WORKING WITH XML & JSON DATA FORMATS

Web Services [WAT 2124C]

UNIVERSITY of TECHNOLOGY, MAURITIUS

Name: BEEHARRY Mohammad Taajuddin (2003_19375)

Cohort: BSE20AFT

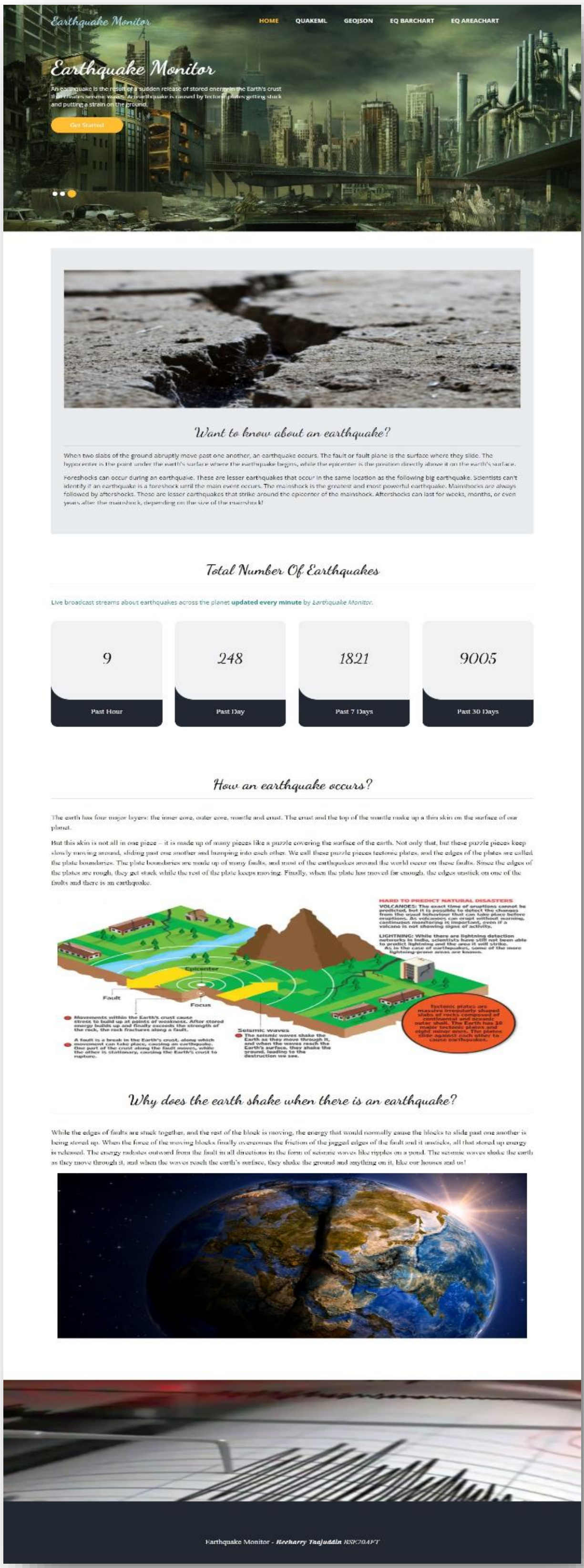
Year 2 Semester 2

BSC (HONS) SOFTWARE ENGINEERING

Lecturer: Dr. G. Suddul

Table of Contents

Homepage (index.php)



Earthquake Monitor

An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground.

Want to know about an earthquake?

When two slabs of the ground abruptly move past one another, an earthquake occurs. The fault or fault plane is the surface where they slide. The hypocenter is the point under the earth's surface where the earthquake begins, while the epicenter is the position directly above it on the earth's surface.

Foreshocks can occur during an earthquake. These are lesser earthquakes that occur in the same location as the following big earthquake. Scientists can't identify if an earthquake is a foreshock until the main event occurs. The mainshock is the greatest and most powerful earthquake. Mainshocks are always followed by aftershocks. These are lesser earthquakes that strike around the epicenter of the mainshock. Aftershocks can last for weeks, months, or even years after the mainshock, depending on the size of the mainshock!

Total Number Of Earthquakes

Live broadcast streams about earthquakes across the planet **updated every minute** by *Earthquake Monitor.*

9	248	1821	9005
Past Hour	Past Day	Past 7 Days	Past 30 Days

How an earthquake occurs?

The earth has four major layers: the inner core, outer core, mantle and crust. The crust and the top of the mantle make up a thin skin on the surface of our planet.

But this skin is not all in one piece – it is made up of many pieces like a puzzle covering the surface of the earth. Not only that, but these puzzle pieces keep slowly moving around, sliding past one another and bumping into each other. We call these puzzle pieces tectonic plates, and the edges of the plates are called the plate boundaries. The plate boundaries are made up of many faults, and most of the earthquakes around the world occur on these faults. Since the edges of the plates are rough, they get stuck while the rest of the plate keeps moving. Finally, when the plate has moved far enough, the edges unstick on one of the faults and there is an earthquake.

HARD TO PREDICT NATURAL DISASTERS

VOLCANOES: The exact time of eruptions cannot be predicted, but it is possible to detect the changes from the usual behaviour that can take place before eruptions. As volcanoes can erupt without warning, continuous monitoring is important, even if a volcano is not showing signs of activity.

LIGHTNING: While there are lightning detection networks in India, scientists have still not been able to predict lightning and the area it will strike. As in the case of earthquakes, some of the more lightning-prone areas are known.

Movements within the Earth's crust cause stress to build up at points of weakness. After stored energy builds up and finally exceeds the strength of the rock, the rock fractures along a fault.

A fault is a break in the Earth's crust, along which movement can take place, causing an earthquake. One part of the crust along the fault moves, while the other is stationary, causing the Earth's crust to rupture.

The seismic waves shake the Earth as they move through it, and when the waves reach the Earth's surface, they shake the ground, leading to the destruction we see.

Tectonic plates are massive irregularly shaped slabs of rocks composed of continental and oceanic outer shell. The Earth has 10 major tectonic plates and eight minor ones. The plates slide against each other to cause earthquakes.

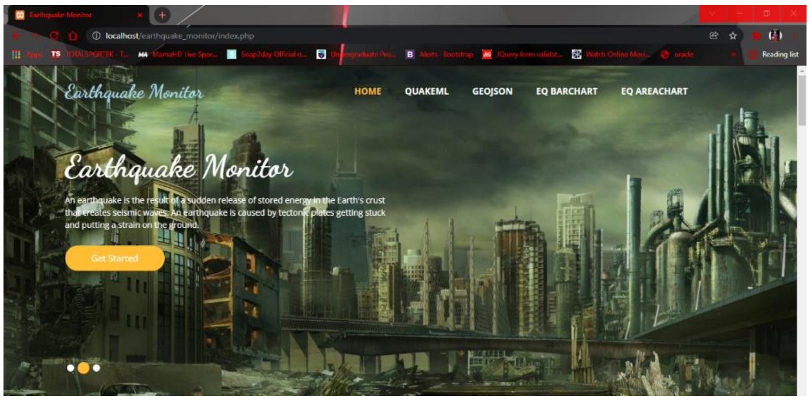
Why does the earth shake when there is an earthquake?

While the edges of faults are stuck together, and the rest of the block is moving, the energy that would normally cause the blocks to slide past one another is being stored up. When the force of the moving blocks finally overcomes the friction of the jagged edges of the fault and it unsticks, all that stored up energy is released. The energy radiates outward from the fault in all directions in the form of seismic waves like ripples on a pond. The seismic waves shake the earth as they move through it, and when the waves reach the earth's surface, they shake the ground and anything on it, like our houses and us!

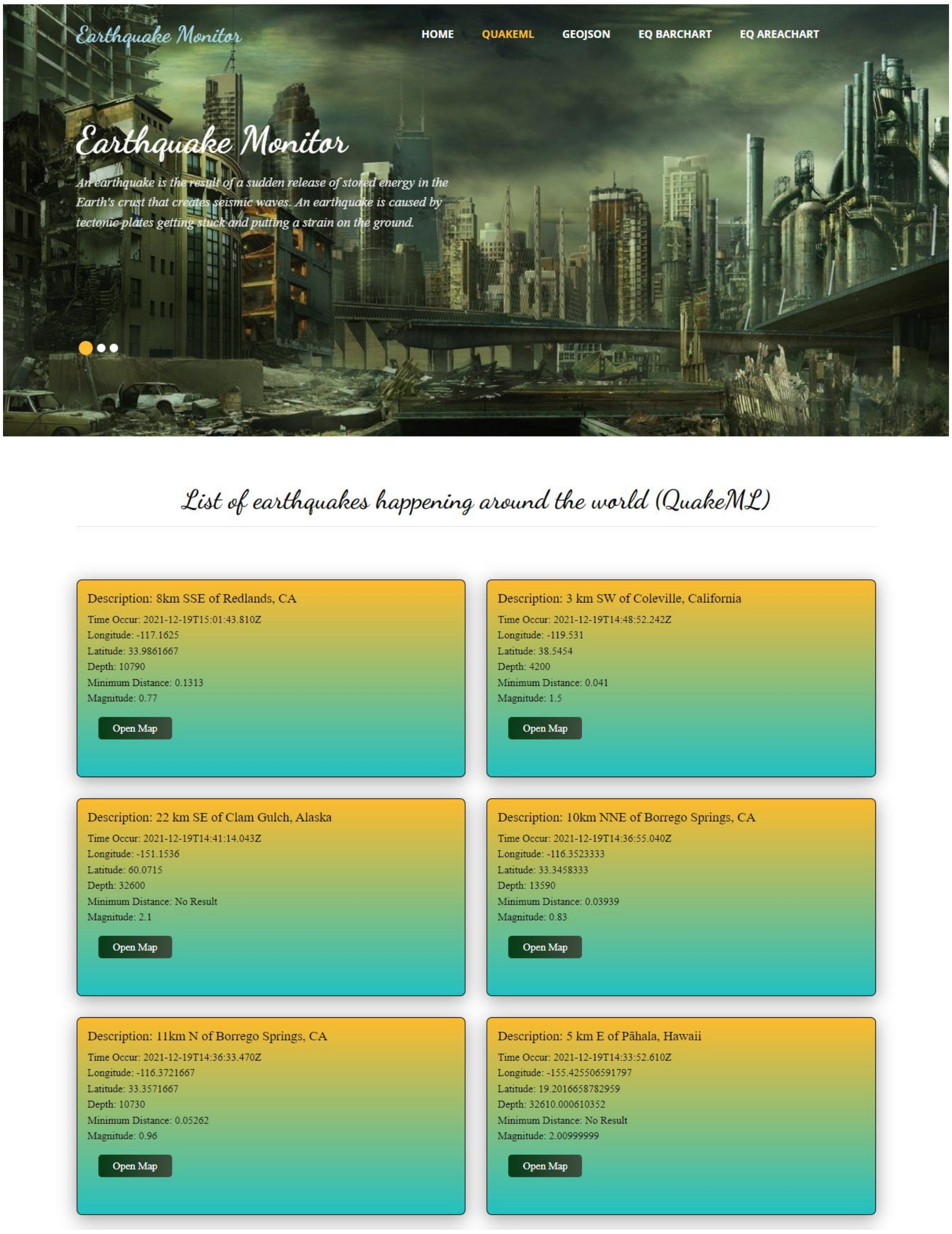
Earthquake Monitor - ***Beeharry Taajuddin*** *BSE20AFT*

Figure 1 shows a full snapshot of the homepage (index.php), which contains all earthquake-related information.

