



# EARTH QUAKE TRACKER

**By Audrey Lee, Jessica Reslan, and Kasey Chung**

# Our Approach

- The API we chose is an earthquake tracker and alert system.
- <https://earthquake.usgs.gov/fdsnws/event/1>
- Our inputs: current longitude and latitude, and how big of a radius you want.
- Our outputs: all the recent earthquakes up to a month old in the determined radius.

# Our MongoDB

```
import pymongo
from pymongo import MongoClient
import json
import pprint
import requests
url = 'https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&latitude=34.069032199&longitude=-118.4435299&ma
posts=requests.get(url).json()
client=MongoClient()
db=client.GEarthquake_Tracker
collection=db.magnitude

post_id= collection.insert_one(posts).inserted_id
post_id
db.list_collection_names()
```

We used our database to import information from our API. However, we were able to access the API using AJAX, so we didn't use MongoDB in our final code.

# Our HTML Code

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6
7     <title>Recent Quakes</title>
8
9     <link href="https://fonts.googleapis.com/css?family=Dosis:400,700" rel="stylesheet" />
10    <link href="style.css" rel="stylesheet" />
11    <script src="script.js"></script>
12    <script src="https://code.jquery.com/jquery-1.7.2.js"></script>
13  </head>
14
15  <body>
16    <div id="root"></div>
17    <form id="fm1">
18      Latitude:<br>
19      <input type="text" name="firstname"><br>
20      Longitude:<br>
21      <input type="text" name="lastname">
22    </form>
23
24    <form id="fm2">
25      <input type="radio" name="range" value=10 checked>10 km<br>
26      <input type="radio" name="range" value=25>25 km<br>
27      <input type="radio" name="range" value=50>50 km<br>
28    </form>
29
30    <input id="submit" type="button" value="Submit" onclick="getCoords();" />
31
32    <input id="usecurr" type="button" value="Use Current Location" onclick="getLocation();" />
33    <p id="currloc"></p>
34  </body>
35 </html>
```

# Our CSS Code

```
1  * {  
2    box-sizing: border-box  
3  }  
4  
5  html {  
6    -webkit-font-smoothing: antialiased;  
7    -moz-osx-font-smoothing: grayscale;  
8    font-family: 'Dosis', sans-serif;  
9    line-height: 1.6;  
10   color: #666;  
11   background: #F6F6F6;  
12 }  
13  
14 #root {  
15   max-width: 1200px;  
16   margin: 0 auto;  
17 }  
18  
19 h1 {  
20   text-align: center;  
21   padding: 1.5rem 2.5rem;  
22   margin: 0 0 2rem 0;  
23   font-size: 1.5rem;  
24   color: white;  
25 }  
26  
27 input[type=text] {  
28   width: 15%;  
29   padding: 5px 10px;  
30   margin: 5px 0;  
31   box-sizing: border-box;  
32   border: 1px solid #555;  
33   outline: none;  
34   border-radius: 5px;  
35 }  
36  
37 input[type=text]:focus {  
38   background-color: lightgreen;  
39   border: 2px solid #555;  
40 }  
41  
42 img {  
43   display: block;  
44   margin: 1rem auto;  
45   max-width: 100%;  
46 }  
47  
48 p {  
49   padding: 0 2.5rem 2.5rem;  
50   margin: 0;  
51 }
```

```
52  
53 .container {  
54   display: flex;  
55   flex-wrap: wrap;  
56 }  
57  
58 .card {  
59   margin: 1rem;  
60   background: white;  
61   box-shadow: 2px 4px 25px rgba(0, 0, 0, .1);  
62   border-radius: 12px;  
63   overflow: hidden;  
64   transition: all .2s linear;  
65 }  
66  
67 .card:hover {  
68   box-shadow: 2px 8px 45px rgba(0, 0, 0, .15);  
69   transform: translate3D(0, -2px, 0);  
70 }  
71  
72 @media screen and (min-width: 600px) {  
73   .card {  
74     flex: 1 1 calc(50% - 2rem);  
75   }  
76 }  
77  
78 @media screen and (min-width: 900px) {  
79   .card {  
80     flex: 1 1 calc(33% - 2rem);  
81   }  
82 }  
83  
84 }  
85  
86 html {  
87   -webkit-font-smoothing: antialiased;  
88   -moz-osx-font-smoothing: grayscale;  
89   font-family: 'Dosis', sans-serif;  
90   line-height: 1.6;  
91   color: #666;  
92   background: #F6F6F6;  
93 }  
94  
95 #root {  
96   max-width: 1200px;  
97   margin: 0 auto;  
98 }  
99  
100 h1 {  
101   text-align: center;  
102   padding: 1.5rem 2.5rem;  
103   margin: 0 0 2rem 0;  
104   font-size: 1.5rem;  
105   color: white;
```

