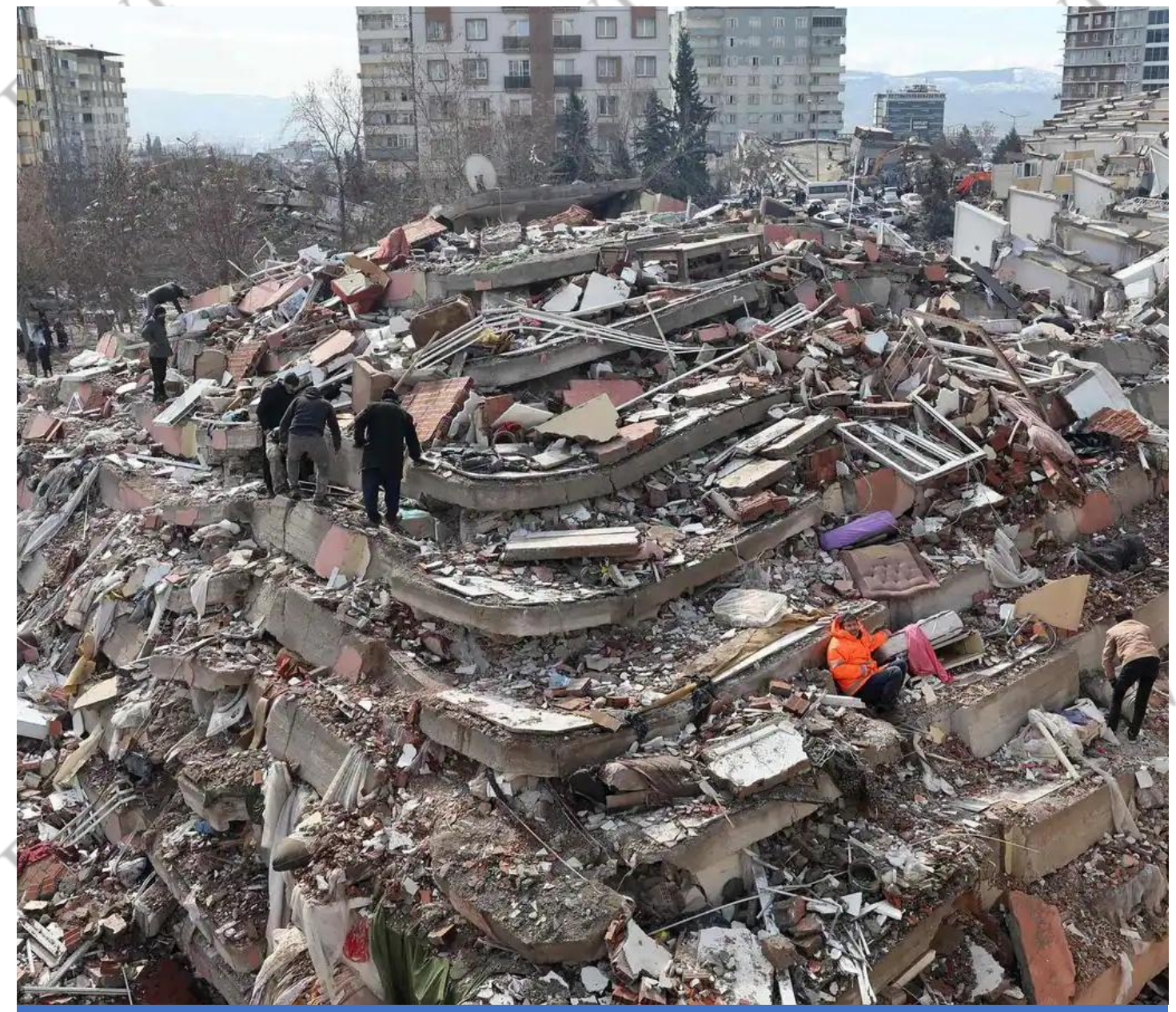


JEETOO STUART

BSE21BFT2-2210_22803

WEB SERVICES

stuartjeetoo@gmail.com

7/3/23

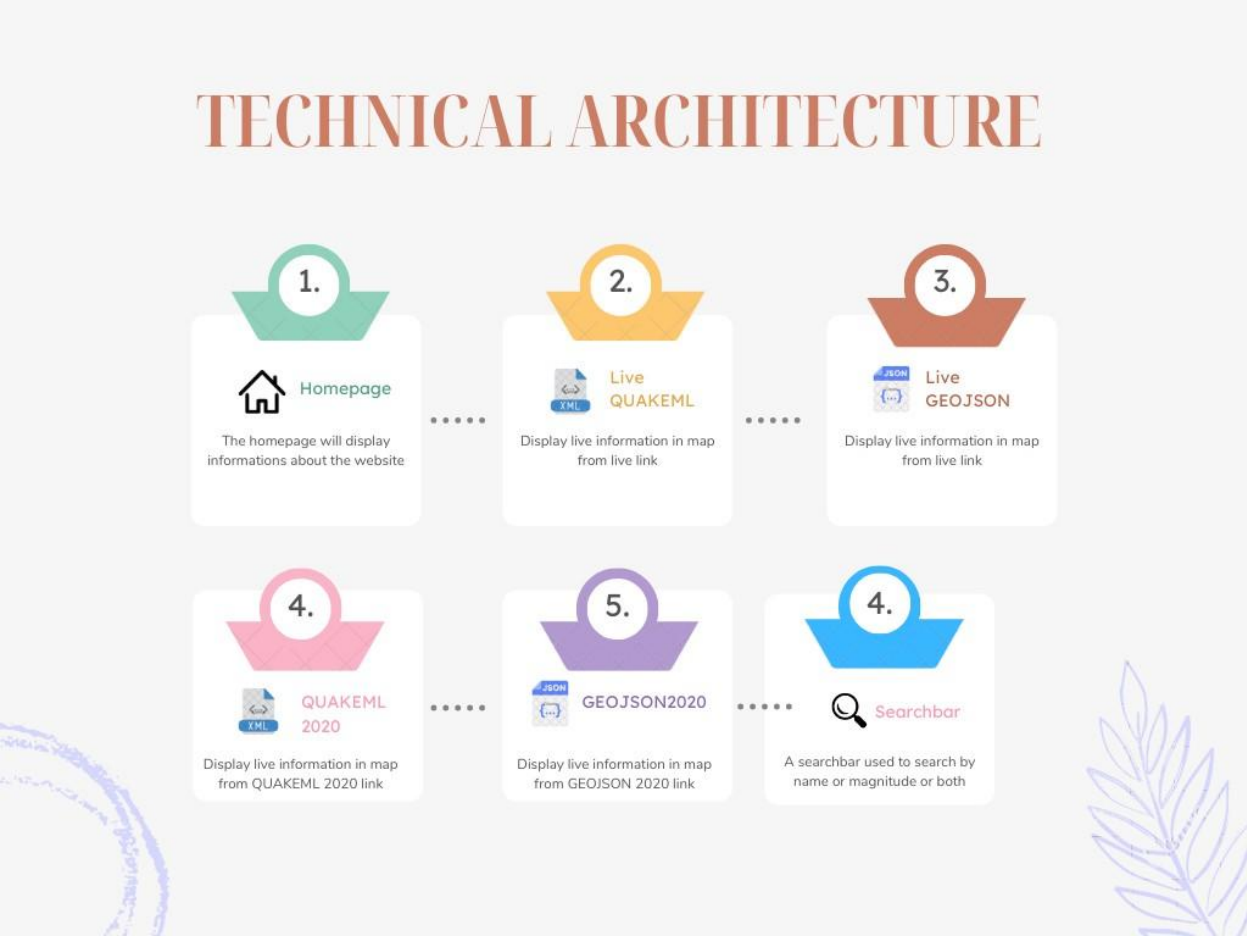
Working with XML & JSON Data Formats

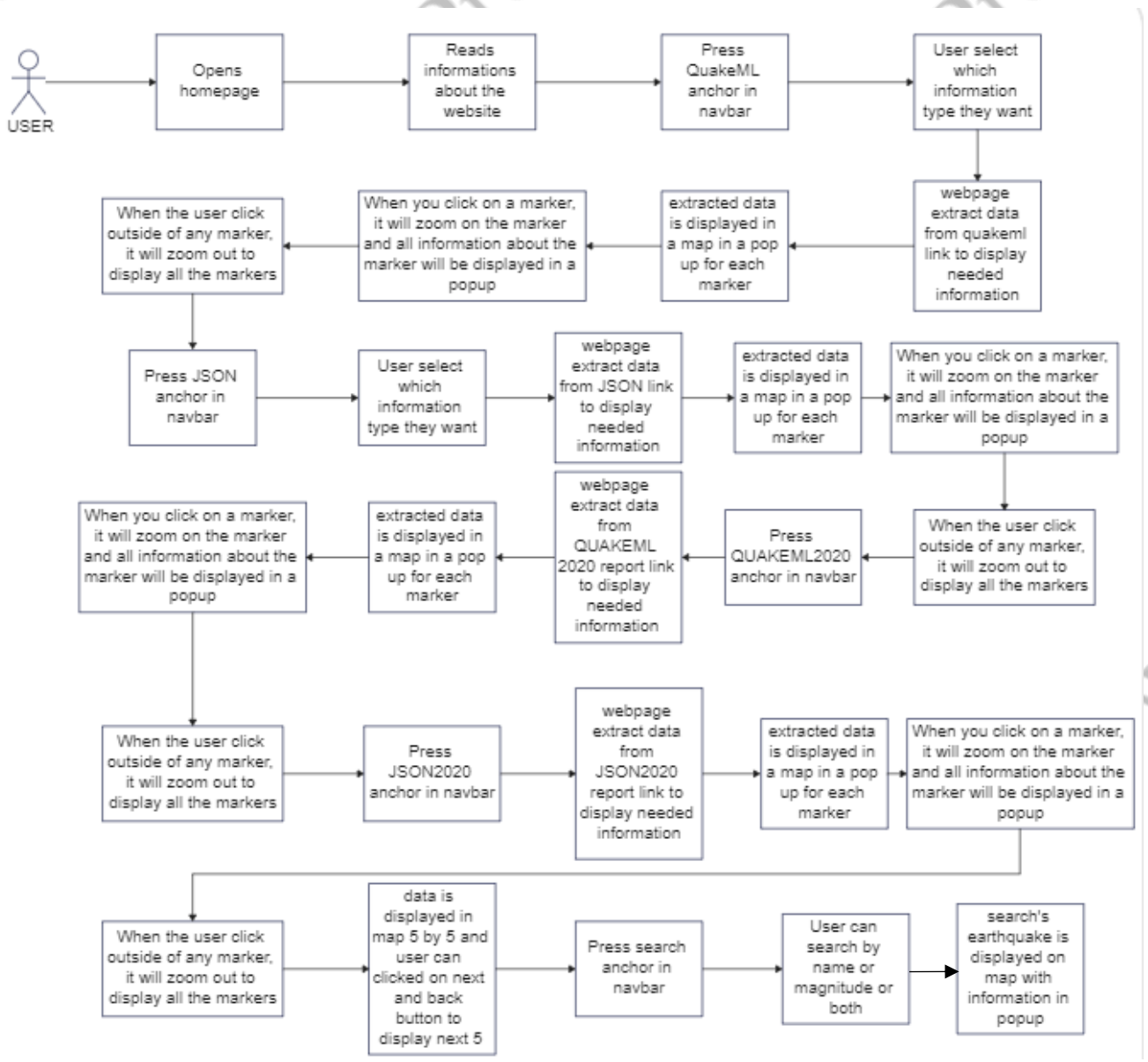
Contents

INTRODUCTION

Welcome to QuakeInfo, your ultimate source for real-time earthquake information. Our website utilizes QuakeML and GeoJSON data formats to provide you with the most accurate and up-to-date seismic data from around the world. With our intuitive interface and advanced search feature, you can easily filter earthquakes based on location, magnitude, depth, and time. Explore comprehensive earthquake details, including magnitude, depth, location, and time of occurrence. Visualize seismic activity through interactive maps and access educational resources on earthquake preparedness and ongoing research. Join us in promoting awareness and understanding of earthquakes. Start exploring Earthquake Tracker today and stay informed about seismic events as they happen.

Architecture diagram




USER
Opens homepage
Reads informations about the website
Press QuakeML anchor in navbar
User select which information type they want
webpage extract data from quakeml link to display needed information
extracted data is displayed in a map in a pop up for each marker
When you click on a marker, it will zoom on the marker and all information about the marker will be displayed in a popup
When the user click outside of any marker, it will zoom out to display all the markers
Press JSON anchor in navbar
User select which information type they want
webpage extract data from JSON link to display needed information
extracted data is displayed in a map in a pop up for each marker
When you click on a marker, it will zoom on the marker and all information about the marker will be displayed in a popup
When the user click outside of any marker, it will zoom out to display all the markers
Press QUAKEML2020 anchor in navbar
webpage extract data from QUAKEML 2020 report link to display needed information
extracted data is displayed in a map in a pop up for each marker
When you click on a marker, it will zoom on the marker and all information about the marker will be displayed in a popup
When the user click outside of any marker, it will zoom out to display all the markers
Press JSON2020 anchor in navbar
webpage extract data from JSON2020 report link to display needed information
extracted data is displayed in a map in a pop up for each marker
When you click on a marker, it will zoom on the marker and all information about the marker will be displayed in a popup
When the user click outside of any marker, it will zoom out to display all the markers
data is displayed in map 5 by 5 and user can clicked on next and back button to display next 5
Press search anchor in navbar
User can search by name or magnitude or both
search's earthquake is displayed on map with information in popup

