

Šolski center Novo mesto

Srednja elektro šola in tehniška gimnazija

Šegova ulica 112

8000 Novo mesto

## **Spletna aplikacija za spremljanje potresov**

(Izdelek in zagovor)

Avtor: Adnan Smlatić, R4A

Mentor: Jernej Kastelic, dipl. inž. rač. splet. teh.

Šentrupert, marec 2021

## **Povzetek in ključne besede**

Samo v letu 2020 so opazovalne državne mreže v Evropi zabeležile več kot 101348 potresov z magnitudo pod 2 in 769 potresov z magnitudo med 4 in 5. Potrese zabeležujejo seizmografske postaje, ki delujejo po vsem svetu in samodejno pošiljajo svoje podatke na osrednje lokacije za rutinsko analizo, zato je na spletu hitro na voljo veliko podatkov o potresnih dejavnostih. Na spletu pa ni na voljo vzdrževane in moderne spletne strani, kjer bi si lahko ogledali te podatke na enostaven in uporabniku prijazen način. V okviru projektne naloge sem razvil moderno spletno aplikacijo, ki bo to rešila, uporabnikom bo omogočala vpogled v potrese na interaktivnem zemljevidu, iskanje in prikazovanje podrobnosti potresov ter dodajanje izjav in slik.

**Ključne besede:** spletna aplikacija, seizmologija, API, PHP, Leaflet, XML, MySQL, FDSNWS

# Kazalo

1	Uvod .....	1
1.1	Cilj .....	1
1.2	Struktura naloge .....	1
2	Pregled sorodnih aplikacij .....	2
2.1	ARSO potresi .....	2
2.2	EMSC .....	2
2.3	ZAMG .....	2
3	Uporabljene tehnologije .....	3
4	Teorija .....	5
4.1	Struktura .....	5
4.2	Podatki .....	6
4.3	Podatkovna baza .....	7
5	Razvoj .....	9
5.1	Zemljevid .....	9
5.2	Podatki .....	10
5.3	Izpisovanje in kartiranje .....	11
5.4	Bootstrap .....	18
5.5	Izjave .....	21
6	Predstavitev aplikacije .....	25
6.1	Glavna stran .....	25
6.2	Podrobnosti potresa .....	29
6.3	Mobilna odzivnost .....	34
7	Zaključek .....	36
7.1	Nadaljnje delo .....	36
7.2	Zahvala .....	36
8	Literatura .....	37

## Kazalo slik

Slika 1: Struktura aplikacije (Lastni vir).....	5
Slika 2: Zgradba REST poizvedbe (Lastni vir).....	6
Slika 3: Relacijski model podatkovne baze .....	8
Slika 4: Prva verzija aplikacije .....	9
Slika 5: Druga verzija aplikacije.....	13
Slika 6: Druga verzija podrobnosti.....	15
Slika 7: Tretja verzija aplikacije .....	18
Slika 8: Tretja verzija - podrobnosti .....	18
Slika 9: Obrazec za vpis podatkov .....	22
Slika 10: Avtomatski vpis lokacije.....	23
Slika 11: Glavna stran aplikacije.....	25
Slika 12: Svetel zemljevid.....	26
Slika 13: Satelitski zemljevid .....	26
Slika 14: Hitri filter .....	27
Slika 15: Rezultati hitrega filtra .....	27
Slika 16: Napredni filter .....	28
Slika 17: Okno stranske vrstice .....	28
Slika 18: Podrobnosti potresa.....	29
Slika 19: Sloji potresa .....	29
Slika 20: Obrazec za izjavo .....	30
Slika 21: Vnašanje lokacije.....	31
Slika 22: Obrazec za vnos kontaktnih informacij .....	31
Slika 23: Izjave .....	32
Slika 24: Tabela za izvore dogodkov .....	32
Slika 25: Tabela za momentne tenzorje .....	32
Slika 26: Zgodovina območja .....	33
Slika 27: Seizmična aktivnost v območju.....	33
Slika 28: Odzivnost glavne strani in stranske vrstice.....	34
Slika 29: Odzivnost podrobnosti potresa in obrazca za izjave.....	35