Figure 1 - Homepage (index.php)

Figure 2 shows the *navigation links* for the other pages in the website.	 Earthquake Monitor localhost/earthquake_monitor/index.php Earthquake Monitor HOME QUAKEML GEOJSON EQ BARCHART EQ AREACHART Get Started Earthquake Monitor An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground. *Figure 2 - Navigation Links & Localhost*

QuakeML page (quakeML.php)



An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground.

List of earthquakes happening around the world (QuakeML)

Description: 8km SSE of Redlands, CA
Time Occur: 2021-12-19T15:01:43.810Z
Longitude: -117.1625
Latitude: 33.9861667
Depth: 10790
Minimum Distance: 0.1313
Magnitude: 0.77

Description: 3 km SW of Coleville, California
Time Occur: 2021-12-19T14:48:52.242Z
Longitude: -119.531
Latitude: 38.5454
Depth: 4200
Minimum Distance: 0.041
Magnitude: 1.5

Description: 22 km SE of Clam Gulch, Alaska
Time Occur: 2021-12-19T14:41:14.043Z
Longitude: -151.1536
Latitude: 60.0715
Depth: 32600
Minimum Distance: No Result
Magnitude: 2.1

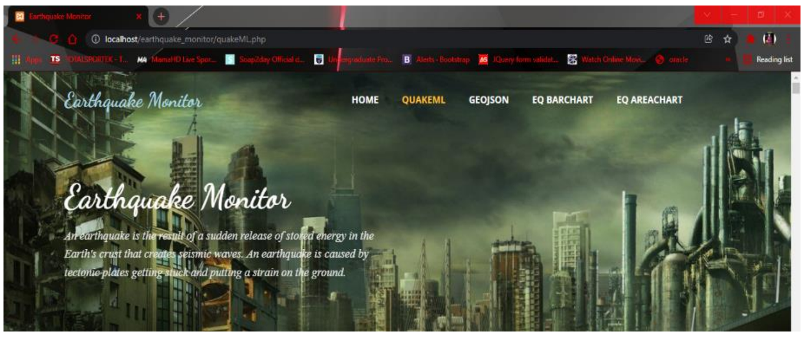
Description: 10km NNE of Borrego Springs, CA
Time Occur: 2021-12-19T14:36:55.040Z
Longitude: -116.3523333
Latitude: 33.3458333
Depth: 13590
Minimum Distance: 0.03939
Magnitude: 0.83

Description: 11km N of Borrego Springs, CA
Time Occur: 2021-12-19T14:36:33.470Z
Longitude: -116.3721667
Latitude: 33.3571667
Depth: 10730
Minimum Distance: 0.05262
Magnitude: 0.96

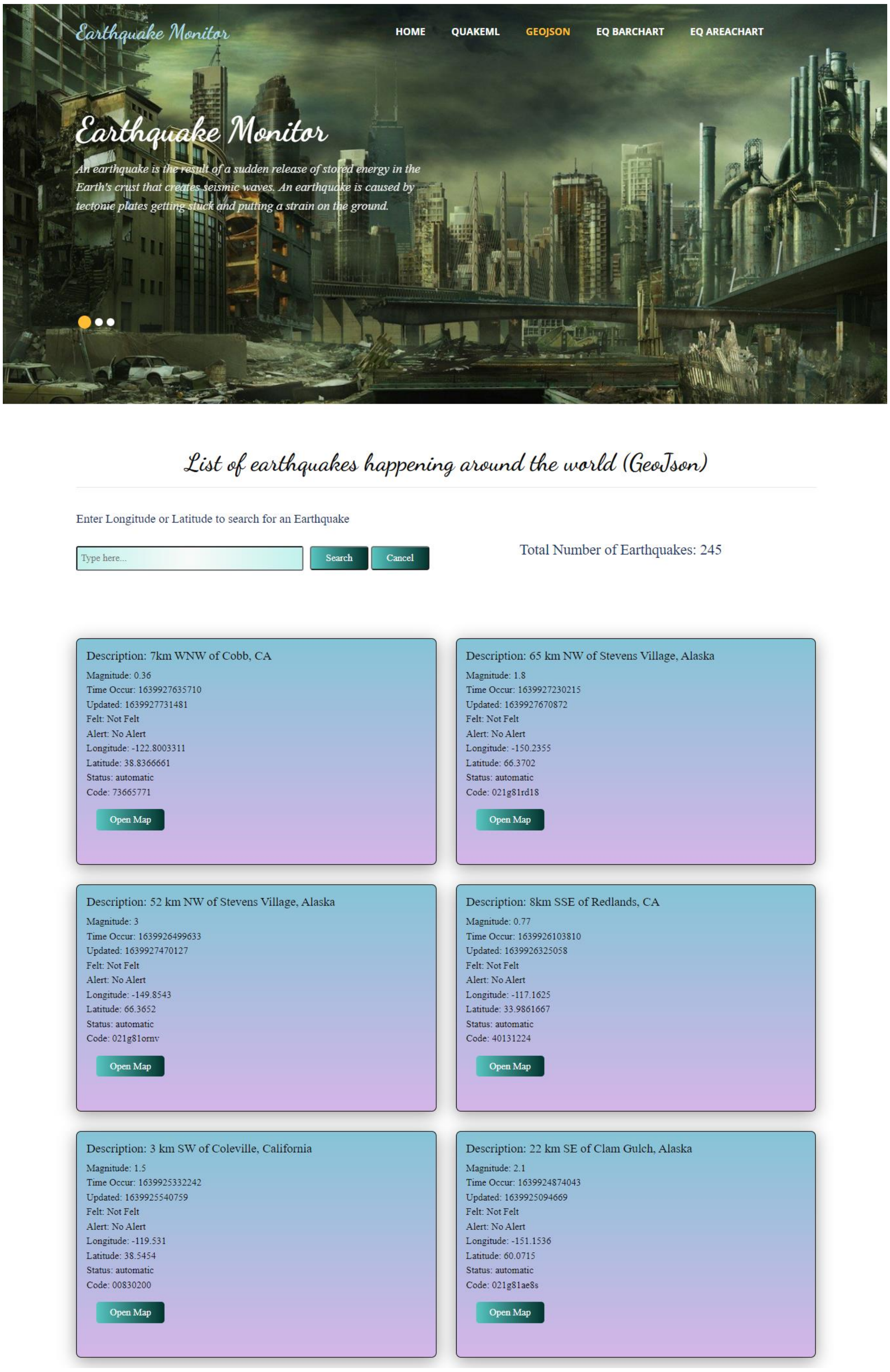
Description: 5 km E of Pāhala, Hawaii
Time Occur: 2021-12-19T14:33:52.610Z
Longitude: -155.425506591797
Latitude: 19.2016658782959
Depth: 32610.000610352
Minimum Distance: No Result
Magnitude: 2.00999999

Figure 3 - QuakeML page (quakeML.php)

Figure 3 shows the QuakeML page where all earthquake information is derived from an xml file. Also that, the occurrence of an earthquake can be seen. A button *'Open Map'* is available to view the earthquake location.

Figure 4 shows the *QuakeML page* is running successfully through the localhost.	 localhost/earthquake_monitor/quakeML.php Earthquake Monitor HOME QUAKEML GEOJSON EQ BARCHART EQ AREACHART Earthquake Monitor An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground. *Figure 4 - QuakeML localhost*

GeoJson page (geojson.php)



Earthquake Monitor

An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground.

List of earthquakes happening around the world (GeoJson)

Enter Longitude or Latitude to search for an Earthquake

Type here...

Total Number of Earthquakes: 245

Description: 7km WNW of Cobb, CA
Magnitude: 0.36
Time Occur: 1639927635710
Updated: 1639927731481
Felt: Not Felt
Alert: No Alert
Longitude: -122.8003311
Latitude: 38.8366661
Status: automatic
Code: 73665771

Description: 65 km NW of Stevens Village, Alaska
Magnitude: 1.8
Time Occur: 1639927230215
Updated: 1639927670872
Felt: Not Felt
Alert: No Alert
Longitude: -150.2355
Latitude: 66.3702
Status: automatic
Code: 021g81rd18

Description: 52 km NW of Stevens Village, Alaska
Magnitude: 3
Time Occur: 1639926499633
Updated: 1639927470127
Felt: Not Felt
Alert: No Alert
Longitude: -149.8543
Latitude: 66.3652
Status: automatic
Code: 021g81ornv

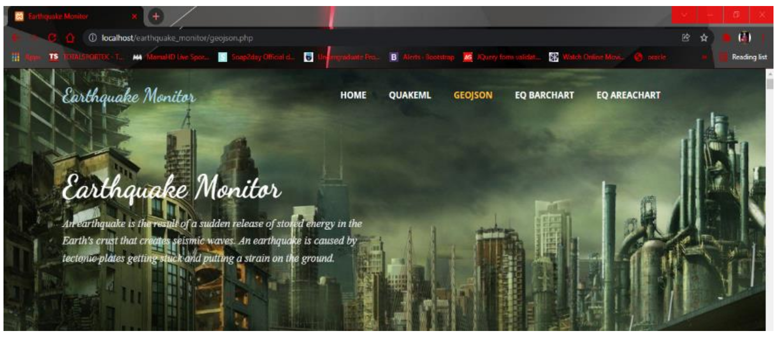
Description: 8km SSE of Redlands, CA
Magnitude: 0.77
Time Occur: 1639926103810
Updated: 1639926325058
Felt: Not Felt
Alert: No Alert
Longitude: -117.1625
Latitude: 33.9861667
Status: automatic
Code: 40131224

Description: 3 km SW of Coleville, California
Magnitude: 1.5
Time Occur: 1639925332242
Updated: 1639925540759
Felt: Not Felt
Alert: No Alert
Longitude: -119.531
Latitude: 38.5454
Status: automatic
Code: 00830200

Description: 22 km SE of Clam Gulch, Alaska
Magnitude: 2.1
Time Occur: 1639924874043
Updated: 1639925094669
Felt: Not Felt
Alert: No Alert
Longitude: -151.1536
Latitude: 60.0715
Status: automatic
Code: 021g81ae8s

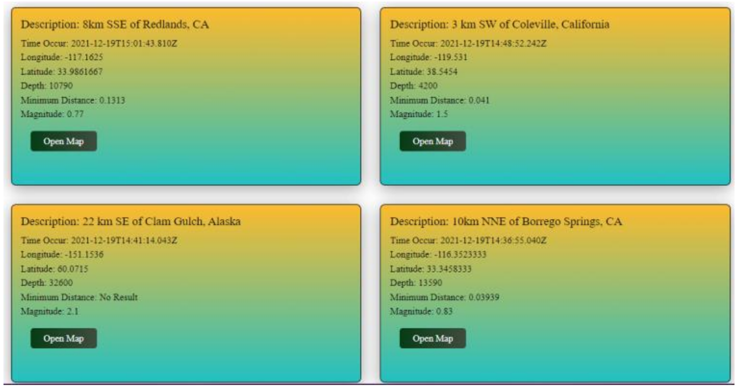
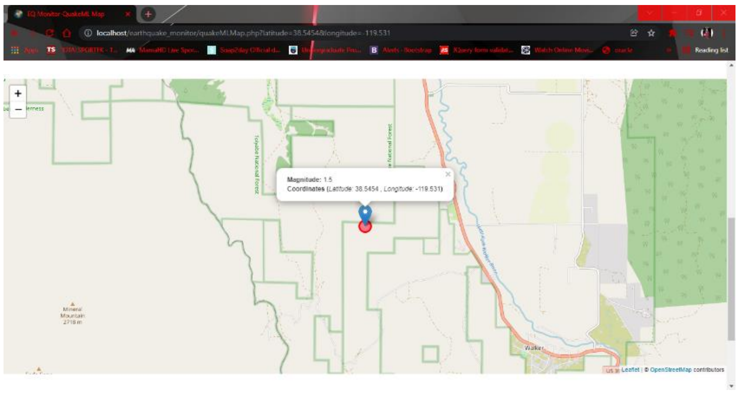
Figure 5 - GeoJson page (geojson.php)

Figure 5 shows the GeoJson page where all earthquake information is derived from a json file. Moreover, the occurrence of an earthquake can be seen and a button *'Open Map'* is available to view the earthquake location. The total number of earthquakes is available, as well as a search for an earthquake by its longitude or latitude is possible.