File structure of website

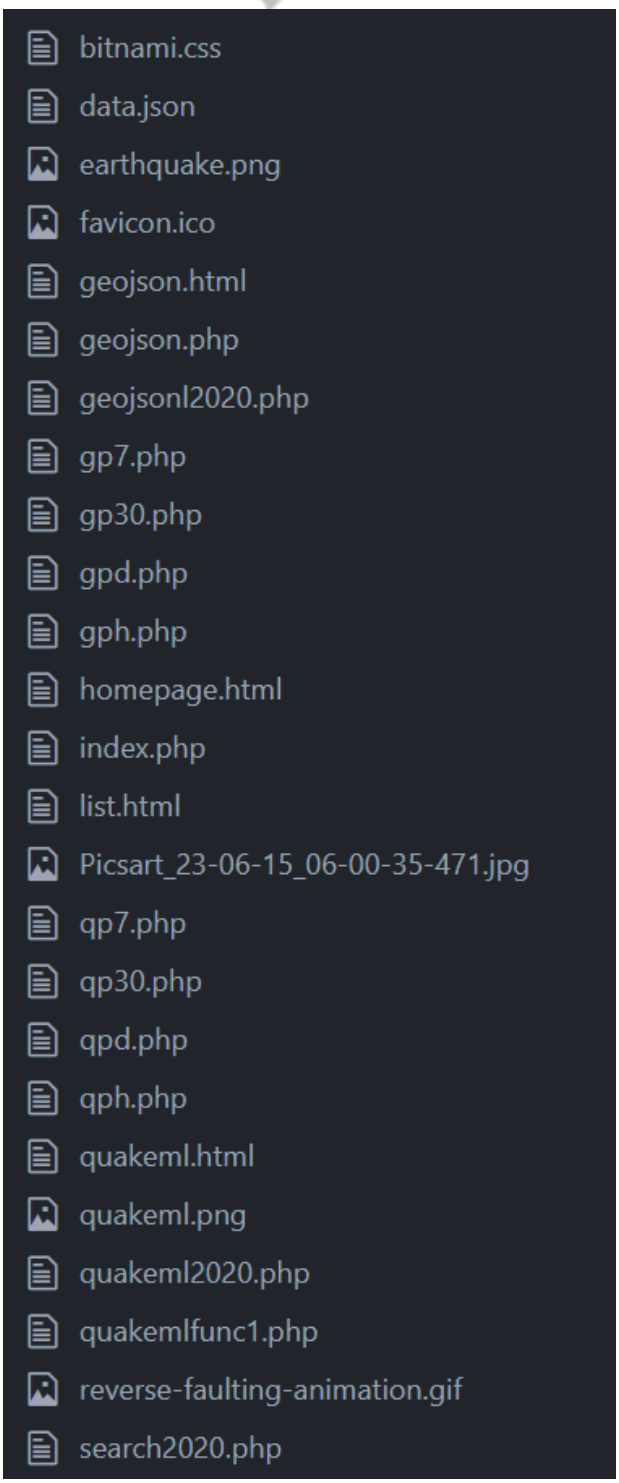


Figure 1

This is all the file needed for the website

Index interface

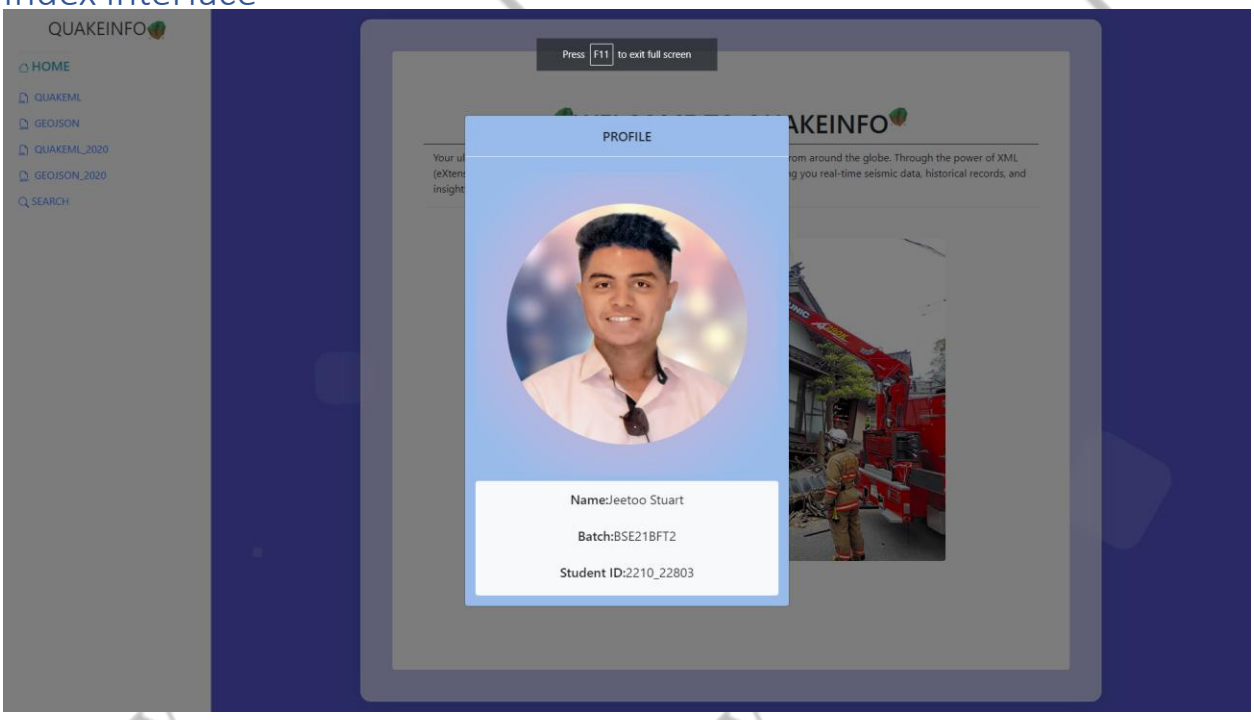


Figure 2

Pop up displaying my profile

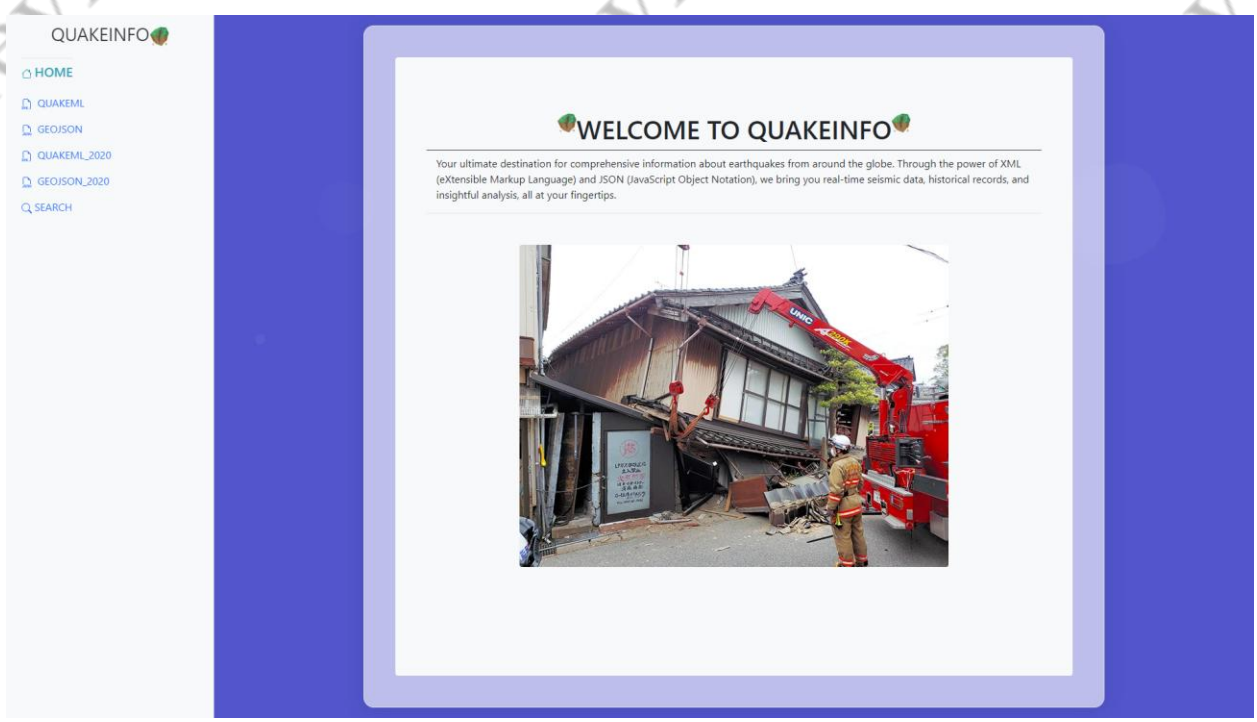


Figure 3

This is my homepage displaying the information of my website

Live QUAKEML information dashboard

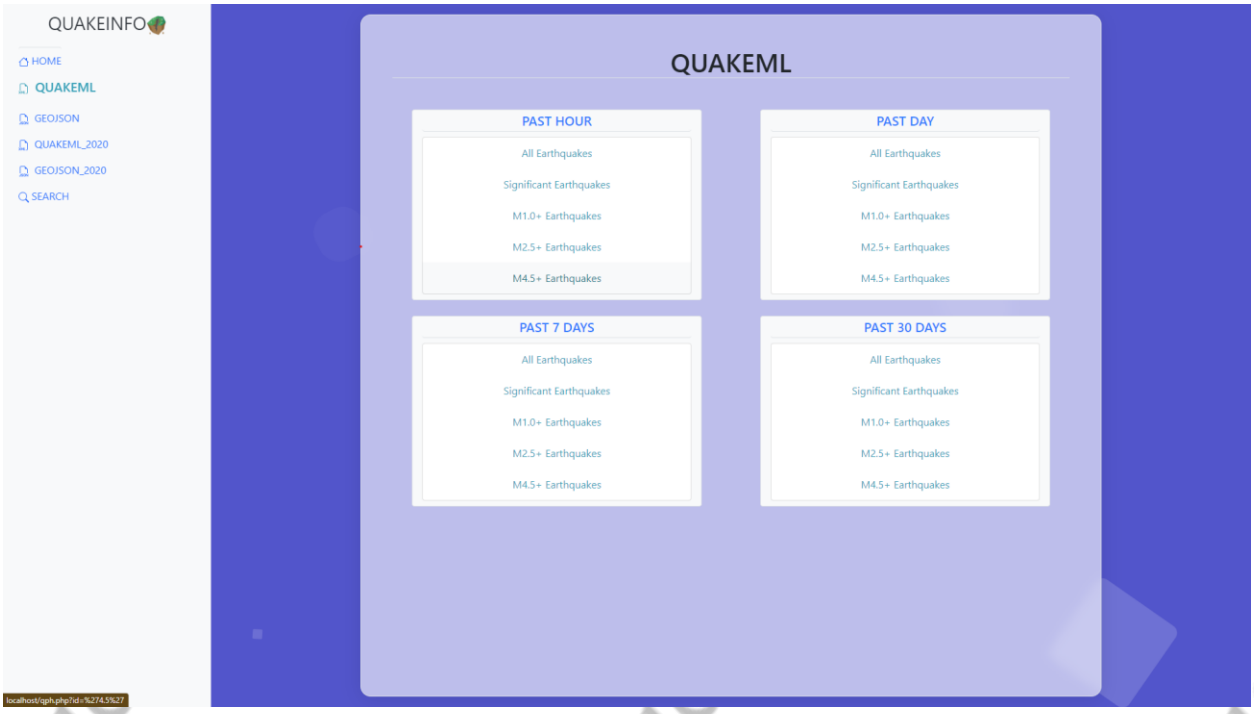


Figure 4

The user can choose what data they want to display either by version or by time

LIVE QUAKEML INFORMATION

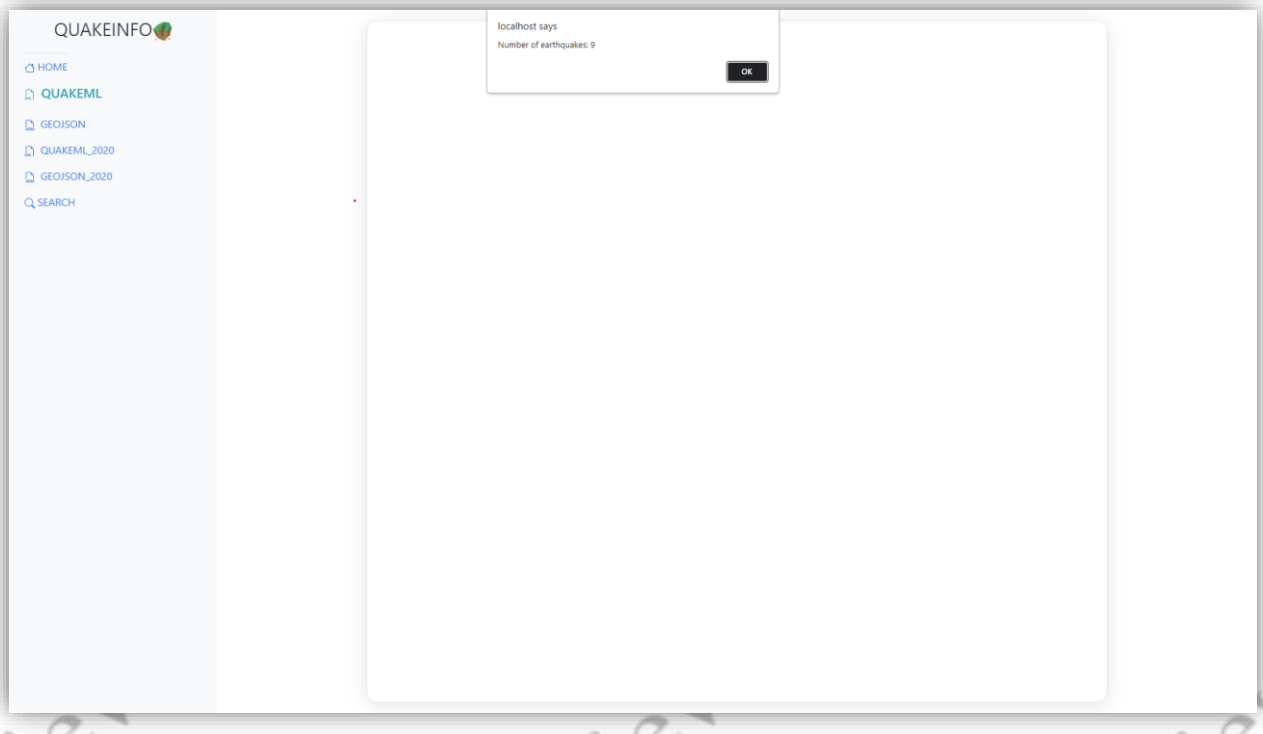


Figure 5 Alert displaying number of earthquake

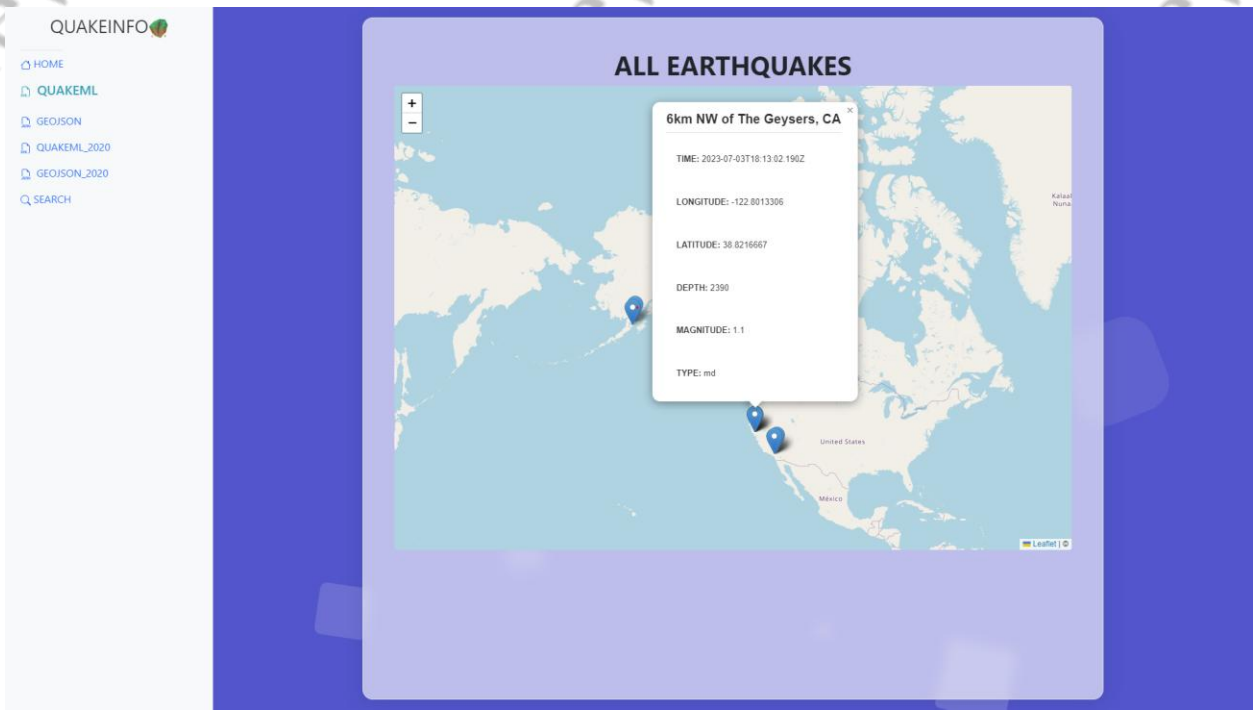


Figure 6

All earthquake displayed
according to version selected

PHP FOR FETCHING LIVE DATA FROM QUAKEML

```
<?php
      $a=array();
       $b=array();
        $c=array();
         $d=array();
          $e=array();
          $f=array();
          $g=array();
          $h=array();
          $i=array();
$name = strval($_GET['id']);
$link;
if ($name == "'all'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.quakeml";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>ALL EARTHQUAKES</b></div>");
}
if ($name == "'sig'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_hour.quakeml";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>SIGNIFICANT EARTHQUAKES</b></div>");
}
if ($name == "'1.0'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/1.0_hour.quakeml";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M1.0 EARTHQUAKES</b></div>");
}
if ($name == "'2.5'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.quakeml";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M2.5 EARTHQUAKES</b></div>");
}
if ($name == "'4.5'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_hour.quakeml";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M4.5 EARTHQUAKES</b></div>");
```

Array to store xml informations

The user can choose what data they want to display either by version or by time

ID from url will be used to display the version needed

```
}
        $xmlDoc=simplexml_load_file("$link");
        if (false === $xmlDoc) {
  echo ("test");
  libxml_clear_errors();
}else{

        foreach($xmlDoc->eventParameters->event as $val){
        $longitude = $val->origin->longitude->value;
        $latitude = $val->origin->latitude->value;
        $magnitude= $val->magnitude->mag->value;
        $time= $val->origin->time->value;
        $description= $val->description->text;
        $depth = $val->origin->depth->value;
        $type = $val->magnitude->type;
        $stationcount = $val->magnitude->stationCount;
        array_push($a,$longitude);
          array_push($b,$latitude);
            array_push($c,$magnitude);
              array_push($d,$time);
               array_push($e,$description);
               array_push($f,$depth);
                array_push($h,$type);
                array_push($i,$stationcount);

      }
}

        ?>
```

XML PARSER

Display test if error in QUAKEML link

Loop used to navigate between different data of QUAKEML