## Kazalo skript

Skripta 1: Struktura deleža XML dokumenta .....	7
Skripta 2: Definiranje zemljevida .....	9
Skripta 3: Funkcija za pridobivanje podatkov .....	10
Skripta 4: Posredovanje JS in izpisovanje stranske vrstice .....	11
Skripta 5: Struktura polja .....	11
Skripta 6: Funkcija za kartiranje potresov .....	13
Skripta 7: Funkcija za momentne tenzorje .....	14
Skripta 8: Struktura zapisa momentnih tenzorjev .....	14
Skripta 9: Funkcija za izvor dogodka .....	16
Skripta 10: Funkcija za pridobivanje odzivov.....	17
Skripta 11: Kartiranje odzivov.....	17

Skripta 12: Funkcija generateQuery .....	19
Skripta 13: Funkcija za hitri filter.....	20
Skripta 14: Ustvarjanje podatkovne baze .....	21
Skripta 15: Funkcija za povezovanje do podatkovne baze .....	21
Skripta 16: Vpisovanje podatkov v podatkovno bazo .....	22
Skripta 17: Funkcija za iskanje lokacije .....	23
Skripta 18: Funkcija za pošiljanje slik na Imgur API .....	24
Skripta 19: Izjave brez rezultatov .....	30

## Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application Program Interface	Programski vmesnik
CDN	Content delivery network	Omrežje za dostavo vsebine
CSS	Cascading Style Sheets	Kaskadne stilske podloge
DOM	Document Object Model	Objektni model dokumenta
HTML	Hypertext Markup Language	Jezik za označevanje nadbesedila
JS	JavaScript	JavaScript
PHP	PHP: Hypertext Preprocessor	PHP: predprocesor nadbesedila
SQL	Structured Query Language	Strukturirani povpraševali jezik za delo s podatkovnimi bazami
URL	Uniform Resource Locator	Enotni naslov vira
XML	Extensible Markup Language	Razširljivi označevalni jezik

Besede, ki se nanašajo na imena spremenljivk in atributov v kodi, so napisane s Courier New pisavo za boljšo preglednost in berljivost. Primer: `getEarthquakes`.

# 1 Uvod

Potres je opredeljen kot pojav, ko tektonske plošče zdrsnejo druga mimo druge in ustvarijo potresne valove, ki potujejo skozi zemeljske kamnine. Učinki so lahko odvisni od intenzivnosti potresov, od manjših strukturnih poškodb stavb do popolnega propada, kar ima za posledico izgubo življenj in premoženja. Intenziteto potresov merimo s pomočjo evropske makro seizmične lestvice. Merske enote zapisujemo z rimskim številkami od I (neobčutljiv) do XII (popolno uničenje) in kratico EMS.

Potresi so nepredvidljivi in jih ni mogoče napovedati s trenutno tehnologijo. Lahko pa izračunamo verjetnost in tveganje potresa na geološko aktivnih območjih z uporabo zgodovinskega zapisa seizmoloških aktivnosti. Po oceni Ministerstva za obrambo je Slovenija država s srednjo potresno nevarnostjo predvsem zaradi plitvih žarišč. (1)

Spletna aplikacija je aplikacijski program, ki je shranjen na oddaljenem strežniku in prek brskalniškega vmesnika dostavljen prek interneta. Dinamične aplikacije so tiste, ki zahtevajo obdelavo podatkov na strani strežnika. Obstoječe spletne aplikacije za spremljanje aktivnosti potresov, so večinoma zastarele, neodzivne, uporabnikom neprijazne in omejene.