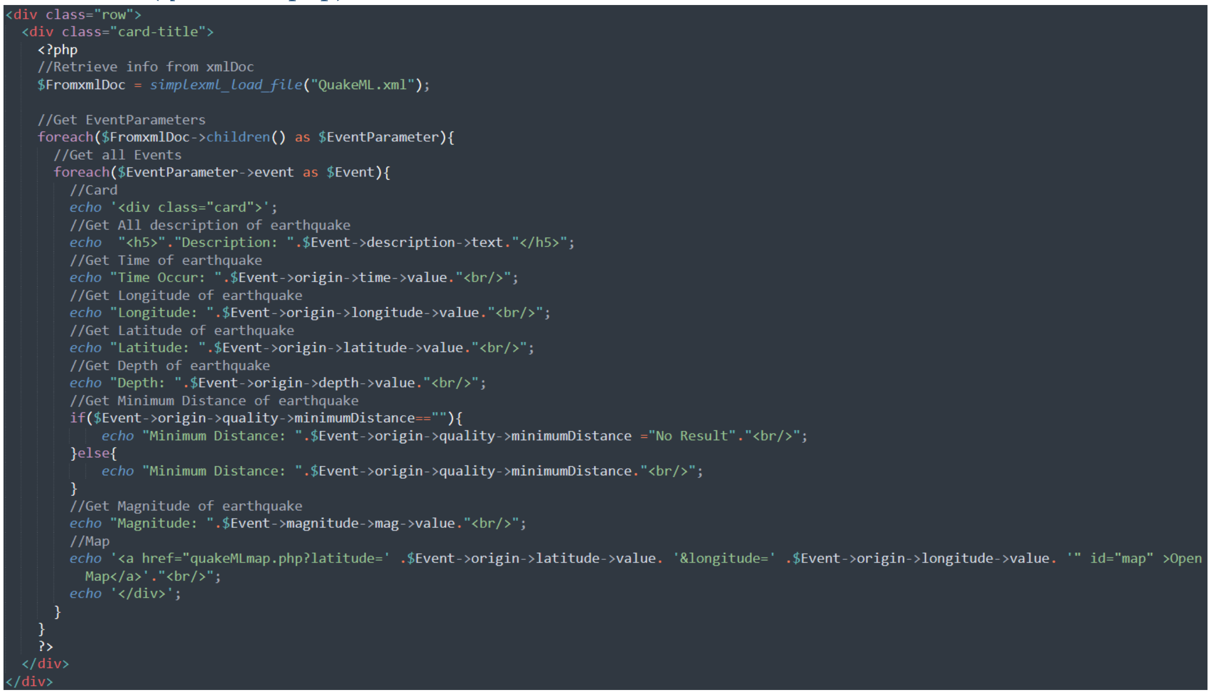
Figure 6 shows the *GeoJson page* is running successfully through the localhost.	 localhost/earthquake_monitor/geojson.php Earthquake Monitor HOME QUAKEML GEOJSON EQ BARCHART EQ AREACHART Earthquake Monitor *An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground.* *Figure 6 - GeoJson localhost*

Functionalities for XML

Functionality 1 – Open Map UI (quakeML.php)

Figure 7 shows the QuakeML page which provides earthquake details. A button *'Open Map'* is available to view the earthquake location.	 Description: 8km SSE of Redlands, CA Time Occur: 2021-12-19T15:01:43.810Z Longitude: -117.1625 Latitude: 33.9861667 Depth: 10790 Minimum Distance: 0.1313 Magnitude: 0.77 Description: 3 km SW of Coleville, California Time Occur: 2021-12-19T14:48:52.242Z Longitude: -119.531 Latitude: 38.5454 Depth: 4200 Minimum Distance: 0.041 Magnitude: 1.5 Description: 22 km SE of Clam Gulch, Alaska Time Occur: 2021-12-19T14:41:14.043Z Longitude: -151.1536 Latitude: 60.0715 Depth: 32600 Minimum Distance: No Result Magnitude: 2.1 Description: 10km NNE of Borrego Springs, CA Time Occur: 2021-12-19T14:36:55.040Z Longitude: -116.3523333 Latitude: 33.3458333 Depth: 13590 Minimum Distance: 0.03939 Magnitude: 0.83 *Figure 7 - Earthquake in XML page*
Once pressing the button *'Open Map'* as shown in figure 7, the user will redirect to map page (quakeMLmap.php) where the chosen earthquake's will be displayed as well as the magnitude and coordinates (longitude, latitude).	 Magnitude: 1.5 Coordinates (Latitude: 38.5454 , Longitude: -119.531) EQ Monitor-QuakeML Map Leaflet | © OpenStreetMap contributors *Figure 8 - Earthquake map location (xml page)*

XML Code (quakeML.php)



```
<div class="row">
  <div class="card-title">
    <?php
    //Retrieve info from xmlDoc
    $FromxmlDoc = simplexml_load_file("QuakeML.xml");

    //Get EventParameters
    foreach($FromxmlDoc->children() as $EventParameter){
      //Get all Events
      foreach($EventParameter->event as $Event){
        //Card
        echo '<div class="card">';
        //Get All description of earthquake
        echo  "<h5>"."Description: ".$Event->description->text."</h5>";
        //Get Time of earthquake
        echo "Time Occur: ".$Event->origin->time->value."<br/>";
        //Get Longitude of earthquake
        echo "Longitude: ".$Event->origin->longitude->value."<br/>";
        //Get Latitude of earthquake
        echo "Latitude: ".$Event->origin->latitude->value."<br/>";
        //Get Depth of earthquake
        echo "Depth: ".$Event->origin->depth->value."<br/>";
        //Get Minimum Distance of earthquake
        if($Event->origin->quality->minimumDistance==""){
            echo "Minimum Distance: ".$Event->origin->quality->minimumDistance ="No Result"."<br/>";
        }else{
            echo "Minimum Distance: ".$Event->origin->quality->minimumDistance."<br/>";
        }
        //Get Magnitude of earthquake
        echo "Magnitude: ".$Event->magnitude->mag->value."<br/>";
        //Map
        echo '<a href="quakeMLmap.php?latitude=' .$Event->origin->latitude->value. '&longitude=' .$Event->origin->longitude->value. '" id="map" >Open
          Map</a>'."<br/>";
        echo '</div>';
      }
    }
    ?>
  </div>
</div>
```

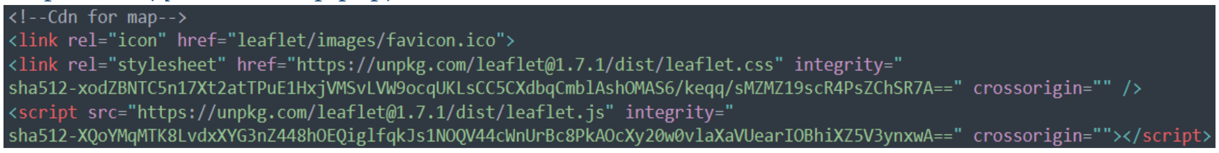
Figure 9 - xml code

The xml file was stored locally after being downloaded from the USGS website. It was used to retrieve the information and converted to an object using *Simple_load_file("QuakeML.xml")*. As seen in figure 9, the variable *$FromxmlDoc* was defined and is holding the xml file. Two foreach loop has been used to loop through the file. The first foreach loop is used to access EventParameters tag, which is a child of QuakeML and the second foreach loop is used to access the Event tag, which is a child of EventParameters.

For the minimum distance, an if else statement is used to check if the earthquake has a minimum distance. If no, 'No Result' will be displayed rather than a blank area else the minimum distance will be displayed.

In figure 9, the data of longitude and latitude were retrieved using query string and passed to quakeMLmap.php (map page) as shown in figure 8.

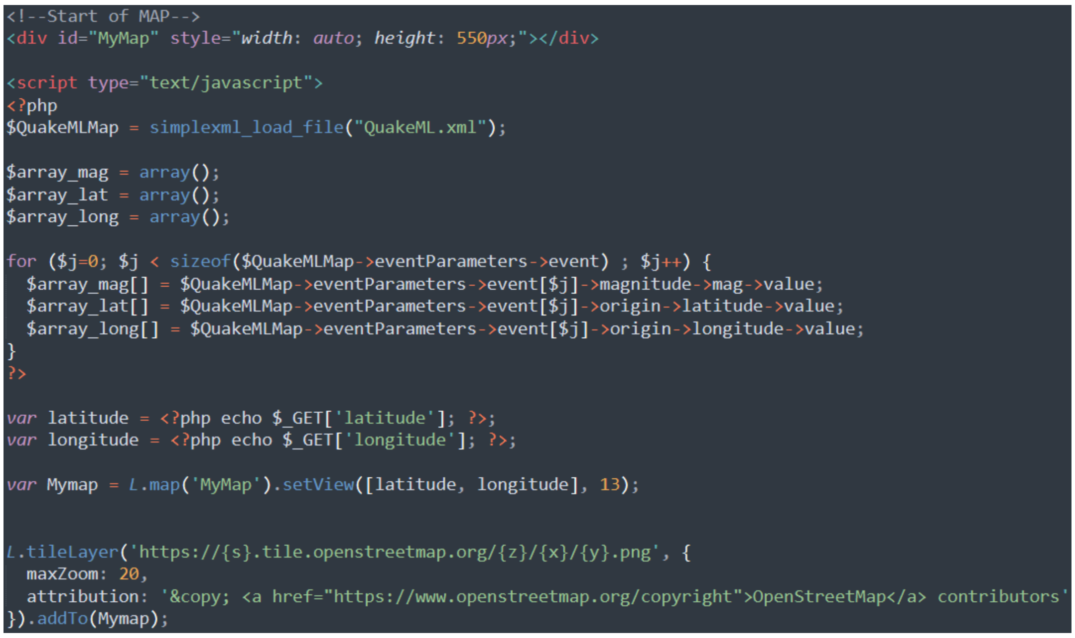
Map Code (quakeMLmap.php)



```
<!--Cdn for map-->
<link rel="icon" href="leaflet/images/favicon.ico">
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="
sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin="" />
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="
sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>
```