Each data is push in a specific array

HTML TO DISPLAY MAP FROM QUAKEML

```
<div class="h-75 rounded" id="map">
```

Container where map is loaded

JS TO LOAD MAP USING INFORMATION

```
const longi = <?php echo json_encode(array_values($a)); ?>;
const name =  <?php echo json_encode($e); ?>;
const latit = <?php echo json_encode($b); ?>;
const magni = <?php echo json_encode($c); ?>;
const time = <?php echo json_encode($d); ?>;
const depth = <?php echo json_encode($f); ?>;
const type = <?php echo json_encode($h); ?>;
const stacount = <?php echo json_encode($i); ?>;
var center1 = 0;
```

PHP array is converted to JAVASCRIPT array using json_encode

```
var center2 = 0;
var l =0;
var p=0;
function extractValue3(arr, prop) {
  let extractedValue = [];

  for (let i=0; i < arr.length ; ++i) {
    // extract value from property
    extractedValue.push(arr[i][prop]);
  }
  return extractedValue;
}
function extractValue(arr, prop) {
  let extractedValue = [];
  for (let i=0; i < arr.length ; ++i) {

    // extract value from property
    extractedValue.push(arr[i][prop]);
  }
  return extractedValue;
}
function extractValue2(arr, prop) {
  let extractedValue = [];
  for (let i=0; i < arr.length ; ++i) {
    // extract value from property
    extractedValue.push(arr[i][prop]);
  }
  return extractedValue;
}
const result = extractValue(longi, '0');
const result2 = extractValue2(latit, '0');
const result3 = extractValue2(name, '0');
const result4 = extractValue2(magni, '0');
const result5 = extractValue2(time, '0');
const result6 = extractValue2(depth, '0');
const result7 = extractValue2(type, '0');
const result8 = extractValue2(stacount, '0');
 for (let i=0; i < result.length ; ++i) {
center1 = center1 + Number(result[i]);
l = l + 1;
 }
 center1 = center1/l;
 for (let i=0; i < result2.length ; ++i) {
  center2 = center2 + Number(result2[i]);
```

Calculate average latitude and longitude for center display

Extract value from each key of array

```
   p = p + 1;
  }
  center2 = center2/p;
var map = L.map('map').setView([center2, center1], 3);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 19,
    attribution: '&copy; <a href="http://www.openstreetmap.org/copyright"></a>'
}).addTo(map);
var marker;
for(let i = 0; i < longi.length; i++){
  marker = L.marker([result2[i],result[i]]).addTo(map);
  marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result2[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p><p class=\"p-3\"><b>TYPE:</b> "+result7[i]+"</p>").openPopup().on('click', onClick);
  map.on('click', onClick2);
}
function onClick(e) {
    var curPos = this.getLatLng();
    var long = curPos.lng;
    var lati = curPos.lat;
    map.setView(new L.LatLng(lati, long), 6);
}
function onClick2(e) {
map.setView(new L.LatLng(center2, center1), 3);
}
let url_str = window.location.href;
let url = new URL(url_str);
let search_params = url.searchParams;
var id="";
id = search_params.get('id');
function more(test){
alert(test+test);
}
alert("Number of earthquakes: "+longi.length);
```