## 1.1 Cilj

Cilj naloge je uporabiti podatke o potresih in seizmološki aktivnosti v Evropi in svetu za razvoj moderne spletne aplikacije, ki bo podatke obdelala in prikazala na enostaven ter uporabniku prijazen način. To vključuje kartiranje potresov na interaktivnemu zemljevidu, vpogled v tehnične podrobnosti potresa, filtriranje potresov po poljubnih parametrih, vpisovanje in prikazovanje izjav in slik uporabnikov, prikazovanje podrobnosti momentnih tenzorjev, žarišča in lokacije, kjer so uporabniki čutili potres ter prikazovanje seizmične zgodovine za to območje.

## 1.2 Struktura naloge

V drugem poglavju so na kratko predstavljene sorodne aplikacije. V nadaljevanju so opisane tehnologije uporabljene pri izdelavi aplikacije in struktura ter delovanje aplikacije s teoretične strani. Sledi predstavitev razvoja aplikacije, vse od inicializacije zemljevida do pridobivanja in obdelave podatkov ter implementacije filtrov, izjav in zgodovine. V zaključku so podane sklepne misli in ugotovitve ter predlogi za nadaljnje delo in izboljšanje spletne aplikacije.

## 2 Pregled sorodnih aplikacij

V tem poglavju so opisane aplikacije, katerih analiza mi je služila kot pomoč pri razvoju spletne aplikacije. Funkcionalnost teh aplikacij povsem zadošča za pregled potresov v državi in vpogled v njihove splošne podatke, ampak s številnimi pomanjkljivostmi, omejitvami in možnostmi za izboljšavo.

### 2.1 ARSO potresi

ARSO potresi je oddelek Agencije Republike Slovenije za Okolje zadolžen za potrese. Spletna stran nudi seznam potresov v Sloveniji, prikazuje osnovne podatke o potresu, lokacijo, kje so ga čutili in seizmogram. Pomanjkljivosti aplikacije so omejitev podatkov na območje Slovenije, pomanjkanje podrobnosti potresov, nedokončana spletna stran in ne omogoča iskanja in filtriranja potresov.

### 2.2 EMSC

European-Mediterranean Seismological Centre je Evropski seizmološki center, ki je bil ustanovljen leta 1976 in ima sedež v Parizu. Spletna stran je bila vzpostavljena leta 2009 in prikazuje potrese v Evropsko-sredozemski regiji. Prikazuje nam zemljevid vseh potresov, pri obisku posamezne strani potresa, nam nudi prikaz lastnosti potresov kot so povzetek podatkov, podrobnosti, zemljevidi, grafi, komentarji in slike. Glavna pomanjkljivost aplikacije je zastarel uporabniški vmesnik, ki otežuje navigacijo in uporabo ter prikazovanje statičnih zemljevidov.

### 2.3 ZAMG

ZAMG (nem. Zentralanstalt für Meteorologie und Geodynamik) je Avstrijska državna meteorološka in geofizična služba, ki je bila ustanovljena leta 1851. Spletna aplikacija nam nudi prikaz potresov v Avstriji in Evropi na statičnem zemljevidu, prikaz splošnih podatkov in seizmogram. Omogoča tudi pogled seizmogramov posameznih postaj v živo. Spletna aplikacija je omejena z zastarelim uporabniškim vmesnikom, neodzivnostjo strani in površnostjo podatkov.



### 3 Uporabljene tehnologije

V tem poglavju so na kratko predstavljene tehnologije, knjižnice in orodja, ki sem jih uporabil pri izdelavi spletne aplikacije na strani strežnika in uporabnika. S tem bo struktura in delovanje projekta veliko bolj razumljivo.

