navbar-light bg-light py-0">

44| <a class="navbar-brand" href="/">

45| | 

47</a>

48<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"

49| aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1">
4   <meta name="Description"
5     content="WeatherVenue lets you find best places to visit in your entourage; Easily finding warmer winter vacation or cool summer escapes.">
6   <title>Welcome to WeatherVenue</title>
7   <script src="https://polyfill.io/v3/polyfill.min.js?features=default" defer></script>
8
9   <script src="https://code.jquery.com/jquery-3.5.1.min.js"
10     integrity="sha256-9/aliU8dGd2tb60SsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous" defer></script>
11   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
12     integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
13   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
14     integrity="sha384-B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"
15     defer></script>
16
17   <script
18     src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBcL91cr-kioveamPxcvnQTT40A9asC1TE&callback=initMap&libraries=places&v=weekly"
19     centerLocation="london" lang="en" defer></script>
20   <link rel="stylesheet" type="text/css" href="../css/style.css" />
21   <script src="../js/cookies-eu-banner.js"></script>
22   <script src="https://cdnjs.cloudflare.com/ajax/libs/screenfull.js/5.1.0/screenfull.min.js"
23     integrity="sha512-SGPHIoS+Nsp1NUL5RohNpDs44JlF36tXLN6H3Cw+EUYenEc5zPXWqfw9D+xmVR00QYUYewQIJQ6P5yH82Vw6Fg=="
24     crossorigin="anonymous"></script>
25   <!-- Global site tag (gtag.js) - Google Analytics -->
26   <script async src="https://www.googletagmanager.com/gtag/js?id=G-59D6066VJ6"></script>
27 </head>
```



```
3 By continuing to visit this site, you accept the use of cookies by Google Analytics for statistical purposes.
4 <a href="https://www.cookie-law.org/google-analytics-eu-cookie-law/" id="cookies-eu-more">Read more</a>
5 <button id="cookies-eu-reject">Reject</button>
6 <button id="cookies-eu-accept">Accept</button>
7 </div>
8 <div id="spinner-back"></div>
9 <div id="spinner-front">
10   <br>
11 </div>
12 <nav class="navbar navbar-expand-lg navbar-light bg-light py-0">
13   <a class="navbar-brand" href="/">
14     
16   </a>
17   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
18     aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
19     <span class="navbar-toggler-icon"></span>
20   </button>
21
22   <div class="collapse navbar-collapse" id="navbarNav">
23     <ul class="navbar-nav mr-auto">
24       <li class="nav-item inactive">
25         <a class="nav-link disabled" id="widget" href="#">City week forecast</a>
26       </li>
27       <li class="nav-item btn btn-secondary">
28         <a class="nav-link" href="weather_map_view">Weather map</a>
29       </li>
30
31       <!-- <li class="nav-item btn btn-secondary">
32         <a class="nav-link" href="#">Donate</a>
33       </li> -->
34       <li class="nav-item inactive">
35         <a class="nav-link disabled" href="#">Version 2</a>
36       </li>
37       <li class="nav-item btn btn-secondary">
38         <a class="nav-link" href="#" data-toggle="modal" data-target="#exampleModal">Disclaimer</a>
39       </li>
40       <li class="nav-item btn btn-secondary">
41         <a class="nav-link"
```