Display map

Display marker and popup using longitude and latitude

LIVE GEOJSON INFORMATION

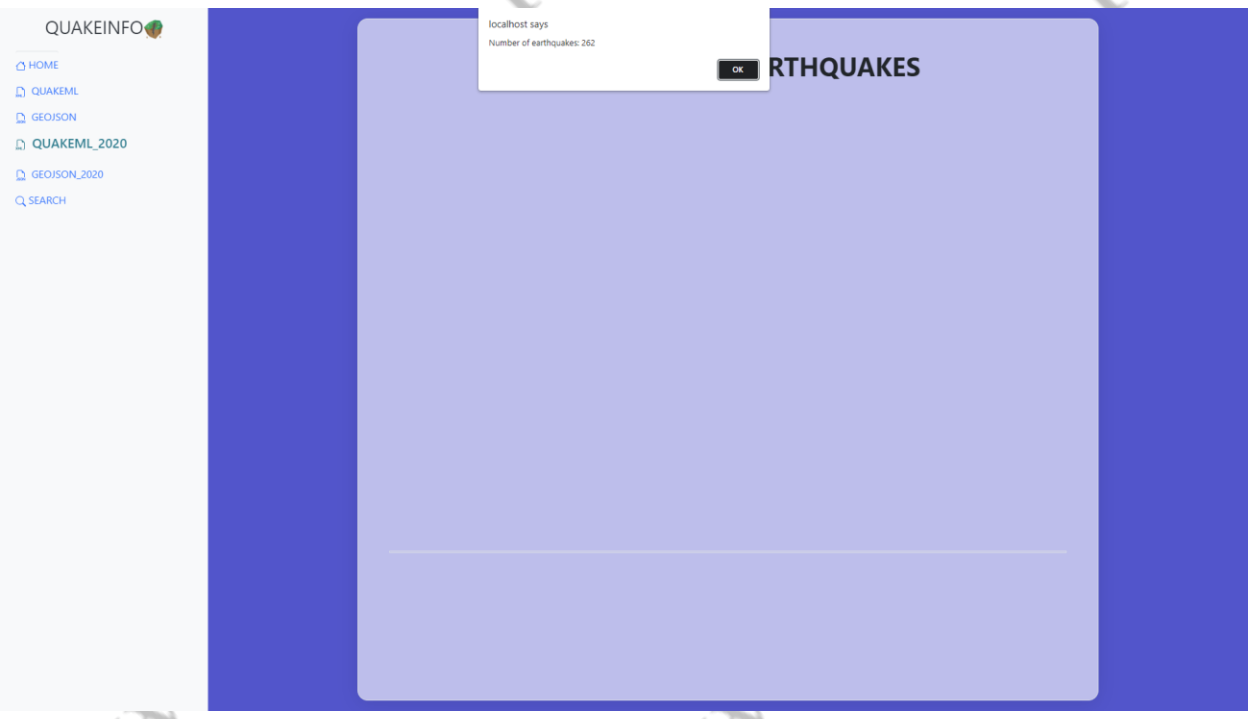


Figure 7 Alert displaying number of earthquake

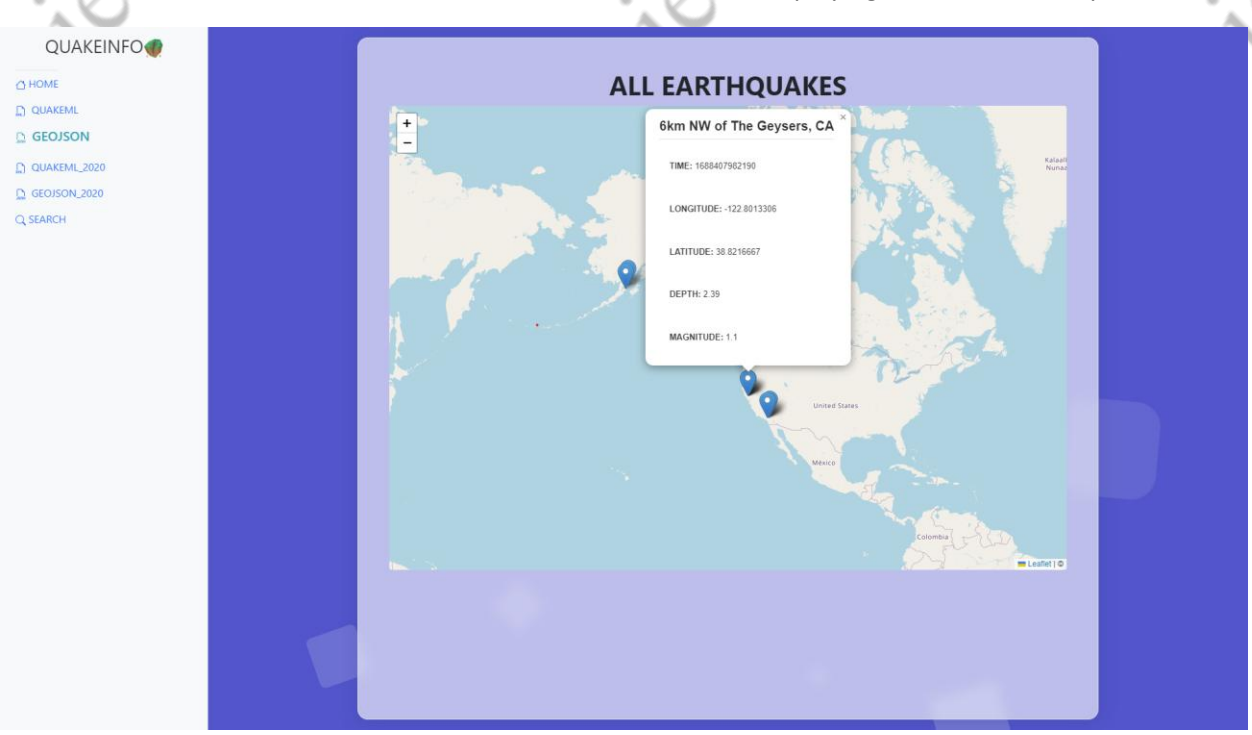


Figure 8

Display all data from GEOJSON
in map according to version

PHP FOR FETCHING LIVE DATA FROM GEOJSON

```
<?php
        $a=array();
         $b=array();
          $c=array();
           $d=array();
            $e=array();
            $f=array();
            $g=array();
            $h=array();
            $i=array();
$name = strval($_GET['id']);
$link;
if ($name == "'all'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>ALL EARTHQUAKES</b></div>");
}
if ($name == "'sig'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_hour.geojson";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>SIGNIFICANT EARTHQUAKES</b></div>");
}
if ($name == "'1.0'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/1.0_hour.geojson";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M1.0 EARTHQUAKES</b></div>");
}
if ($name == "'2.5'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_day.geojson";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M2.5 EARTHQUAKES</b></div>");
}
if ($name == "'4.5'")
{
 $link = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_hour.geojson";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>M4.5 EARTHQUAKES</b></div>");
}
$jsondata = file_get_contents($link);
        if (false === $jsondata) {
```

JSON Parser

```
 echo ("test");
 libxml_clear_errors();
}else{
$objdec = json_decode($jsondata);
        foreach($objdec->features as $value){
          $magnitude = $value->properties->mag;
 $description = $value->properties->place;
 $time = $value->properties->time;
 $longlat = $value->geometry->coordinates;
 $longitude= $longlat[0];
 $latitude = $longlat[1];
 $depth = $longlat[2];
        array_push($a,$longitude);
          array_push($b,$latitude);
          array_push($f,$depth);
            array_push($c,$magnitude);
              array_push($d,$time);
                array_push($e,$description);
      }
}
        ?>
```

Push specific data from JSON file where to specific arrays

HTML TO DISPLAY MAP FROM QUAKEML

```
<div class="h-75 rounded" id="map">
```

JS TO LOAD MAP USING INFORMATION

```
const longi = <?php echo json_encode(array_values($a)); ?>;
const name =  <?php echo json_encode($e); ?>;
const latit = <?php echo json_encode($b); ?>;
const magni = <?php echo json_encode($c); ?>;
const time = <?php echo json_encode($d); ?>;
const depth = <?php echo json_encode($f); ?>;
var center1 = 0;
var center2 = 0;
var l =0;
var p=0;
const result = longi;
const result2 = latit;
const result3 = name;
const result4 = magni;
const result5 = time;
const result6 = depth;
for (let i=0; i < result.length ; ++i) {
}
```

Convert PHP array to Javascript array

```
for (let i=0; i < result.length ; ++i) {
center1 = center1 + Number(result[i]);
l = l + 1;
 }
 center1 = center1/l;

 for (let i=0; i < result2.length ; ++i) {
   center2 = center2 + Number(result2[i]);
   p = p + 1;
 }
 center2 = center2/p;

var map = L.map('map').setView([center2, center1], 3);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
   maxZoom: 19,
   attribution: '&copy; <a href="http://www.openstreetmap.org/copyright"></a>'
}).addTo(map);
var marker;
for(let i = 0; i < longi.length; i++){
 marker = L.marker([result2[i],result[i]]).addTo(map);
 marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result2[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p>").openPopup().on('click', onClick);
 map.on('click', onClick2);
}
function onClick(e) {
   var curPos = this.getLatLng();
   var long = curPos.lng;
   var lati = curPos.lat;
   map.setView(new L.LatLng(lati, long), 6);

}
function onClick2(e) {
map.setView(new L.LatLng(center2, center1), 3);

}
let url_str = window.location.href;
let url = new URL(url_str);
let search_params = url.searchParams;
var id="";
id = search_params.get('id');
function more(test){
```

```
alert(test+test);
}
alert("Number of earthquakes: "+longi.length);
```

2020 QUAKEML EARTHQUAKES

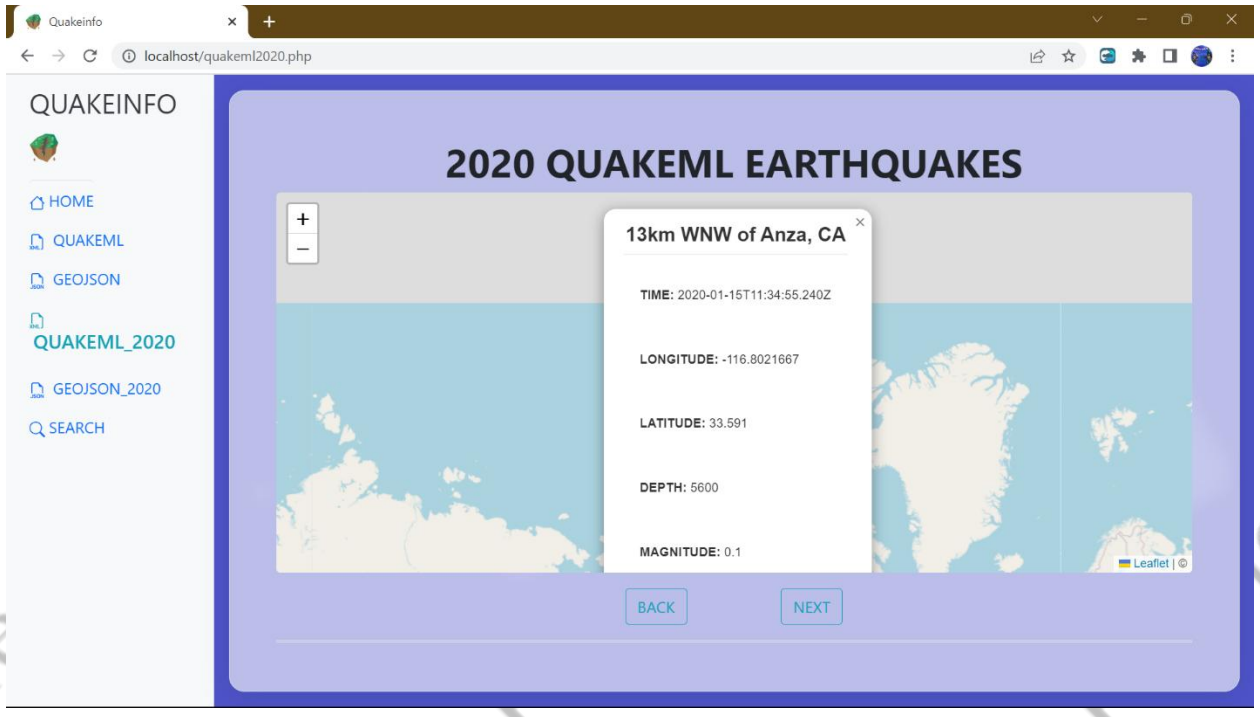


Figure 9

PHP FOR FETCHING LIVE DATA FROM QUAKEML

```
<?php
    $a=array();
     $b=array();
      $c=array();
       $d=array();
        $e=array();
        $f=array();
        $g=array();
        $h=array();
        $i=array();
$link = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=quakeml&starttime=2020-01-15T00:00:00&endtime=2020-01-15T12:00:00";
 echo("<div class=\"d-flex h1 justify-content-center\"><b>2020 QUAKEML EARTHQUAKES</b></div>");
    $xmlDoc=simplexml_load_file("$link");
    if (false === $xmlDoc) {
```