#### 3.1 HTML

HTML5 je najnovejša standardizirana verzija označevalnega jezika HTML, ki predstavlja osnovo spletnega dokumenta za končni prikaz v spletnem brskalniku. HTML dokument je sestavljen iz parov značk (npr. `<h1>` in `</h1>`) s katerimi definiramo elemente in njihovo privzeto obnašanje. Vsebina parov značk je po navadi besedilo, ki ga želimo prikazati, lahko so pa tudi gnezdene značke. (2)

#### 3.2 CSS

CSS spada pod slogovne jezike, s katerim lahko definiramo stil HTML elementov v smislu pravil, kako naj se ti prikažejo v brskalniku. Določamo lahko barve, velikosti, poravnave in druge lastnosti, prav tako pa lahko nadziramo aktivnosti, ki jih uporabnik izvaja nad elementi. CSS strukturo strani loči od njene predstavitve, kar v dokumentu HTML prinese preglednost, olajša urejanje in dodajanje novih stilov. Zmanjša se ponavljanje in dolžina kode, saj stranem omogoča uporabo istih podlog. (3)

#### 3.3 PHP

PHP: Hypertext Preprocessor (4) je odprtokodni skriptni programski jezik, namenjen olajšanemu razvoju dinamičnih spletnih vsebin in zahtevnejših aplikacij. PHP se razlikuje od nečesa, kot je JavaScript na strani odjemalca, po tem, da se koda izvede na strežniku in ustvari HTML, ki se nato pošlje odjemalcu. Uporabnik prejme rezultate izvajanja tega skripta, vendar ne ve, kaj je izvorna koda. Trenutna verzija je 8.0.3 (5), ki je izšla 4. marca 2021.

#### 3.4 JavaScript

JavaScript (6) je objektni skriptni programski jezik, ki ga je leta 1995 razvilo podjetje Netscape. Večinoma se uporablja na strani odjemalca, spletni strani omogoča komunikacijo z uporabnikom. Z njim lahko dinamično spreminjamo DOM, izvajamo animacije, spreminjamo izgled in pozicijo elementov, jih dodamo ali odstranimo, pošljemo dogodke, izvajamo računske in logične operacije ipd. Trenutna stabilna različica je ECMAScript 2020, ki je izšla v juniju 2020. (7)

### 3.5 Bootstrap

Bootstrap (8) je odprtokodno ogrodje (angl. Framework), namenjeno izdelavi odzivnih spletnih stani. Vsebuje stilske predloge v obliki razredov CSS, uporaba katerih omogoči hitrejši razvoj spletnih strani in aplikacij. Nekatere funkcije kot so pogovorna okna in namigi, potrebujejo še JavaScript. Beta različica 5.0 je izšla 16. junija 2020. (9)

### 3.6 Leaflet

Leaflet.js (10) je odprtokodna JavaScript knjižnica za ustvarjanje interaktivnih zemljevidov. Ustvaril jo je Ukrajinski programer Vladimir Agafonkin. Najnovejša različica - Leaflet 1.7.1 je izšla 4. septembra 2020. (11)

### 3.7 MySQL

MySQL (12) je odprtokodna implementacija relacijske podatkovne baze, ki za delovanje oziroma za delo s podatki uporablja strukturirani povpraševali jezik SQL (angl. Structured Query Language). Prva različica je izšla leta 1995, zadnja različica 8.0.23 pa januarja 2021. (13)

### 3.8 XML

XML oziroma eXtensible Markup Language je označevalni jezik, ki se v večini uporablja za shranjevanje in prenašanje podatkov. Podoben je HTML-ju, vendar brez vnaprej določenih oznak za uporabo. Namesto tega določimo lastne oznake, oblikovane posebej za naše potrebe. Osnovna oblika XML je standardizirana, če delimo dokument, prejemnik lahko še vedno razčleni podatke zaradi standardizirane sintakse XML. (14)

### 3.9 REST

Representational state transfer je arhitekturni stil, ki se uporablja za izdelavo spletnih vmesnikov oz. API. To je skupek pravil, ki jim morata strežnik in odjemalec slediti, da se lahko med seboj pravilno sporazumevata. Odjemalec pošlje zahtevo na vnaprej določen osnoven URL na katerega doda še želene parametre, strežnik nato odgovori če je poizvedba bila uspešna (npr. koda 200) in če smo vnesli pravilne parametre, nam vrne podatke. (15)