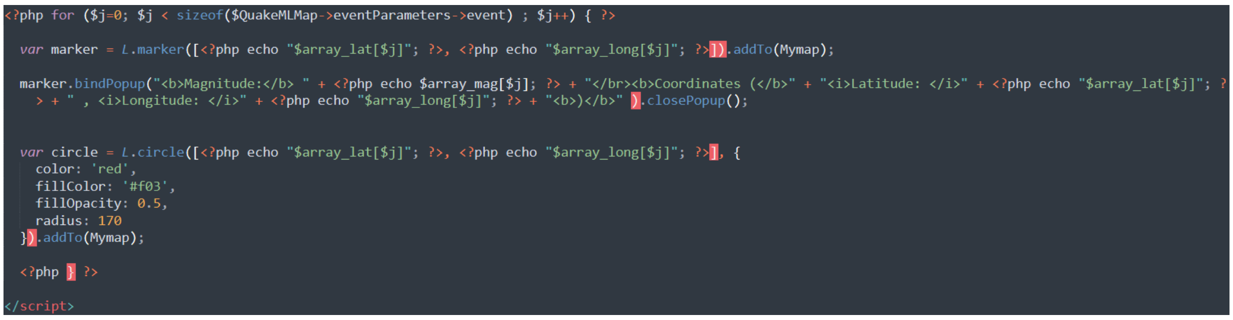
Figure 10 - CDN code for Map (xml page)

The CDN was used to illustrate the position of each earthquake in figure 10.



```
<!--Start of MAP-->
<div id="MyMap" style="width: auto; height: 550px;"></div>

<script type="text/javascript">
<?php
$QuakeMLMap = simplexml_load_file("QuakeML.xml");

$array_mag = array();
$array_lat = array();
$array_long = array();

for ($j=0; $j < sizeof($QuakeMLMap->eventParameters->event) ; $j++) {
  $array_mag[] = $QuakeMLMap->eventParameters->event[$j]->magnitude->mag->value;
  $array_lat[] = $QuakeMLMap->eventParameters->event[$j]->origin->latitude->value;
  $array_long[] = $QuakeMLMap->eventParameters->event[$j]->origin->longitude->value;
}
?>

var latitude = <?php echo $_GET['latitude']; ?>;
var longitude = <?php echo $_GET['longitude']; ?>;

var Mymap = L.map('MyMap').setView([latitude, longitude], 13);


L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 20,
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
}).addTo(Mymap);
```

Figure 11 - Code for Map (1)



```
<?php for ($j=0; $j < sizeof($QuakeMLMap->eventParameters->event) ; $j++) { ?>

  var marker = L.marker([<?php echo "$array_lat[$j]"; ?>, <?php echo "$array_long[$j]"; ?>]).addTo(Mymap);

  marker.bindPopup("<b>Magnitude:</b> " + <?php echo $array_mag[$j]; ?> + "</br><b>Coordinates (</b>" + "<i>Latitude: </i>" + <?php echo "$array_lat[$j]"; ?
    > + " , <i>Longitude: </i>" + <?php echo "$array_long[$j]"; ?> + "<b>)</b>" ).closePopup();


  var circle = L.circle([<?php echo "$array_lat[$j]"; ?>, <?php echo "$array_long[$j]"; ?>], {
    color: 'red',
    fillColor: '#f03',
    fillOpacity: 0.5,
    radius: 170
  }).addTo(Mymap);

  <?php } ?>

</script>
```

Figure 12 – Code for Map (2)

In figure 11, xml file was used to retrieve information and three empty arrays were declared to contain the magnitude, latitude and longitude. The querystring's latitude and longitude data were retrieved using the *$_GET[]* method.

Figure 12 shows a marker is used to show where the magnitude and coordinates (latitude and longitude) are located.

Functionality 2 – Using Charts

Bar Chat UI (chart.php)

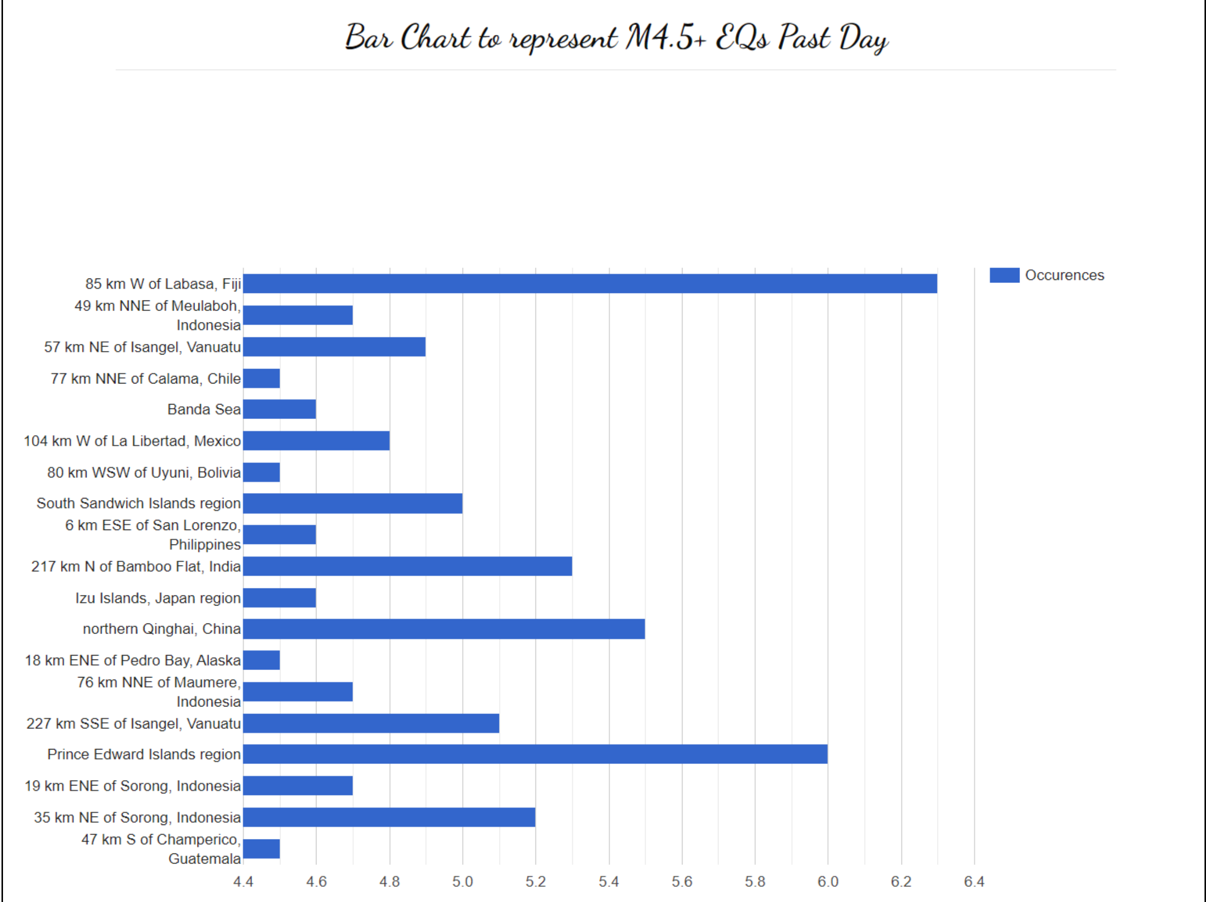
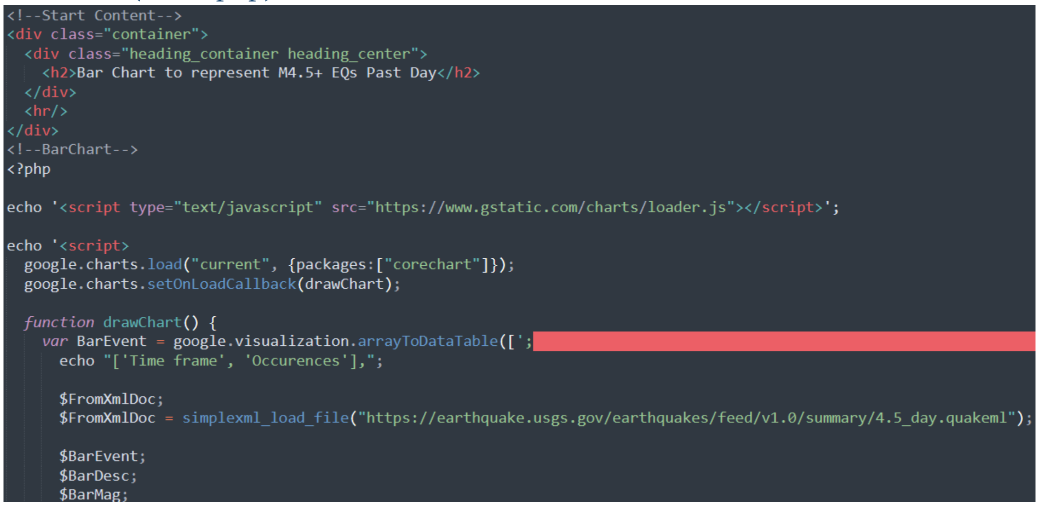


Figure 13 - Bar Chart M4.5+

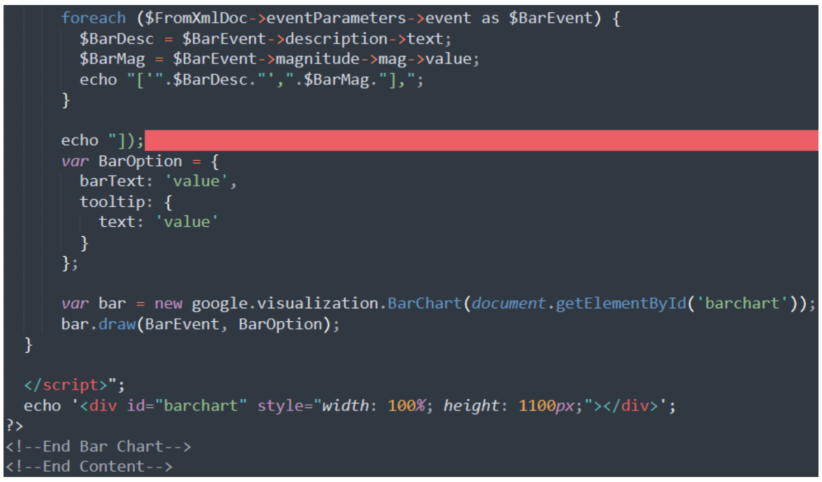
Figure 13 is a bar chart showing earthquakes with a **magnitude of 4.5+ for past day**. The earthquakes' names are displayed against its magnitude.

Bar Chart Code (chart.php)



```
<!--Start Content-->
<div class="container">
  <div class="heading_container heading_center">
    <h2>Bar Chart to represent M4.5+ EQs Past Day</h2>
  </div>
  <hr/>
</div>
<!--BarChart-->
<?php

echo '<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>';

echo '<script>
  google.charts.load("current", {packages:["corechart"]});
  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {
    var BarEvent = google.visualization.arrayToDataTable([';
      echo "['Time frame', 'Occurences'],";

      $FromXmlDoc;
      $FromXmlDoc = simplexml_load_file("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_day.quakeml");

      $BarEvent;
      $BarDesc;
      $BarMag;
```

Figure 14 - Bar Char Code (1) for M4.5+ Past Day



```
      foreach ($FromXmlDoc->eventParameters->event as $BarEvent) {
        $BarDesc = $BarEvent->description->text;
        $BarMag = $BarEvent->magnitude->mag->value;
        echo "['".$BarDesc."',".$BarMag."],";
      }

      echo "]);
      var BarOption = {
        barText: 'value',
        tooltip: {
          text: 'value'
        }
      };

      var bar = new google.visualization.BarChart(document.getElementById('barchart'));
      bar.draw(BarEvent, BarOption);
    }

  </script>";
  echo '<div id="barchart" style="width: 100%; height: 1100px;"></div>';
?>
<!--End Bar Chart-->
<!--End Content-->
```

Figure 15 - Bar Char Code (2) for M4.5+ Past Day

As illustrated in figure 14, a URL for **magnitude 4.5+ past day** from USGS website is utilized and google chart was used to create the bar chart. The *simple_load_file()* is used to parse the content in the xml as an object.