Use link for QUAKEML2020

```
 echo ("test");
 libxml_clear_errors();
}else{

        foreach($xmlDoc->eventParameters->event as $val){
        $longitude = $val->origin->longitude->value;
        $latitude = $val->origin->latitude->value;
        $magnitude= $val->magnitude->mag->value;
        $time= $val->origin->time->value;
        $description= $val->description->text;
        $depth = $val->origin->depth->value;
        $type = $val->magnitude->type;
        $stationcount = $val->magnitude->stationCount;
        array_push($a,$longitude);
          array_push($b,$latitude);
            array_push($c,$magnitude);
              array_push($d,$time);
                array_push($e,$description);
                array_push($f,$depth);
                array_push($h,$type);
                array_push($i,$stationcount);
    }
}
        ?>
```

HTML TO DISPLAY MAP FROM QUAKEML

```
    <div class="h-75 rounded" id="map">

 </div>

 <div class="container d-flex justify-content-center my-3">

   <button type="button" class="btn btn-outline-info mr-5" name="button" id="btn1"
onclick="back()">BACK</button>

    <button type="button" class="btn btn-outline-info ml-5" name="button" id="btn2"
onclick="next()">NEXT</button>

 </div>
```

Button for back and next 5 earthquakes

JS TO LOAD MAP USING INFORMATION

```
const longi = <?php echo json_encode(array_values($a)); ?>;
const name =  <?php echo json_encode($e); ?>;
const latit = <?php echo json_encode($b); ?>;
const magni = <?php echo json_encode($c); ?>;
const time = <?php echo json_encode($d); ?>;
const depth = <?php echo json_encode($f); ?>;
const type = <?php echo json_encode($h); ?>;
const stacount = <?php echo json_encode($i); ?>;
var center1 = 0;
var center2 = 0;
var l =0;
var p=0;
var initial = 0;
var max=0;
function extractValue3(arr, prop) {
   let extractedValue = [];
   for (let i=0; i < arr.length ; ++i) {
      // extract value from property
      extractedValue.push(arr[i][prop]);
   }
   return extractedValue;
}
function extractValue(arr, prop) {
   let extractedValue = [];
   for (let i=0; i < arr.length ; ++i) {

      // extract value from property
      extractedValue.push(arr[i][prop]);
   }
   return extractedValue;
}
function extractValue2(arr, prop) {
   let extractedValue = [];
   for (let i=0; i < arr.length ; ++i) {
      // extract value from property
      extractedValue.push(arr[i][prop]);
   }
   return extractedValue;
}
const result = extractValue(longi, '0');
const result2 = extractValue2(latit, '0');
```

```
 const result3 = extractValue2(name, '0');
 const result4 = extractValue2(magni, '0');
 const result5 = extractValue2(time, '0');
 const result6 = extractValue2(depth, '0');
 const result7 = extractValue2(type, '0');
 const result8 = extractValue2(stacount, '0');
  for (let i=0; i < result.length ; ++i) {
 center1 = center1 + Number(result[i]);
 l = l + 1;
  }
  center1 = center1/l;
  for (let i=0; i < result2.length ; ++i) {
    center2 = center2 + Number(result2[i]);
    p = p + 1;
  }
  center2 = center2/p;
var map = L.map('map').setView([center2, center1], 3);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
   maxZoom: 19,
   attribution: '&copy; <a href="http://www.openstreetmap.org/copyright"></a>'
}).addTo(map);
var marker;
max = max+5;
$(".leaflet-marker-icon").remove();
$(".leaflet-popup").remove();
$(".leaflet-zoom-animated").remove();
$(".leaflet-interactive").remove();
while (max>initial){
  marker = L.marker([result2[initial],result[initial]]).addTo(map);
  marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[initial]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[initial]+"</p><p
class=\"p-3\"><b>LONGITUDE:</b> "+result[initial]+"</p><p class=\"p-3\"><b>LATITUDE:</b>
"+result2[initial]+"</p><p class=\"p-3\"><b>DEPTH:</b> "+result6[initial]+"</p><p class=\"p-
3\"><b>MAGNITUDE:</b> "+result4[initial]+"</p><p class=\"p-3\"><b>TYPE:</b>
"+result7[initial]+"</p>").openPopup().on('click', onClick);
  map.on('click', onClick2);
  initial = initial+1;
}
if (initial>longi.length-5){
  $("#btn2").hide();
}else {
  $("#btn2").show();
}
if (initial==5) {
```

Hide or show button if more than 5 earthquakes is remained to be displayed

```
  $("#btn1").hide();
 }else {
    $("#btn1").show();
 }
 function next(){
  $(".leaflet-marker-icon").remove();
  $(".leaflet-popup").remove();
  $(".leaflet-zoom-animated").remove();
  $(".leaflet-interactive").remove();
  max = max+5;
  while (max!=initial){
    marker = L.marker([result2[initial],result[initial]]).addTo(map);
    marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[initial]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[initial]+"</p><p
class=\"p-3\"><b>LONGITUDE:</b> "+result[initial]+"</p><p class=\"p-3\"><b>LATITUDE:</b>
"+result2[initial]+"</p><p class=\"p-3\"><b>DEPTH:</b> "+result6[initial]+"</p><p class=\"p-
3\"><b>MAGNITUDE:</b> "+result4[initial]+"</p><p class=\"p-3\"><b>TYPE:</b>
"+result7[initial]+"</p>").openPopup().on('click', onClick);
    map.on('click', onClick2);
    initial = initial+1;
  }
  if (initial>longi.length-5){
    $("#btn2").hide();
  }else {
    $("#btn2").show();
  }
  if (initial==5) {
    $("#btn1").hide();
  }else {
     $("#btn1").show();
  }
 }
 function back(){
  $(".leaflet-marker-icon").remove();
  $(".leaflet-popup").remove();
  $(".leaflet-zoom-animated").remove();
  $(".leaflet-interactive").remove();
  max = max-5;
  while (max!=initial){
    marker = L.marker([result2[initial],result[initial]]).addTo(map);
    marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[initial]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[initial]+"</p><p
class=\"p-3\"><b>LONGITUDE:</b> "+result[initial]+"</p><p class=\"p-3\"><b>LATITUDE:</b>
"+result2[initial]+"</p><p class=\"p-3\"><b>DEPTH:</b> "+result6[initial]+"</p><p class=\"p-
```

Function used to display next 5 earthquakes triggered by next button

Hide or show button if less than 5 earthquakes is remained to be displayed

Function used to display last 5 earthquakes triggered by back button

```
3\"><b>MAGNITUDE:</b> "+result4[initial]+"</p><p class=\"p-3\"><b>TYPE:</b>
"+result7[initial]+"</p>").openPopup().on('click', onClick);
   map.on('click', onClick2);
   initial = initial-1;
 }
 if (initial>longi.length-5){
   $("#btn2").hide();
 }else {
   $("#btn2").show();
 }
 if (initial==5) {
   $("#btn1").hide();
 }else {
    $("#btn1").show();
 }
}
alert("Number of earthquakes: "+longi.length);
function onClick(e) {
   var curPos = this.getLatLng();
   var long = curPos.lng;
   var lati = curPos.lat;
   map.setView(new L.LatLng(lati, long), 6);
}
function onClick2(e) {
map.setView(new L.LatLng(center2, center1), 3);
}
let url_str = window.location.href;
let url = new URL(url_str);
let search_params = url.searchParams;
var id="";
id = search_params.get('id');
```

2020 GEOJSON EARTHQUAKES

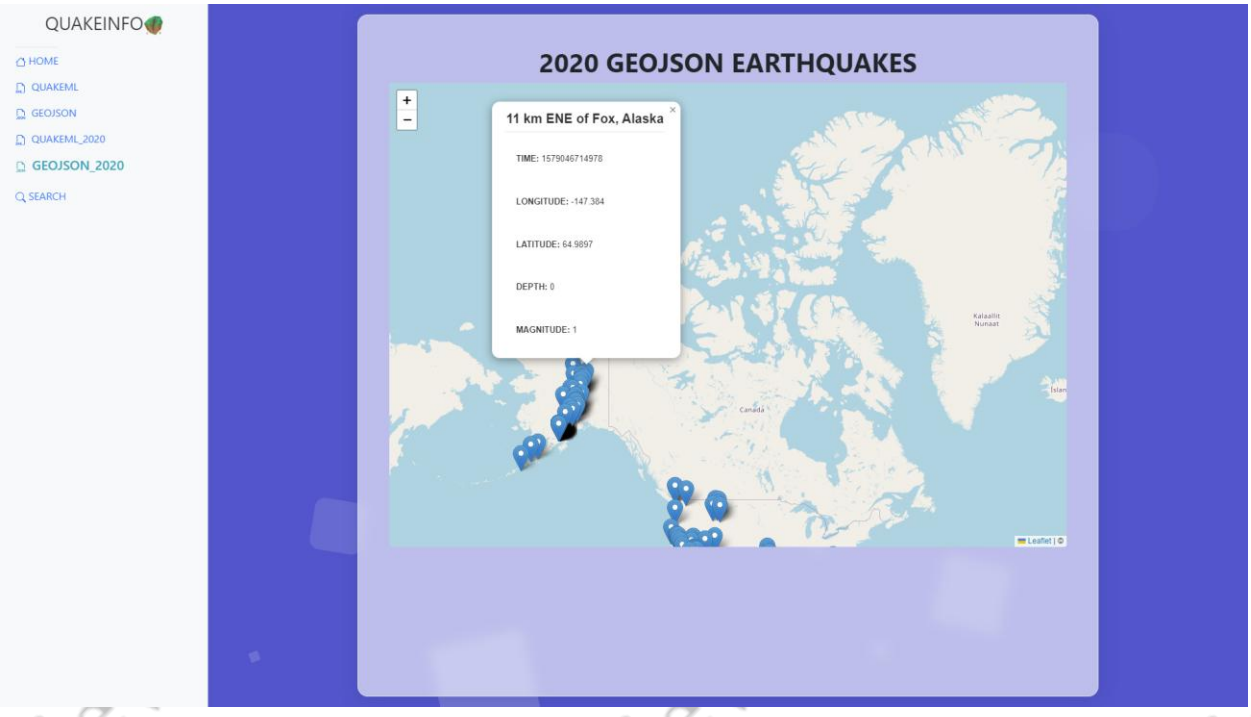


Figure 10

PHP FOR FETCHING LIVE DATA FROM GEOJSON 2020

```
 <?php
        $a=array();
         $b=array();
          $c=array();
           $d=array();
            $e=array();
            $f=array();
            $g=array();
            $h=array();
            $i=array();
             echo("<div class=\"d-flex h1 justify-content-center\"><b>2020 GEOJSON
EARTHQUAKES</b></div>");
$link = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2020-01-
15T00:00:00&endtime=2020-01-15T12:00:00";
$jsondata = file_get_contents($link);
        if (false === $jsondata) {
  echo ("test");
  libxml_clear_errors();
}else{
$objdec = json_decode($jsondata);
```

```
        foreach($objdec->features as $value){
          $magnitude = $value->properties->mag;
$description = $value->properties->place;
$time = $value->properties->time;
$longlat = $value->geometry->coordinates;
$longitude= $longlat[0];
$latitude = $longlat[1];
$depth = $longlat[2];
        array_push($a,$longitude);
          array_push($b,$latitude);
          array_push($f,$depth);
            array_push($c,$magnitude);
              array_push($d,$time);
                array_push($e,$description);
      }
}
        ?>
```