## 4 Teorija

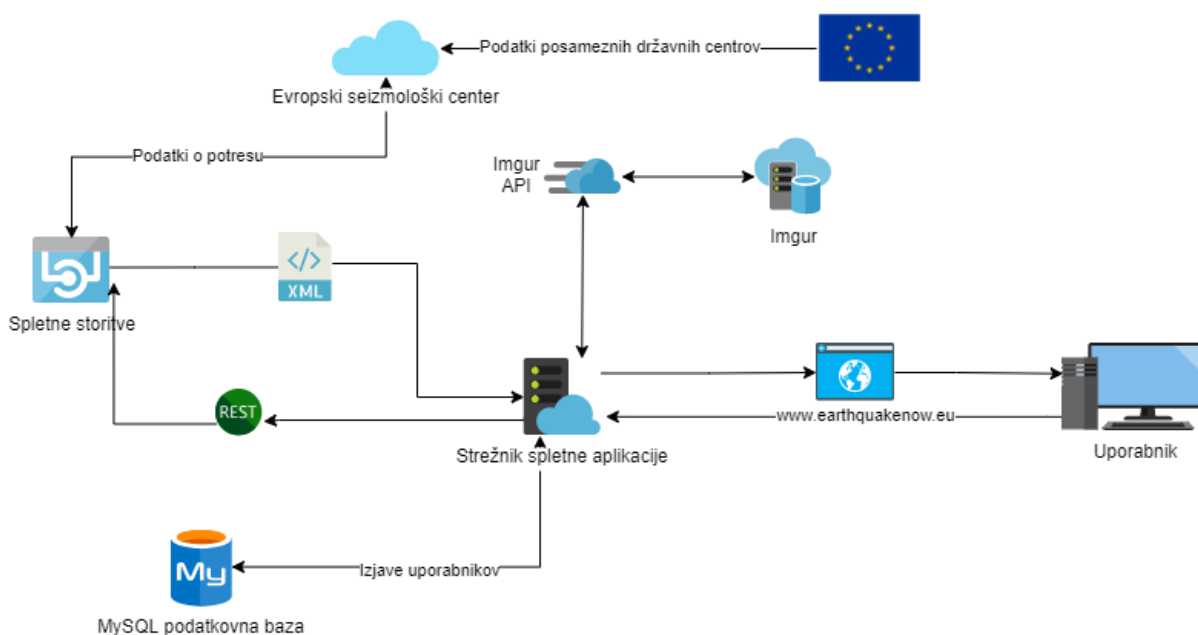
V tem poglavju bo predstavljeno delovanje spletne aplikacije s teoretične strani. Najprej bo predstavljeno splošno delovanje, nato sledi podroben opis poizvedbe in podatkovne baze.

### 4.1 Struktura

Strukturo spletne aplikacije lahko razdelimo na tri glavne komponente:

1. Strežnik s samo aplikacijo, ki bo podatke obdelovala in prikazovala
2. Spletne storitve, ki omogočajo dostop do podatkov o potresih
3. MySQL podatkovna baza za izjave in shranjevanje slik na Imgur API

Slika 1 prikazuje poenostavljeno delovanje spletne aplikacije. Ko strežnik prejme zahtevo od uporabnika, bo glede na zahtevo naredil REST poizvedbo na spletne storitve EMSC. Spletne storitve nam vrnejo iskane podatke v izbranem formatu XML, ki ga nato obdelamo na strani strežnika s pomočjo programskega jezika PHP. Po potrebi bo PHP izpisal ali vpisal podatke iz MySQL podatkovne baze, ki hrani izjave in povezave do slik uporabnikov. Če uporabnik naloži novo sliko, bo PHP poskrbel za nalaganje slike na Imgur gostovanje. Nato bo PHP podatke izpisal v HTML dokumentu in jih poslal v JavaScript, ki jih bo nato kartiral in prikazal na zemljevidu. Na koncu še strežnik vrne generirano spletno stran uporabniku.