```
106  
107 input[type=text] {  
108   width: 15%;  
109   padding: 5px 10px;  
110   margin: 5px 0;  
111   box-sizing: border-box;  
112   border: 1px solid #555;  
113   outline: none;  
114   border-radius: 5px;  
115 }  
116  
117 input[type=text]:focus {  
118   background-color: lightgreen;  
119   border: 2px solid #555;  
120 }  
121  
122 img {  
123   display: block;  
124   margin: 1rem auto;  
125   max-width: 100%;  
126 }  
127  
128 p {  
129   padding: 0 2.5rem 2.5rem;  
130   margin: 0;  
131 }  
132  
133 .container {  
134   display: flex;  
135   flex-wrap: wrap;  
136 }  
137  
138 .card {  
139   margin: 1rem;  
140   background: white;  
141   box-shadow: 2px 4px 25px rgba(0, 0, 0, .1);  
142   border-radius: 12px;  
143   overflow: hidden;  
144   transition: all .2s linear;  
145 }  
146  
147 .card:hover {  
148   box-shadow: 2px 8px 45px rgba(0, 0, 0, .15);  
149   transform: translate3D(0, -2px, 0);  
150 }  
151  
152 @media screen and (min-width: 600px) {  
153   .card {  
154     flex: 1 1 calc(50% - 2rem);  
155   }  
156 }  
157 }
```

# Our JS Code

## Geolocation

```
53 function getRadioVal(id) {
54   var val;
55   var radios = document.getElementById(id);
56   for (var i = 0, len=radios.length; i < len; i++) {
57     if (radios[i].checked) {
58       val = radios[i].value;
59       break;
60     }
61   }
62   return val;
63 }
64
65 function getLocation() {
66   var x = document.getElementById("currloc");
67
68   if (navigator.geolocation) {
69     navigator.geolocation.getCurrentPosition(showPosition);
70   } else {
71     x.innerHTML = "Geolocation is not supported by this browser.";
72   }
73 }
```

## Accessing the API

```
8 function doFunction(lat,long) {
9   var latitude = "&latitude=" + lat;
10  var longitude = "&longitude=" + long;
11  var rad = "&maxradiuskm=" + getRadioVal("fm2");
12
13  const app = document.getElementById('root')
14  const container = document.createElement('div')
15  container.setAttribute('class','container')
16  app.appendChild(container)
17
18  var orig = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&minmagnitude=0.0" + latitude + longitude + rad;
19  var request = new XMLHttpRequest();
20  request.open("GET", orig, true);
```

## Determining Max Radius

```
52
53 function getRadioVal(id) {
54   var val;
55   var radios = document.getElementById(id);
56   for (var i = 0, len=radios.length; i < len; i++) {
57     if (radios[i].checked) {
58       val = radios[i].value;
59       break;
60     }
61   }
62   return val;
63 }
```

## User Input

```
1 function getCoords(){
2   var x = document.getElementById("fm1")
3   lat = x.elements[0].value;
4   long = x.elements[1].value;
5   doFunction(lat,long);
6 }
```

# Our JS Code

## Deciding Colors

```
81 function between(mag) {  
82   var compare = parseFloat(mag);  
83   if(compare >= 0 && compare <= 2.5) {  
84     return "green-yellow";  
85   } else if (compare > 2.5 && compare <= 5.0) {  
86     return "yellow-orange";  
87   } else if (compare > 5.0 && compare <= 7.5) {  
88     return "orange-red";  
89   } else {  
90     return "red-black";  
91   }  
92 }
```

## Displaying Data

```
22 request.onload = function() {  
23   var data = JSON.parse(this.response);  
24   var newArr = data.features;  
25  
26   newArr.forEach(quake => {  
27     prop = quake.properties  
28  
29     const card = document.createElement('div')  
30     card.setAttribute('class', 'card')  
31  
32     const h1 = document.createElement('h1')  
33     h1.textContent = prop.place  
34  
35     const p = document.createElement('p')  
36     text = "Magnitude: "  
37     p.textContent = text.concat(prop.mag)  
38  
39     var att = document.createAttribute("style")  
40     colors = between(prop.mag)  
41     cols = colors.split("-")  
42     att.value="background-image:linear-gradient(120deg, " + cols[0] + ", " + cols[1] + ")"  
43     h1.setAttributeNode(att);  
44  
45     container.appendChild(card)  
46     card.appendChild(h1)  
47     card.appendChild(p)  
48   })  
49 }  
50 request.send();  
51 }
```