HTML TO DISPLAY MAP FROM QUAKEML

```
<div class="h-75 rounded" id="map">
```

JS TO LOAD MAP USING INFORMATION

```
const longi = <?php echo json_encode(array_values($a)); ?>;
const name =  <?php echo json_encode($e); ?>;
const latit = <?php echo json_encode($b); ?>;
const magni = <?php echo json_encode($c); ?>;
const time = <?php echo json_encode($d); ?>;
const depth = <?php echo json_encode($f); ?>;
var center1 = 0;
var center2 = 0;
var l =0;
var p=0;
const result = longi;
const result2 = latit;
const result3 = name;
const result4 = magni;
const result5 = time;
const result6 = depth;
for (let i=0; i < result.length ; ++i) {
}
for (let i=0; i < result.length ; ++i) {
center1 = center1 + Number(result[i]);
```

```
l = l + 1;
 }
 center1 = center1/l;

 for (let i=0; i < result2.length ; ++i) {
  center2 = center2 + Number(result2[i]);
  p = p + 1;
 }
 center2 = center2/p;

var map = L.map('map').setView([center2, center1], 3);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
  attribution: '&copy; <a href="http://www.openstreetmap.org/copyright"></a>'
}).addTo(map);
var marker;
for(let i = 0; i < longi.length; i++){
 marker = L.marker([result2[i],result[i]]).addTo(map);
 marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result2[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p>").openPopup().on('click', onClick);
 map.on('click', onClick2);
}
var popup = L.popup()
   .setLatLng(latlng)
   .setContent('<p>Hello world!<br />This is a nice popup.</p>')
   .openOn(map);
function onClick(e) {
  var curPos = this.getLatLng();
  var long = curPos.lng;
  var lati = curPos.lat;
  map.setView(new L.LatLng(lati, long), 6);

}
function onClick2(e) {
map.setView(new L.LatLng(center2, center1), 3);
}
let url_str = window.location.href;
let url = new URL(url_str);
let search_params = url.searchParams;
var id="";
id = search_params.get('id');
```

```
function more(test){
alert(test+test);
}
alert("Number of earthquakes: "+longi.length);
```