```
1 </body>
2 <script src="../../js/js_variables.js"></script>
3 <script src="../../js/lang_mappings.js"></script>
4 <script src="../../js/html_holders.js"></script>
5 <script src="../../js/accessibility.js"></script>
6 <script src="../../js/GMap.js"></script>
7 <script>
8   new CookiesEuBanner(function () {
9     // Your code to launch when user accept cookies
10    // Google analytics
11    window.dataLayer = window.dataLayer || []
12    function gtag() { dataLayer.push(arguments) }
13    gtag('js', new Date())
14    gtag('config', 'G-59D6066VJ6')
15    console.log('accepted')
16  });
17  fullScreenBtn = __id('fullscreen');
18  fullScreenBtn.addEventListener('click', () => {
19    if (screenfull.isEnabled) {
20      screenfull.request();
21    } else {
22      // Ignore or do something else
23    }
24  });
25  screenfull.on('change', () => {
26    screenfull.isFullscreen ? fullScreenBtn.style.visibility = "hidden" : fullScreenBtn.style.visibility = "visible";
27  });
28 </script>
29
30 </html>
```

## Client-Server roundtrip

- In client side using Google Map, we capture user inputs.
- We then construct a valid Ajax call so that the one index page hits the API in server without refreshing the view.
- In server side, our API hits the openweathermap onecall API, retrieves weather data, and finally extract usefull data and send formatted JSON back as a response.

Client side

# Map initialization

```
1 function initMap () {
2   let center = { Lat: -33.8688, Lng: 151.2195 }
3   const scripts = document.getElementsByTagName('script')
4   const mapScript = scripts[3]
5   Language = mapScript.getAttribute('lang')
6   const centerLocation = mapScript.getAttribute('centerLocation')
7   switch (centerLocation) {
8     case 'algiers':
9       center = { Lat: 36.75, Lng: 3.05 }
10      break
11     case 'paris':
12       center = { Lat: 48.85, Lng: 2.35 }
13      break
14     case 'london':
15       center = { Lat: 51.50, Lng: 0.12 }
16      break
17     default:
18       break
19   }
20
21   // When initMap is called for a second time, choose the earlier center not to move the map center away i
22   n the globe
23   if (currentList && currentList.features && currentList.features.length > 0) {
24     const coordinates = currentList.features[0].geometry.coordinates
25     center = {
26       Lat: coordinates[1],
27       Lng: coordinates[0]
28     }
29   }
30   // Instantiate the map or clean it if it already exists
31   if (!map) {
32     map = new google.maps.Map(__id('map'), {
33       center: center,
34       zoom: 10
35     })
36   } else {
37     (function (m) {
38       m.data.forEach(function (f) {
39         m.data.remove(f)
40       })
41     })(map)
42     google.maps.event.trigger(map, 'resize')
43   }
44 }
```

Client side

# Construct the Ajax call

```
1 function nearbyRequest (place) {
2   _showLoading() // Block page while Loading
3   var cache = _getWithExpiry('response_' + place.name)
4   if (cache && cache.length > 0) {
5     currentList = cache
6     __id('location').innerHTML = currentList.features[0].properties.name
7     renderForecastDays(currentList.weather[0].daily)
8     initMap()
9     _hideLoading() // Unblock page
10    return
11  }
12  const request = new XMLHttpRequest()
13  const requestObject = JSON.stringify({
14    lat: place.geometry.location.lat(),
15    lng: place.geometry.location.lng(),
16    cityname: place.name,
17    language: Language
18  })
19  request.open('GET', 'nearby/' + requestObject)
20  request.responseType = 'json'
21  request.onload = function () {
22    currentList = request.response.data
23    _setWithExpiry('response_' + place.name, currentList)
24    __id('location').innerHTML = currentList.features[0].properties.name
25    renderForecastDays(currentList.weather[0].daily)
26    initMap()
27    populateHeatMap(0)
28    _hideLoading() // Unblock page
29    const googleTemplate = _adsHolder('Google')
30    __id('forecast-items').insertAdjacentHTML('beforeend', googleTemplate)
31  }
32  request.send()
33 }
```

Server side

validate request  
and respond if  
error