To cycle over the xml file, foreach loop has been used as shown in figure 15 to access the EventParameters tag, which is a child. In addition, the text and value which is the earthquake's description and magnitude has been extracted.

Area Chart UI (chart2.php)

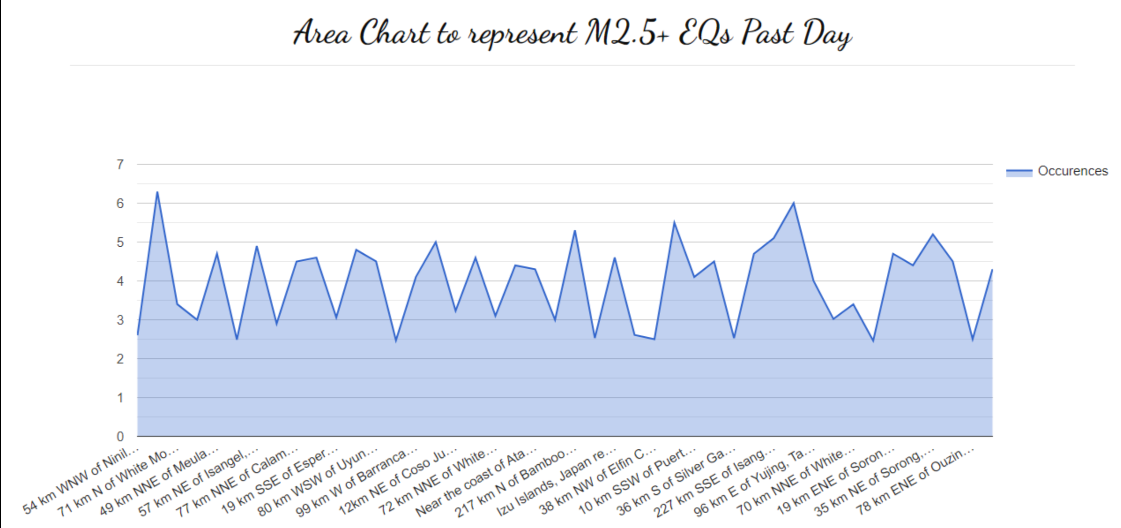
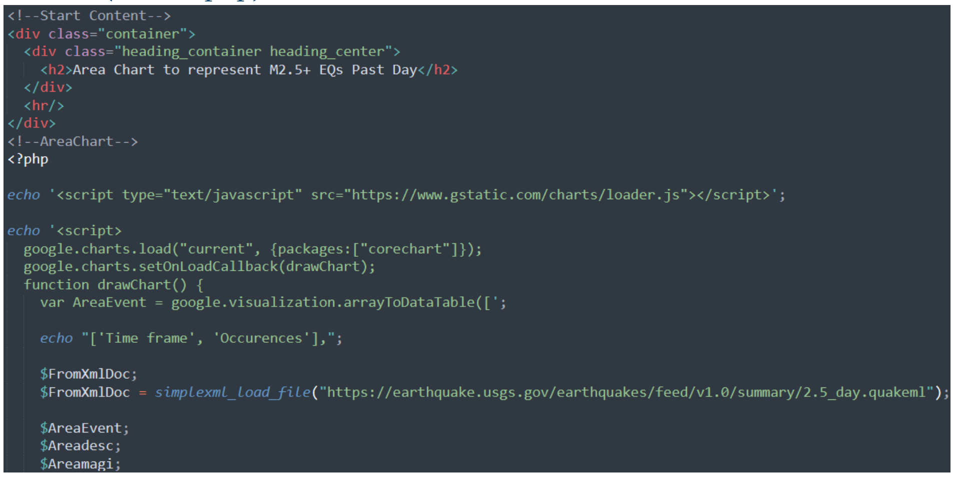


Figure 16 - Area Chart M2.5+

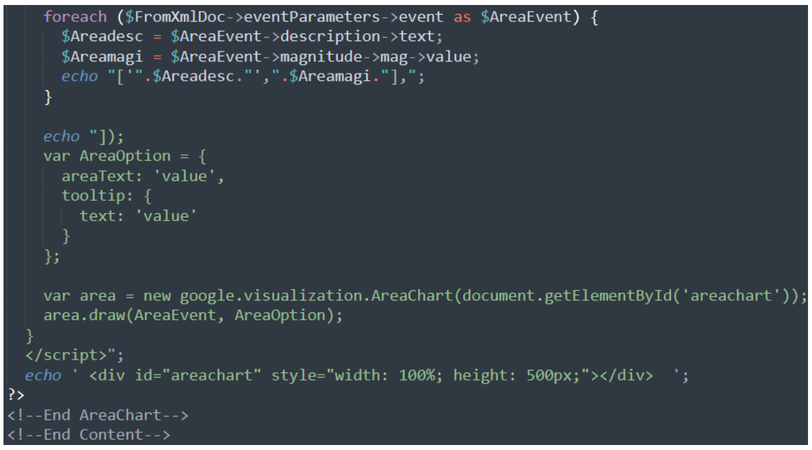
Figure 16 is an area chart showing earthquakes with a **magnitude of 2.5+ past day**. The magnitude is displayed against its earthquakes' names.

Area Chart Code (chart2.php)



```
<!--Start Content-->
<div class="container">
  <div class="heading_container heading_center">
    <h2>Area Chart to represent M2.5+ EQs Past Day</h2>
  </div>
  <hr/>
</div>
<!--AreaChart-->
<?php

echo '<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>';

echo '<script>
  google.charts.load("current", {packages:["corechart"]});
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var AreaEvent = google.visualization.arrayToDataTable([';

    echo "['Time frame', 'Occurences'],";

    $FromXmlDoc;
    $FromXmlDoc = simplexml_load_file("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_day.quakeml");

    $AreaEvent;
    $Areadesc;
    $Areamagi;
```

Figure 17 - Area Char Code (1) for M2.5+ Past Day



```
      foreach ($FromXmlDoc->eventParameters->event as $AreaEvent) {
        $Areadesc = $AreaEvent->description->text;
        $Areamagi = $AreaEvent->magnitude->mag->value;
        echo "['".$Areadesc."',".$Areamagi."],";
      }

      echo "]);
      var AreaOption = {
        areaText: 'value',
        tooltip: {
          text: 'value'
        }
      };

      var area = new google.visualization.AreaChart(document.getElementById('areachart'));
      area.draw(AreaEvent, AreaOption);
    }
    </script>";
    echo ' <div id="areachart" style="width: 100%; height: 500px;"></div>  ';
?>
<!--End AreaChart-->
<!--End Content-->
```

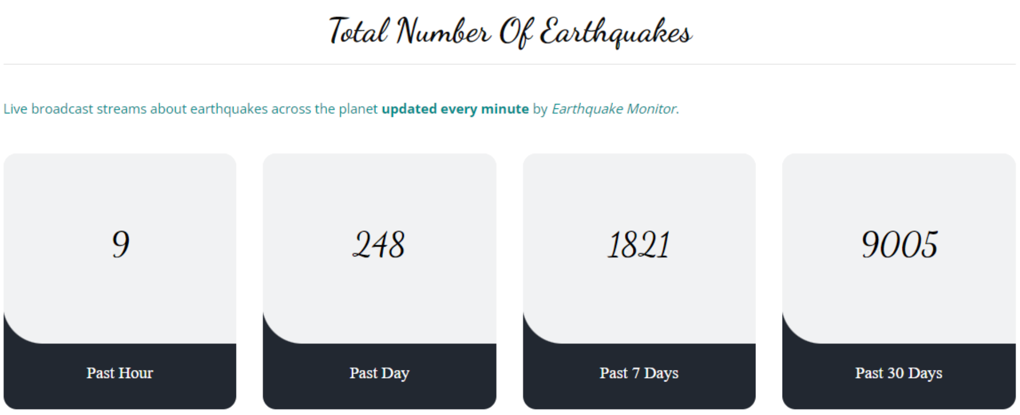
Figure 18 - Area Char Code (2) for M2.5+ Past Day

As illustrated in figure 17, a URL for **magnitude 2.5+ for past day** from USGS website is utilized and google chart was used to create the area chart. The *simple_load_file()* is used to parse the content in the xml as an object.

To cycle over the xml file, foreach loop has been used as shown in figure 18 to access the EventParameters tag, which is a child. In addition, the text and value which is the earthquake's description and magnitude has been extracted.

Functionalities for GeoJson

Functionality 1 – Count UI (index.php)


Total Number Of Earthquakes

Live broadcast streams about earthquakes across the planet **updated every minute** by *Earthquake Monitor.*

9	248	1821	9005
Past Hour	Past Day	Past 7 Days	Past 30 Days

Figure 19 - Total Number of Earthquakes

In figure 19, it shows the total number of earthquakes that have occurred in the last hour, day, week and month.

Count Code (index.php)