SEARCHBAR FUNCTIONALITY

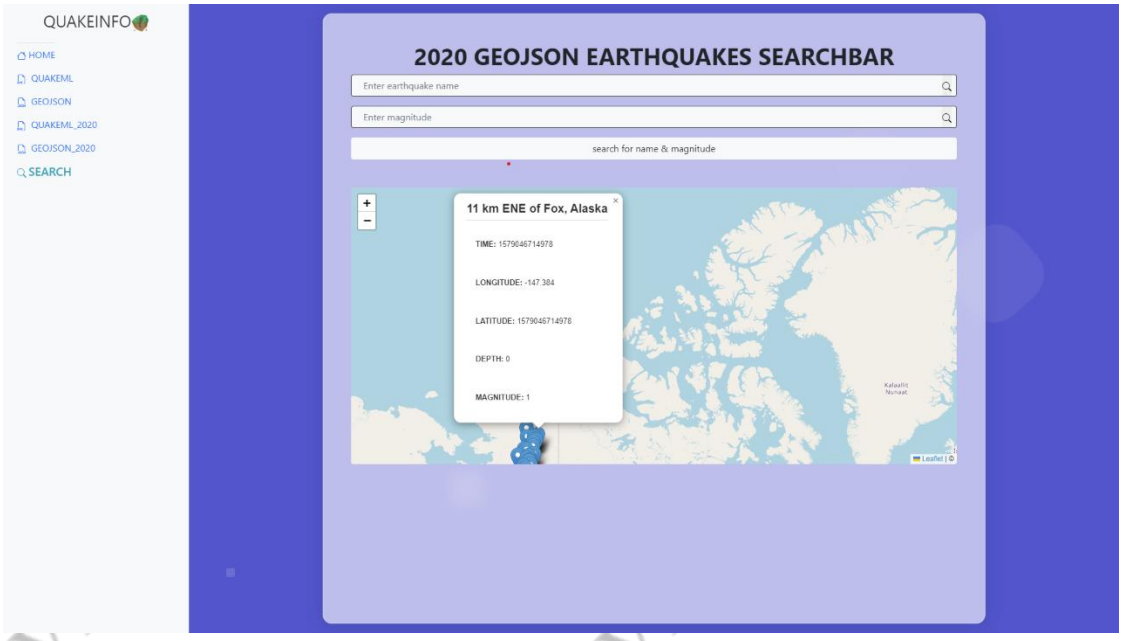


Figure 11

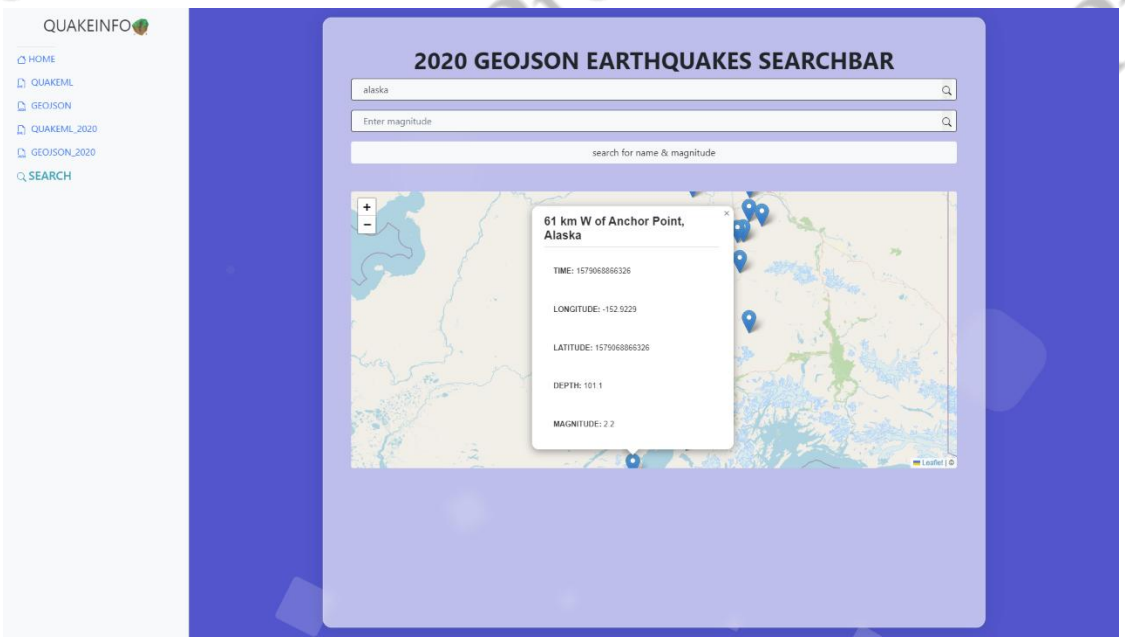


Figure 12

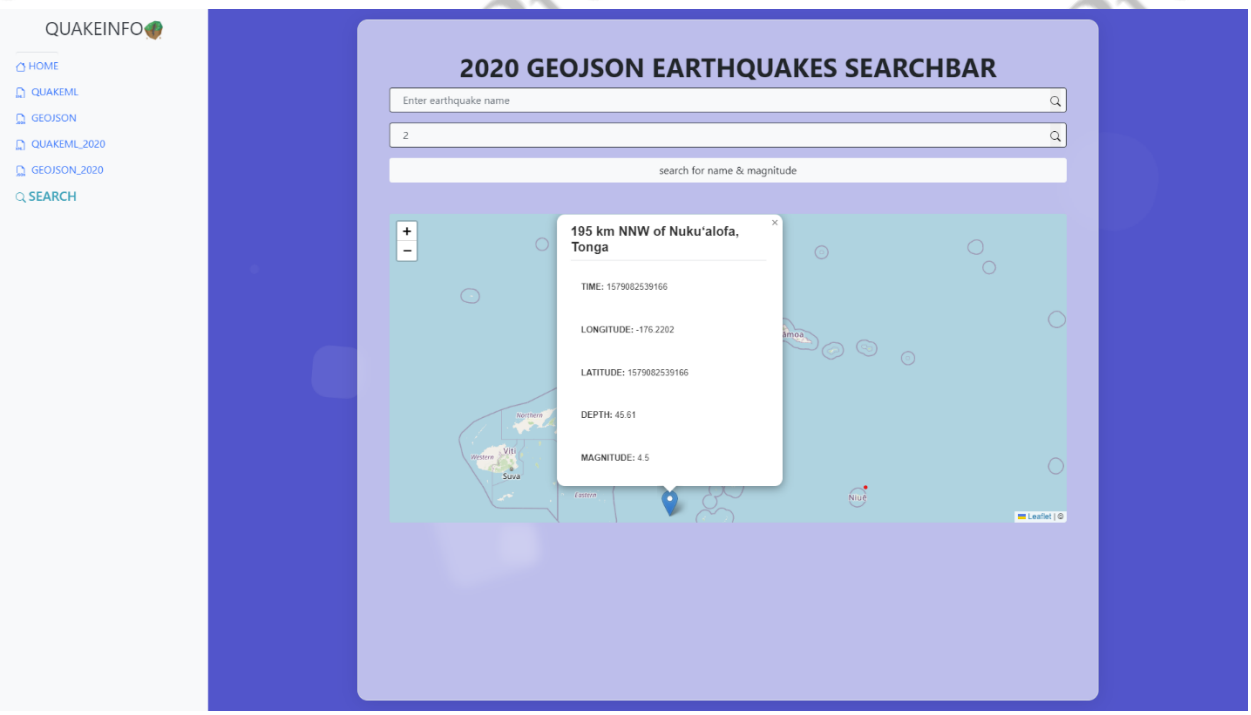


Figure 13

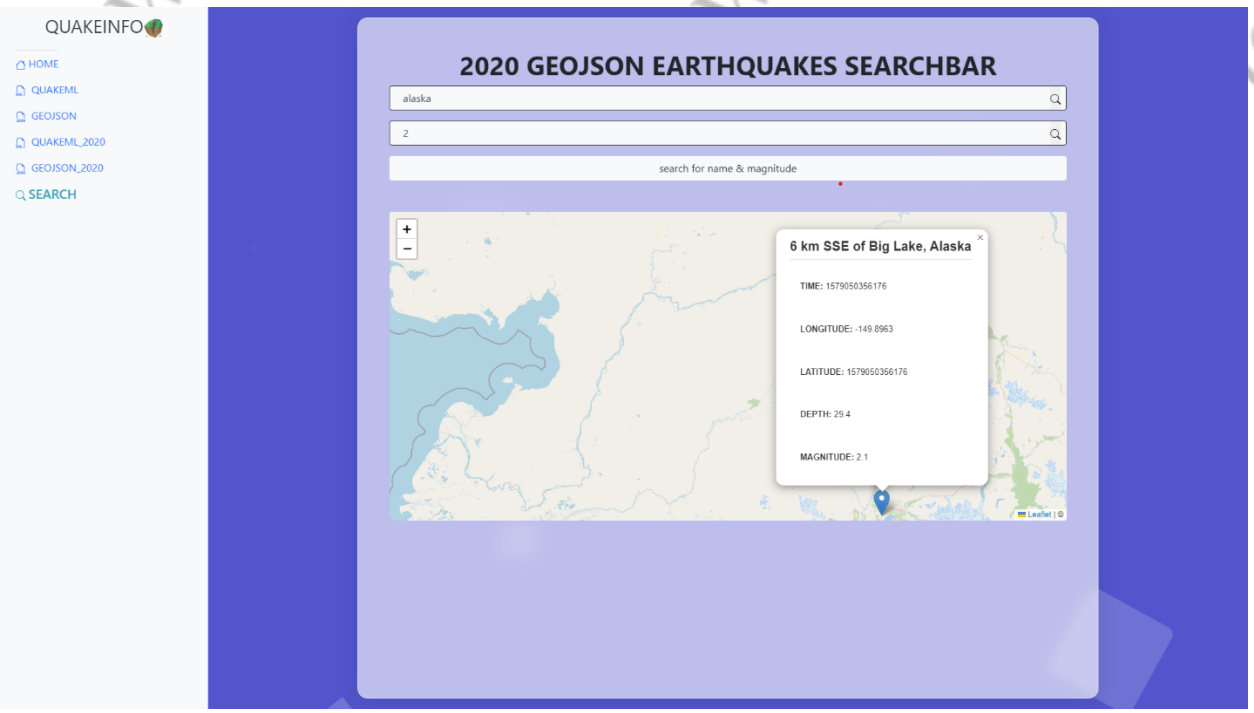


Figure 14

PHP FOR SEARCH DATA FROM GEOJSON2020

```
 <?php
        $a=array();
         $b=array();
          $c=array();
           $d=array();
            $e=array();
            $f=array();
            $g=array();
            $h=array();
            $i=array();
             echo("<div class=\"d-flex h1 justify-content-center\"><b>2020 GEOJSON EARTHQUAKES
SEARCHBAR</b></div>");
$link = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2020-01-
15T00:00:00&endtime=2020-01-15T12:00:00";
$jsondata = file_get_contents($link);
        if (false === $jsondata) {
  echo ("test");
  libxml_clear_errors();
}else{
$objdec = json_decode($jsondata);
        foreach($objdec->features as $value){
          $magnitude = $value->properties->mag;
 $description = $value->properties->place;
 $time = $value->properties->time;
 $longlat = $value->geometry->coordinates;
 $longitude= $longlat[0];
 $latitude = $longlat[1];
 $depth = $longlat[2];
        array_push($a,$longitude);
          array_push($b,$latitude);
          array_push($f,$depth);
           array_push($c,$magnitude);
            array_push($d,$time);
             array_push($e,$description);
      }
}
        ?>
```

JSON parser

HTML TO DISPLAY MAP FROM GEOJSON 2020

```
<div class="h-75 rounded" id="map">
<button type="button" name="button" class="container btn btn-light d-flex justify-content-center mt-3"
onclick="both()">search for name & magnitude</button>
```

JS TO LOAD MAP USING INFORMATION

```
const longi = <?php echo json_encode(array_values($a)); ?>;
const name =  <?php echo json_encode($e); ?>;
const latit = <?php echo json_encode($b); ?>;
const magni = <?php echo json_encode($c); ?>;
const time = <?php echo json_encode($d); ?>;
const depth = <?php echo json_encode($f); ?>;
var center1 = 0;
var center2 = 0;
var l =0;
var p=0;
const result = longi;
const result2 = latit;
const result3 = name;
const result4 = magni;
const result5 = time;
const result6 = depth;
for (let i=0; i < result.length ; ++i) {
}
for (let i=0; i < result.length ; ++i) {
center1 = center1 + Number(result[i]);
l = l + 1;
 }
 center1 = center1/l;

 for (let i=0; i < result2.length ; ++i) {
   center2 = center2 + Number(result2[i]);
   p = p + 1;
 }
 center2 = center2/p;
var map = L.map('map').setView([center2, center1], 3);
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
   maxZoom: 19,
   attribution: '&copy; <a href="http://www.openstreetmap.org/copyright"></a>'
}).addTo(map);
var marker;
function search(){
  $(".leaflet-marker-icon").remove();
  $(".leaflet-popup").remove();
  $(".leaflet-zoom-animated").remove();
  $(".leaflet-interactive").remove();
let text = $("#searchtxt").val();
for(let i = 0; i < longi.length; i++){
  if (  result3[i].toLowerCase().includes(text.toLowerCase())) {
```

Remove old markers and add new markers

Check if name is same as value form search box

```