```
const reqSchema = Joi.object({
  lat: Joi.number().min(-90).max(90).required(),
  lng: Joi.number().min(-180).max(180).required(),
  cityname: Joi.string().min(3).max(180).required(),
  language: Joi.string().min(2).max(2).required()
})
// 3 //////////////////////////////////////
app.get('/nearby/:city', function rootHandler (req, res) {
  try {
    if (!req.params.city) {
      return res.status(400).send({
        error: true,
        message: 'Bad request',
        data: 'Bad request'
      })
    }
  }

  const geometry = JSON.parse(req.params.city)
  const valid = reqSchema.validate(geometry)
  if (valid.error) {
    return res.status(400).send({
      error: true,
      message: 'Bad request',
      data: 'Bad request error ' + valid.error
    })
  }
})
```

Server side

fetch  
Openweathermap  
API and  
respond

```
1 // Check the redis store for the data first
2 client.get(cityname, async (err, result) => {
3   // redis unexpected errors
4   if (err) {
5     console.error(err)
6     return res.status(500).send({
7       error: true,
8       message: 'Server error',
9       data: 'Server error'
10    })
11  }
12  if (result) {
13    return res.status(200).send({
14      error: false,
15      message: `Weather data for nearby cities for ${cityname} from the cache`,
16      data: JSON.parse(result)
17    })
18  } else {
19    const query = {
20      latitude: geometry.lat,
21      longitude: geometry.lng
22    }
23    const cities = nearbyCities(query).slice(0, 10)
24    const actions = cities.map(fetchWeather)
25    Promise.all(actions).then(function (forecasts) {
26      var weathers = forecasts.map(elem => { return elem.weather })
27      var pollutions = forecasts.map(elem => { return elem.pollution })
28      const result = formatCities(cities, weathers, pollutions)
29      client.setex(cityname, 24 * 60 * 3, JSON.stringify(result))
30      return res.status(200).send({
31        error: false,
32        message: 'Weather data for nearby cities from the server',
33        data: result
34      })
35    })
36  }
37 })
```



```
1 let language = 'en'
2 async function fetchWeather (city) {
3   return new Promise(async (resolve, reject) => {
4     const APIUrlWeather = `https://api.openweathermap.org/data/2.5/onecall?lat=${city.Lat}&lon=${city.Lon}
    &lang=${language}&exclude=hourly,minutely,hourly&units=metric&appid=${OPENWEATHERMAP_API_KEY}`
5     const body0 = await axios.get(APIUrlWeather)
6     const data0 = await body0.data
7     const APIUrlPollution = `http://api.openweathermap.org/data/2.5/air_pollution?lat=${city.Lat}&lon=${city
    .Lon}&appid=${OPENWEATHERMAP_API_KEY}`
8     const body1 = await axios.get(APIUrlPollution)
9     const data1 = await body1.data
10    resolve({ weather: data0, pollution: data1 })
11  })
12 }
```



# Other considerations

Next, I show server configuration.

It uses "Express" framework.

We set "helmet", It helps you secure your Express apps by setting various HTTP headers.

We use "axios", It is "Promise" based HTTP client for the browser and node.js.

"dotenv" to separate configuration keys and values from code.

"express-rate-limit" to limit repeated requests to public APIs and/or endpoints such as password reset.

# Back to configuration and server bootstrap