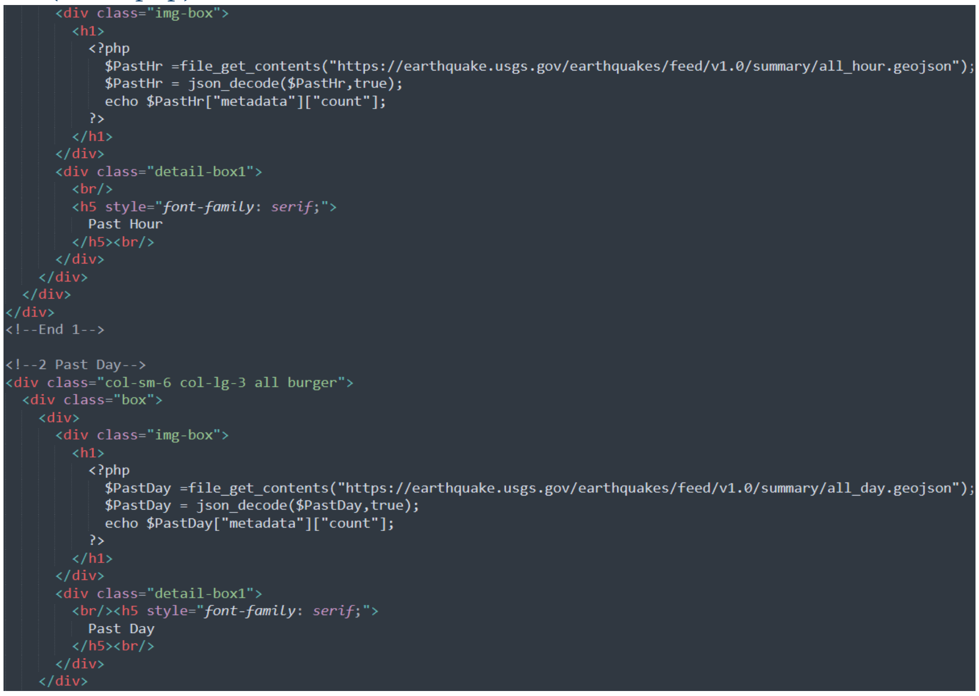
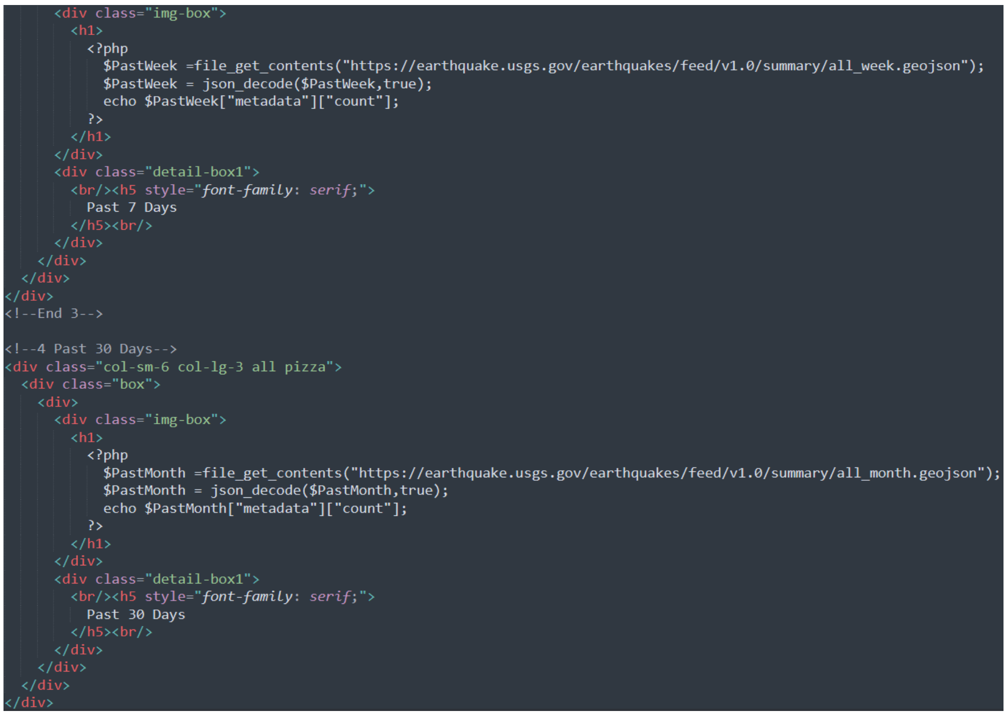

```
      <div class="img-box">
        <h1>
          <?php
            $PastHr =file_get_contents("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson");
            $PastHr = json_decode($PastHr,true);
            echo $PastHr["metadata"]["count"];
          ?>
        </h1>
      </div>
      <div class="detail-box1">
        <br/>
        <h5 style="font-family: serif;">
          Past Hour
        </h5><br/>
      </div>
    </div>
  </div>
</div>
<!--End 1-->

<!--2 Past Day-->
<div class="col-sm-6 col-lg-3 all burger">
  <div class="box">
    <div>
      <div class="img-box">
        <h1>
          <?php
            $PastDay =file_get_contents("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson");
            $PastDay = json_decode($PastDay,true);
            echo $PastDay["metadata"]["count"];
          ?>
        </h1>
      </div>
      <div class="detail-box1">
        <br/><h5 style="font-family: serif;">
          Past Day
        </h5><br/>
      </div>
    </div>
```

Figure 20 - Count code (1)

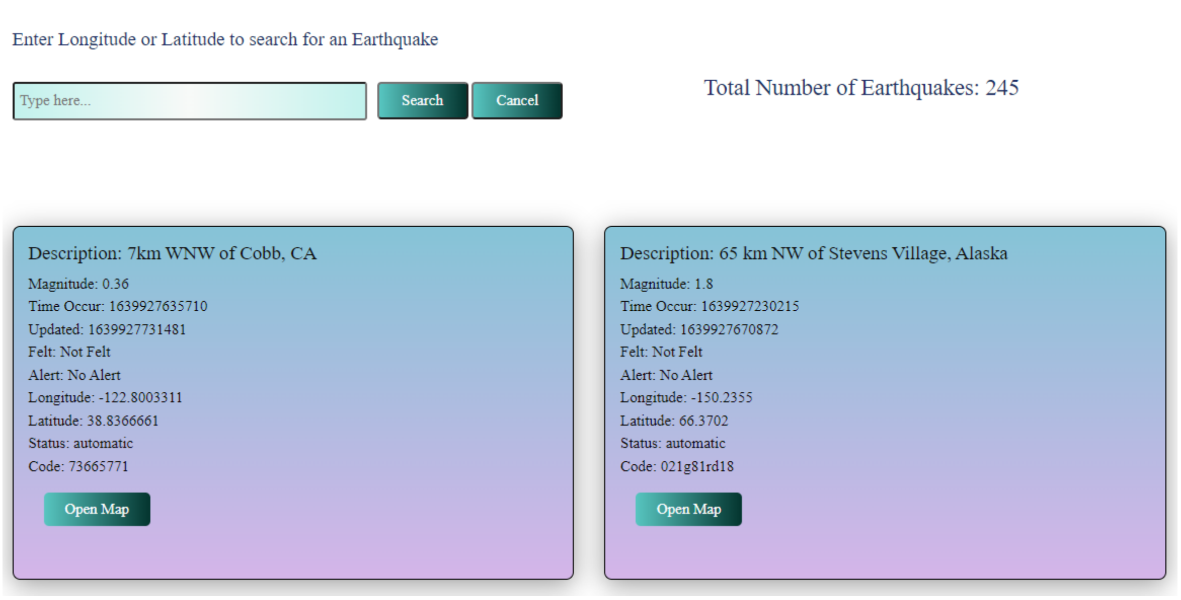


```
      <div class="img-box">
        <h1>
          <?php
            $PastWeek =file_get_contents("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_week.geojson");
            $PastWeek = json_decode($PastWeek,true);
            echo $PastWeek["metadata"]["count"];
          ?>
        </h1>
      </div>
      <div class="detail-box1">
        <br/><h5 style="font-family: serif;">
          Past 7 Days
        </h5><br/>
      </div>
    </div>
  </div>
</div>
<!--End 3-->

<!--4 Past 30 Days-->
<div class="col-sm-6 col-lg-3 all pizza">
  <div class="box">
    <div>
      <div class="img-box">
        <h1>
          <?php
            $PastMonth =file_get_contents("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.geojson");
            $PastMonth = json_decode($PastMonth,true);
            echo $PastMonth["metadata"]["count"];
          ?>
        </h1>
      </div>
      <div class="detail-box1">
        <br/><h5 style="font-family: serif;">
          Past 30 Days
        </h5><br/>
      </div>
    </div>
  </div>
</div>
```

Figure 21 - Count code (2)

Figure 20 and 21 show the code for displaying the total number of earthquakes and how URLs were utilized from USGS website in the *file_get_contents()* function to display the overall number of earthquakes. The data is retrieved by using the *json_decode()* function. Thus, by using these URLs, the website can keep up to date.

Functionality 2 – Search Bar UI (geojson.php)


Enter Longitude or Latitude to search for an Earthquake

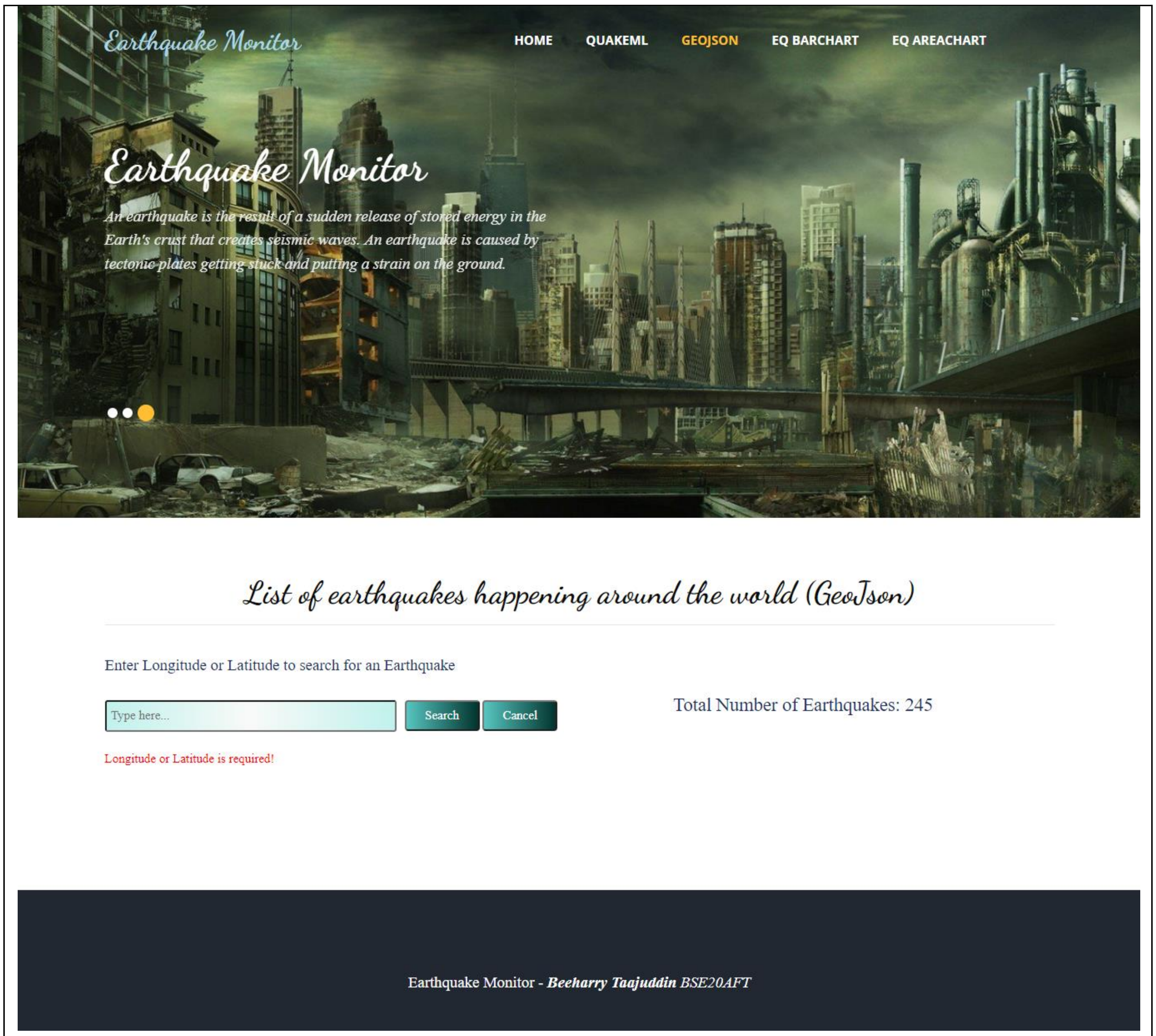
Total Number of Earthquakes: 245

Description: 7km WNW of Cobb, CA
Magnitude: 0.36
Time Occur: 1639927635710
Updated: 1639927731481
Felt: Not Felt
Alert: No Alert
Longitude: -122.8003311
Latitude: 38.8366661
Status: automatic
Code: 73665771

Description: 65 km NW of Stevens Village, Alaska
Magnitude: 1.8
Time Occur: 1639927230215
Updated: 1639927670872
Felt: Not Felt
Alert: No Alert
Longitude: -150.2355
Latitude: 66.3702
Status: automatic
Code: 021g81rd18

Figure 22 - Search bar Json page

Figure 22 shows the implementation which is a search and in addition to that, a button *'Open Map'* is available to view the earthquake location. At the right side, it shows the total number of earthquakes occurred.