 marker = L.marker([result2[i],result[i]]).addTo(map);
 marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result5[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p>").openPopup().on('click', onClick);
 map.on('click', onClick2);
}
}
}
function search2(){
  $(".leaflet-marker-icon").remove();
  $(".leaflet-popup").remove();
  $(".leaflet-zoom-animated").remove();
  $(".leaflet-interactive").remove();
let text = $("#searchtxt2").val();
for(let i = 0; i < longi.length; i++){
  if (result4[i]>text) {
 marker = L.marker([result2[i],result[i]]).addTo(map);
 marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result5[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p>").openPopup().on('click', onClick);
 map.on('click', onClick2);
}
}
}
function onClick(e) {
   var curPos = this.getLatLng();
   var long = curPos.lng;
   var lati = curPos.lat;
   map.setView(new L.LatLng(lati, long), 6);

}
function onClick2(e) {
map.setView(new L.LatLng(center2, center1), 3);
}
let url_str = window.location.href;
let url = new URL(url_str);
let search_params = url.searchParams;
var id="";
id = search_params.get('id');
function more(test){
```

Remove old markers and add new markers

Check if magnitude is more than magnitude from searchbox

Zoom on markers when clicked on marker

Zoom out of map when click outside of map

```
alert(test+test);
}
function both(){
  $(".leaflet-marker-icon").remove();
  $(".leaflet-popup").remove();
  $(".leaflet-zoom-animated").remove();
  $(".leaflet-interactive").remove();
let text = $("#searchtxt").val();
let text2 = $("#searchtxt2").val();
for(let i = 0; i < longi.length; i++){
  if (  result3[i].toLowerCase().includes(text.toLowerCase())&&result4[i]>text2) {
 marker = L.marker([result2[i],result[i]]).addTo(map);
 marker.bindPopup("<h1 class=\"h5 pb-2 border-bottom d-flex justify-content-
center\"><b>"+result3[i]+"</b></h1><p class=\"p-3\"><b>TIME:</b> "+result5[i]+"</p><p class=\"p-
3\"><b>LONGITUDE:</b> "+result[i]+"</p><p class=\"p-3\"><b>LATITUDE:</b> "+result5[i]+"</p><p
class=\"p-3\"><b>DEPTH:</b> "+result6[i]+"</p><p class=\"p-3\"><b>MAGNITUDE:</b>
"+result4[i]+"</p>").openPopup().on('click', onClick);
 map.on('click', onClick2);
}
}
}
```

Remove old markers and add new markers

Check if magnitude is more than magnitude from searchbox and If name matches name searched in textbox

---

In conclusion, QuakeInfo is your trusted companion in understanding and monitoring seismic activity. Our website harnesses the power of QuakeML and GeoJSON data formats to provide you with real-time earthquake information from around the globe. With our intuitive search feature, you can easily explore and filter earthquakes based on various criteria, enabling you to uncover valuable insights and stay informed. We strive to empower individuals and communities with knowledge about earthquakes through educational resources and promote preparedness for seismic events. Join us on Earthquake Tracker and be a part of our mission to enhance awareness and understanding of earthquakes. Together, let's navigate the fascinating world of seismic activity and stay one step ahead.