```
1  const dotenv = require('dotenv')
2  dotenv.config()
3  console.log(`Your port is ${process.env.PORT}`) // 8626
4  const express = require('express')
5  const helmet = require('helmet')
6  const Joi = require('joi')
7  const axios = require('axios')
8  const redis = require('redis')
9  const nearbyCities = require('nearby-cities')
10 const favicon = require('serve-favicon')
11 const path = require('path')
12 const app = express()
13 const Sentry = require('@sentry/node')
14 const Tracing = require('@sentry/tracing')
15 if (process.env.NODE_ENV !== 'dev') {
16   Sentry.init({
17     dsn: `https://${process.env.SENTRY_KEY}.ingest.sentry.io/1871185`,
18     integrations: [
19       new Sentry.Integrations.Http({ tracing: true }),
20       new Tracing.Integrations.Express({ app })
21     ],
22     tracesSampleRate: 1.0
23   })
24 }
25 app.use(Sentry.Handlers.requestHandler())
26 app.use(Sentry.Handlers.tracingHandler())
27 app.use(favicon(path.join(__dirname, 'public/img', 'icon.png')))
28 app.use(helmet({ contentSecurityPolicy: false }))
29 app.set('view engine', 'ejs')
30 const rateLimit = require('express-rate-limit')
31 const limiter = rateLimit({
32   windowMs: 15 * 60 * 1000, // 15 minutes
33   max: 30 // limit each IP to 30 requests per windowMs
34 })
35 app.use(limiter)
36 const nodePort = process.env.PORT
37 const redisPort = process.env.REDIS_PORT
38 const OPENWEATHERMAP_API_KEY = process.env.OPENWEATHERMAP_API_KEY
39
40 app.listen(nodePort, () => {
41   console.log(`Server running on port ${nodePort}`)
42 })
43 app.use(express.static(__dirname + '/'))
```

Server side

Capture bad  
visitors and send  
them to "nil"

```
1 const dns = require('dns')
2
3 app.use(function (req, res, next) {
4   let ip = req.headers['x-forwarded-for'] || req.connection.remoteAddress
5   if (ip.substr(0, 7) === '::ffff:') {
6     ip = ip.substr(7)
7   }
8
9   if (process.env.NODE_ENV === 'dev' || ip.split('.')[0] === '127') { return next() }
10  const reversedIp = ip.split('.').reverse().join('.')
11  dns.resolve4([process.env.HONEYPOT_KEY, reversedIp, 'dnsbl.httpbl.org'].join('.'), function (err, addresses) {
12    if (!addresses) { return next() }
13    const _response = addresses.toString().split('.').map(Number)
14    const test = (_response[0] === 127 && _response[3] > 0) // visitor_type[_response[3]]
15    if (test) { res.send({ msg: 'we hate spam to begin with!' }) }
16    return next()
17  })
18 })
```

I did not forget, how  
I generated the  
static "Index.html" in  
an ugly manner x)

```
1 app.use(function(req, res, next) {
2   res.status(404)
3   if (req.accepts('html')) {
4     res.render('404', { url: req.url })
5   }
6 })
7 const ejs = require('ejs')
8 const fs = require('fs')
9 app.get('/render#####', function (req, res) {
10   const data = {}
11   ejs.renderFile(path.join(__dirname, '/views/index.ejs'), data, (
12     err, result) => {
13     if (err) {
14       console.log('info', 'error encountered: ' + err)
15     } else {
16       try {
17         fs.writeFileSync('./public/static/index.html', result,
18           'utf8')
19       } catch (err) {
20         if (err) {
21           throw err
22         }
23       }
24     }
25   })
26   res.render('index')
27 })
```

Back to client,  
when the Ajax  
'*request.onload*'  
is triggered

- Gmap.js dependency comes to practice again.
- It format data and manipulate dom elements.
- '*renderForecastDays()*' renders card divs in HTML
- '*initMap()*' is called each time to refresh the map with new markers (surrounding cities) and not to instantiate the map again.
- '*getPictures()*' relies on *google.maps.placesAPI*



# Demonstration

See video here <https://www.awesomescreenshot.com/video/2839516?key=572abb2269fbbdec1740f3e4d73eee55>

# What's next ?

- Style, style, style !
- This project is on Github and is firstcomers friendly ! Nothing particularly intensive as you have seen. It is a chance for beginners to learn these technologies through a real world example also to learn collaborating using Git and Github.
- I am not a web developer myself, so, you can contribute, and I would not be demanding too much (I most likely accept your contributions)



# See you there !

- Github  
repository: <https://github.com/bacloud14/WeatherVenue>
- Open issues:  
<https://github.com/bacloud14/WeatherVenue/issues>
- Deployed version:  
<https://www.weathervenue.com/>
- Contact `b@cloud14[(at)]gmail(dot)c0m`