Earthquake Monitor

An earthquake is the result of a sudden release of stored energy in the Earth's crust that creates seismic waves. An earthquake is caused by tectonic plates getting stuck and putting a strain on the ground.

List of earthquakes happening around the world (GeoJson)

Enter Longitude or Latitude to search for an Earthquake

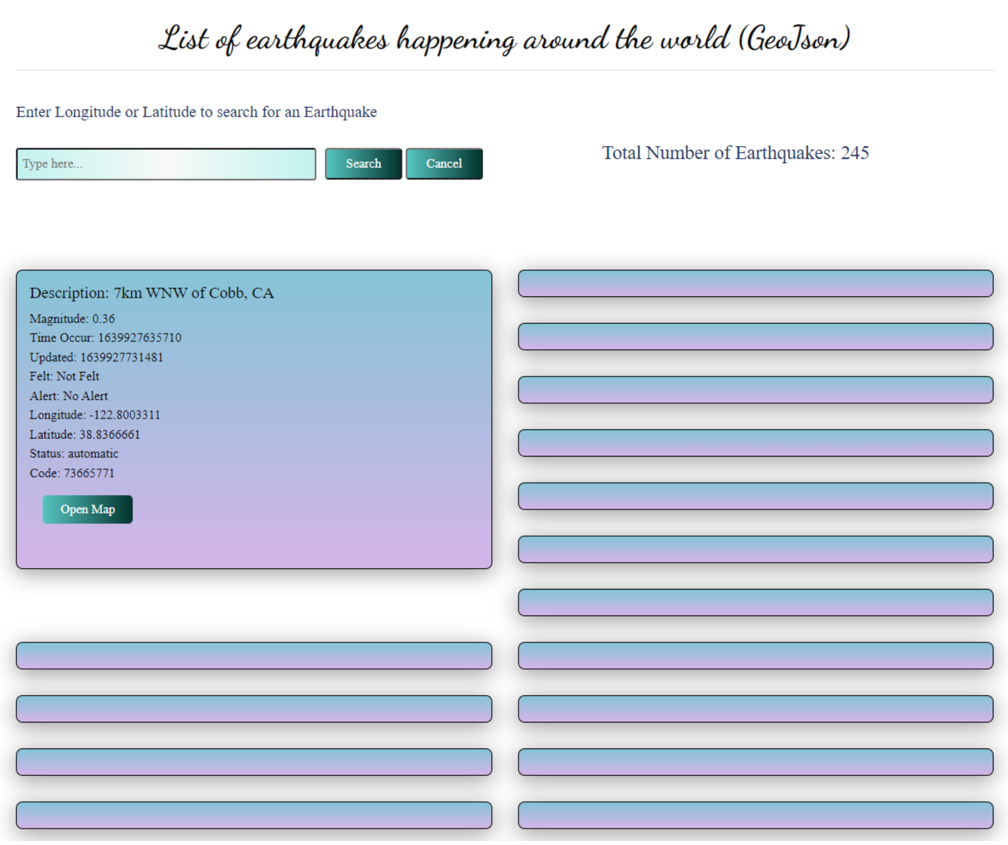
Total Number of Earthquakes: 245

Longitude or Latitude is required!

Earthquake Monitor - ***Beeharry Taajuddin*** *BSE20AFT*

Figure 23 - Search Error Json page

When a user pressed the search button even though the search box is empty, an error message will appear as shown in figure 23. If a user tries to put a non-existing coordinates, nothing will be displayed.


List of earthquakes happening around the world (GeoJson)

Enter Longitude or Latitude to search for an Earthquake

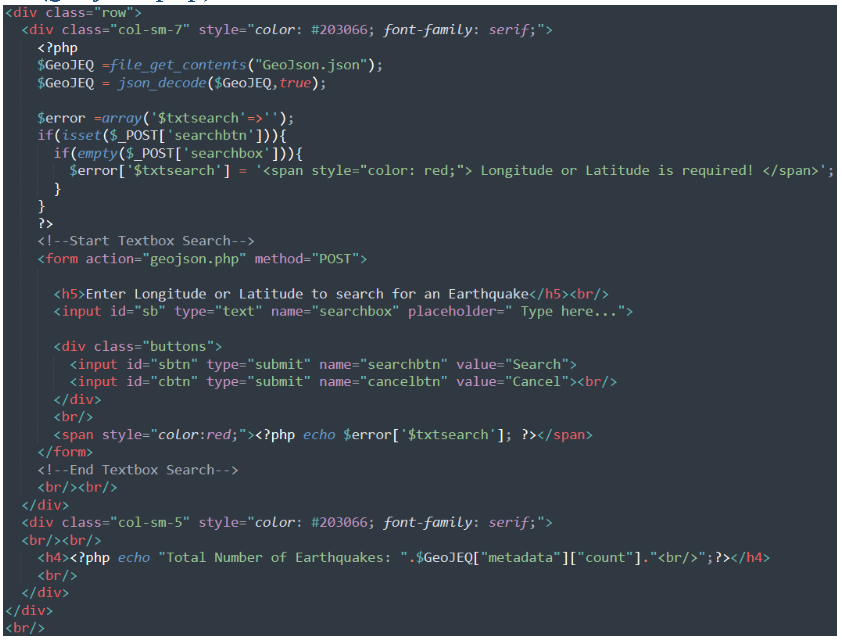
Total Number of Earthquakes: 245

Description: 7km WNW of Cobb, CA
Magnitude: 0.36
Time Occur: 1639927635710
Updated: 1639927731481
Felt: Not Felt
Alert: No Alert
Longitude: -122.8003311
Latitude: 38.8366661
Status: automatic
Code: 73665771

Figure 24 - Successfully search result Json page

Only the earthquake will be visible after searching for it by its latitude or longitude as shown in figure 24.

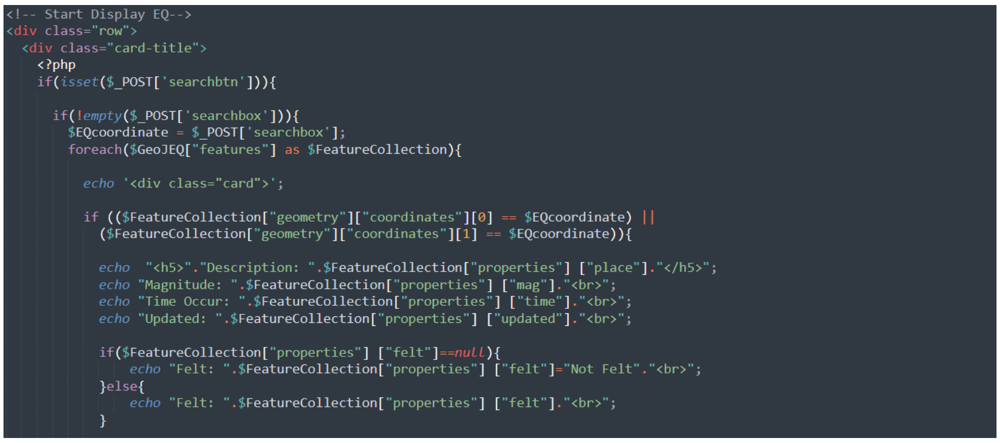
Search Bar code (geojson.php)



```
<div class="row">
  <div class="col-sm-7" style="color: #203066; font-family: serif;">
    <?php
    $GeoJEQ =file_get_contents("GeoJson.json");
    $GeoJEQ = json_decode($GeoJEQ,true);

    $error =array('$txtsearch'=>'');
    if(isset($_POST['searchbtn'])){
      if(empty($_POST['searchbox'])){
        $error['$txtsearch'] = '<span style="color: red;"> Longitude or Latitude is required! </span>';
      }
    }
    ?>
    <!--Start Textbox Search-->
    <form action="geojson.php" method="POST">

      <h5>Enter Longitude or Latitude to search for an Earthquake</h5><br/>
      <input id="sb" type="text" name="searchbox" placeholder=" Type here...">

      <div class="buttons">
        <input id="sbtn" type="submit" name="searchbtn" value="Search">
        <input id="cbtn" type="submit" name="cancelbtn" value="Cancel"><br/>
      </div>
      <br/>
      <span style="color:red;"><?php echo $error['$txtsearch']; ?></span>
    </form>
    <!--End Textbox Search-->
    <br/><br/>
  </div>
  <div class="col-sm-5" style="color: #203066; font-family: serif;">
  <br/><br/>
    <h4><?php echo "Total Number of Earthquakes: ".$GeoJEQ["metadata"]["count"]."<br/>";?></h4>
    <br/>
  </div>
</div>
<br/>
```

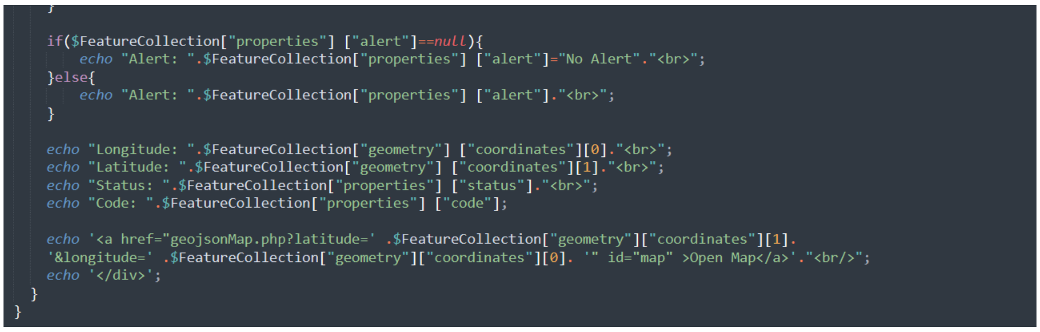
Figure 25 - Decode function and Count (Total EQs) - Json page

The json file was stored locally after being downloaded from the USGS website. The *file_get_contents()* method reads a file's contents into a string and then the *json_decode()* function transforms JSON to PHP and turns the file to an array when *true* is included as illustrates in figure 25.



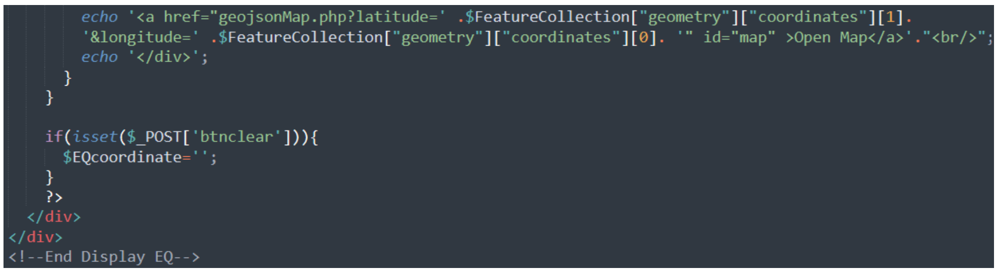
```
<!-- Start Display EQ-->
<div class="row">
  <div class="card-title">
    <?php
    if(isset($_POST['searchbtn'])){

      if(!empty($_POST['searchbox'])){
        $EQcoordinate = $_POST['searchbox'];
        foreach($GeoJEQ["features"] as $FeatureCollection){

          echo '<div class="card">';

          if (($FeatureCollection["geometry"]["coordinates"][0] == $EQcoordinate) ||
            ($FeatureCollection["geometry"]["coordinates"][1] == $EQcoordinate)){

            echo  "<h5>"."Description: ".$FeatureCollection["properties"] ["place"]."</h5>";
            echo "Magnitude: ".$FeatureCollection["properties"] ["mag"]."<br>";
            echo "Time Occur: ".$FeatureCollection["properties"] ["time"]."<br>";
            echo "Updated: ".$FeatureCollection["properties"] ["updated"]."<br>";

            if($FeatureCollection["properties"] ["felt"]==null){
                echo "Felt: ".$FeatureCollection["properties"] ["felt"]="Not Felt"."<br>";
            }else{
                echo "Felt: ".$FeatureCollection["properties"] ["felt"]."<br>";
            }
```

Figure 26 - Search code & Display EQs(1) (Json page)



```
    }

    if($FeatureCollection["properties"] ["alert"]==null){
        echo "Alert: ".$FeatureCollection["properties"] ["alert"]="No Alert"."<br>";
    }else{
        echo "Alert: ".$FeatureCollection["properties"] ["alert"]."<br>";
    }

    echo "Longitude: ".$FeatureCollection["geometry"] ["coordinates"][0]."<br>";
    echo "Latitude: ".$FeatureCollection["geometry"] ["coordinates"][1]."<br>";
    echo "Status: ".$FeatureCollection["properties"] ["status"]."<br>";
    echo "Code: ".$FeatureCollection["properties"] ["code"];

    echo '<a href="geojsonMap.php?latitude=' .$FeatureCollection["geometry"]["coordinates"][1].
    '&longitude=' .$FeatureCollection["geometry"]["coordinates"][0]. '" id="map" >Open Map</a>'."<br/>";
    echo '</div>';
  }
}
```

Figure 27 - Search code & Display EQs(2) (Json page)



```
          echo '<a href="geojsonMap.php?latitude=' .$FeatureCollection["geometry"]["coordinates"][1].
          '&longitude=' .$FeatureCollection["geometry"]["coordinates"][0]. '" id="map" >Open Map</a>'."<br/>";
          echo '</div>';
        }
      }

    if(isset($_POST['btnclear'])){
      $EQcoordinate='';
    }
    ?>
  </div>
</div>
<!--End Display EQ-->
```

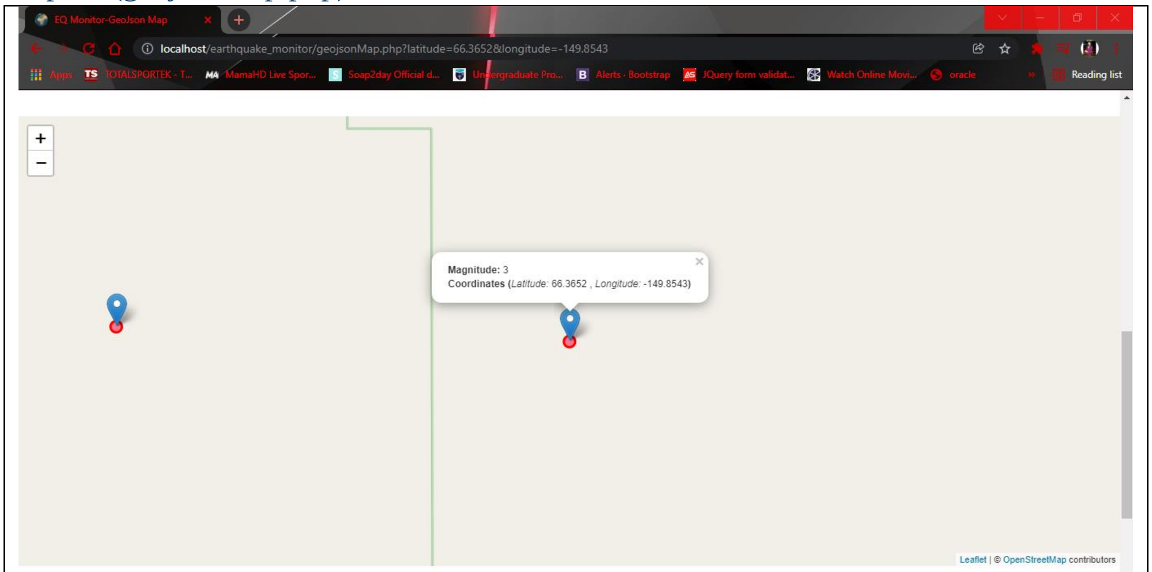
Figure 28 - Search code & Display EQs(3) (Json page)

Each result is displayed in square brackets in an array format after the data is retrieved from the json file.

In figure 26 and 27, for the **felt** and **alert**, an if else statement is used to check if the earthquake has been felt/ alert. If no, *'Not Felt/No Alert'* will be displayed rather than a blank area else the value felt will be displayed as well as the alert.

The input tag, search and cancel buttons were used to perform the search. Figure 26 shows how the page posts back to itself for search purposes.

Map UI (geojsonMap.php)

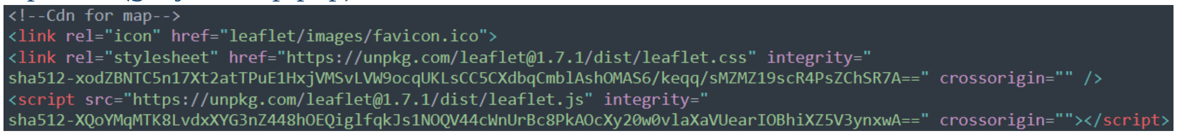


Magnitude: 3
Coordinates (*Latitude*: 66.3652 , *Longitude*: -149.8543)

Figure 29 - Earthquake map location (Json page)

Figure 29 shows an earthquake map location (geojsonMap.php) and when pressing on the marker the magnitude as well as its coordinates (longitude, latitude) appears.

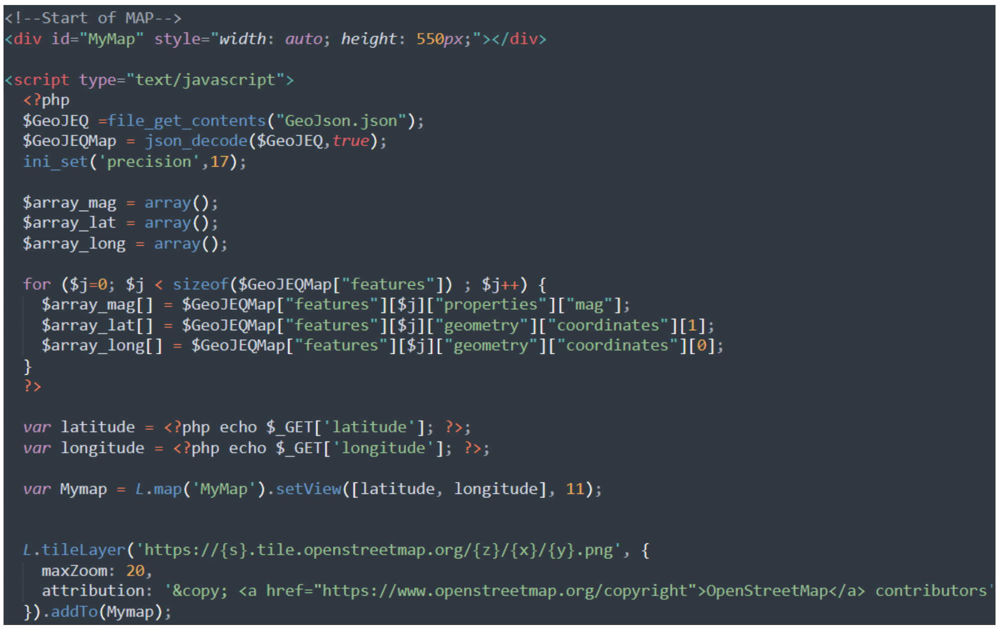
Map code (geojsonMap.php)



```
<!--Cdn for map-->
<link rel="icon" href="leaflet/images/favicon.ico">
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="
sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin="" />
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="
sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>
```

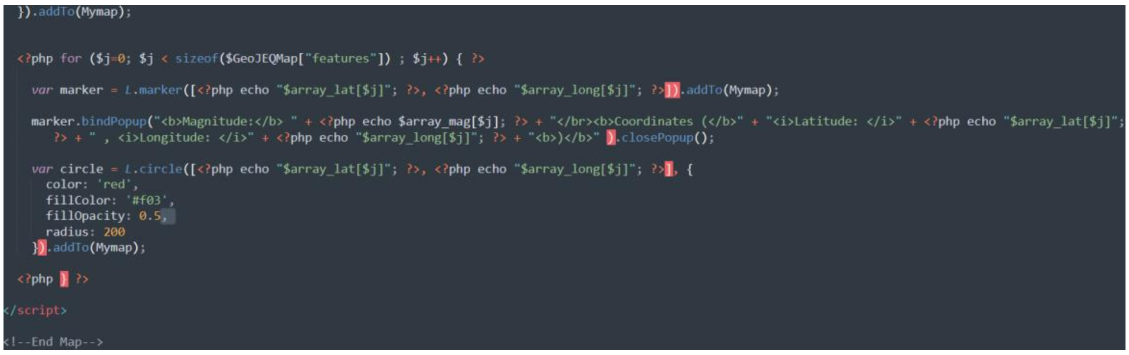
Figure 30 - CDN code for Map (json page)

The CDN was used to illustrate the position of each earthquake in figure 30.



```
<!--Start of MAP-->
<div id="MyMap" style="width: auto; height: 550px;"></div>

<script type="text/javascript">
  <?php
  $GeoJEQ =file_get_contents("GeoJson.json");
  $GeoJEQMap = json_decode($GeoJEQ,true);
  ini_set('precision',17);

  $array_mag = array();
  $array_lat = array();
  $array_long = array();

  for ($j=0; $j < sizeof($GeoJEQMap["features"]) ; $j++) {
    $array_mag[] = $GeoJEQMap["features"][$j]["properties"]["mag"];
    $array_lat[] = $GeoJEQMap["features"][$j]["geometry"]["coordinates"][1];
    $array_long[] = $GeoJEQMap["features"][$j]["geometry"]["coordinates"][0];
  }
  ?>

  var latitude = <?php echo $_GET['latitude']; ?>;
  var longitude = <?php echo $_GET['longitude']; ?>;

  var Mymap = L.map('MyMap').setView([latitude, longitude], 11);


  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 20,
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
  }).addTo(Mymap);
```

Figure 31 - Code for Map (1)



```
  }).addTo(Mymap);


  <?php for ($j=0; $j < sizeof($GeoJEQMap["features"]) ; $j++) { ?>

    var marker = L.marker([<?php echo "$array_lat[$j]"; ?>, <?php echo "$array_long[$j]"; ?>]).addTo(Mymap);

    marker.bindPopup("<b>Magnitude:</b> " + <?php echo $array_mag[$j]; ?> + "</br><b>Coordinates (</b>" + "<i>Latitude: </i>" + <?php echo "$array_lat[$j]";
      ?> + " , <i>Longitude: </i>" + <?php echo "$array_long[$j]"; ?> + "<b>)</b>" ).closePopup();

    var circle = L.circle([<?php echo "$array_lat[$j]"; ?>, <?php echo "$array_long[$j]"; ?>], {
      color: 'red',
      fillColor: '#f03',
      fillOpacity: 0.5,
      radius: 200
    }).addTo(Mymap);

  <?php } ?>

</script>

<!--End Map-->
```

Figure 32 - Code for Map (2)

In figure 31, json file was used to retrieve information and three empty arrays were declared to contain the magnitude, latitude and longitude. The querystring's latitude and longitude data were retrieved using the *$_GET[]* method.

Figure 32 shows a marker is used to show where the magnitude and coordinates (latitude and longitude) are located.