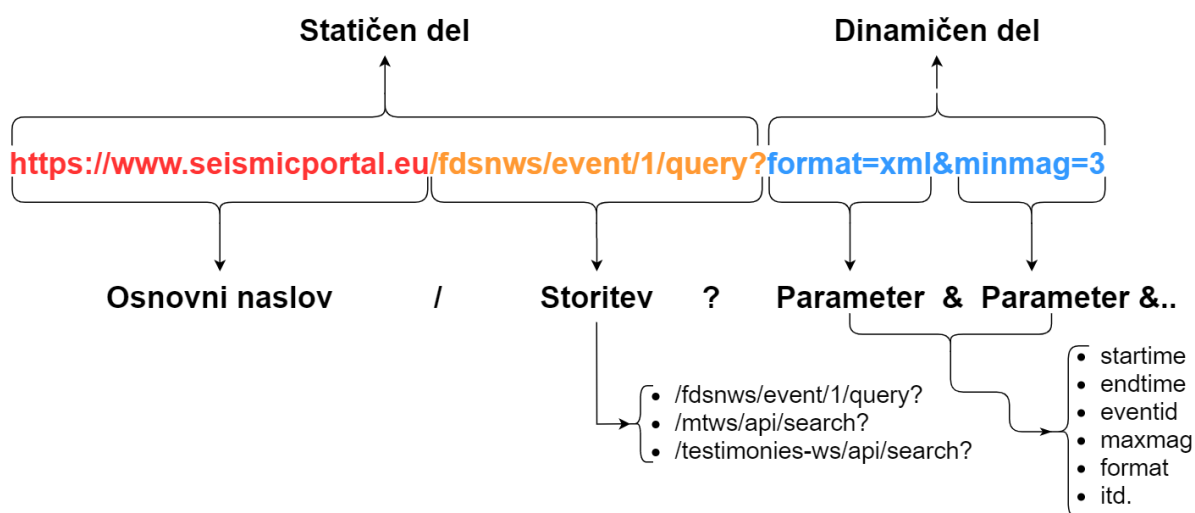


Slika 1: Struktura aplikacije (Lastni vir)

## 4.2 Podatki

EMSC deluje kot center za shranjevanje in obdelovanje podatkov o potresih iz posameznih evropskih državnih seizmoloških centrov na enem mestu. Podatki so na voljo javnosti pod licenco CC BY-NC, kar pomeni, da podatke lahko prosto uporabljamo za nekomercialne storitve.

Za dostopanje do samih podatkov, bomo uporabili spletne storitve (angl. Web Services), ki so definirane po specifikacijah FDSN (angl. International Federation of Digital Seismograph Networks). Delujejo na arhitekturnem stilu REST, ki je sestavljen iz statičnega in dinamičnega dela, kot je prikazano na Slika 2.



Slika 2: Zgradba REST poizvedbe (Lastni vir)

Statičen del poizvedbe je sestavljen iz osnovnega naslova API in storitve, do katere želimo dostopati. Ostal bo večinoma enak za vse poizvedbe in se ne bo spreminjal, razen če želimo zamenjati storitev.

Storitve API določajo vrsto podatkov do katerih želimo dostopati in se delijo na:

- fdsnws – Splošni podatki o potresu kot so moč, lokacija, globina, agencija, itd.
- mtws – Podatki momentnega tenzorja, ki predstavljajo deformacijo v regiji, ki je generirala seizmične valove.
- testimonies-ws – Število, čas in lokacija uporabnikov ki so čutili potres.

Dinamičen del je sestavljen iz enega ali več parametrov, ki delujejo kot filtri za podatke. Parametri lahko določajo datum dogodka, čas, moč, geografske omejitve, itd. in jih

bomo generirali s funkcijami, da bi zožili izbor podatkov. Poizvedba na Slika 2 bi nam vrnila vse splošne podatke o potresih z magnitudo večjo od 3 v XML formatu.

API vrne odgovor poizvedbe v enem od treh formatov: TEXT, JSON ali XML. Za aplikacijo bomo uporabljali XML, saj ima enostavno in človeku berljivo strukturo ter omogoča iskanje elementov po značkah.

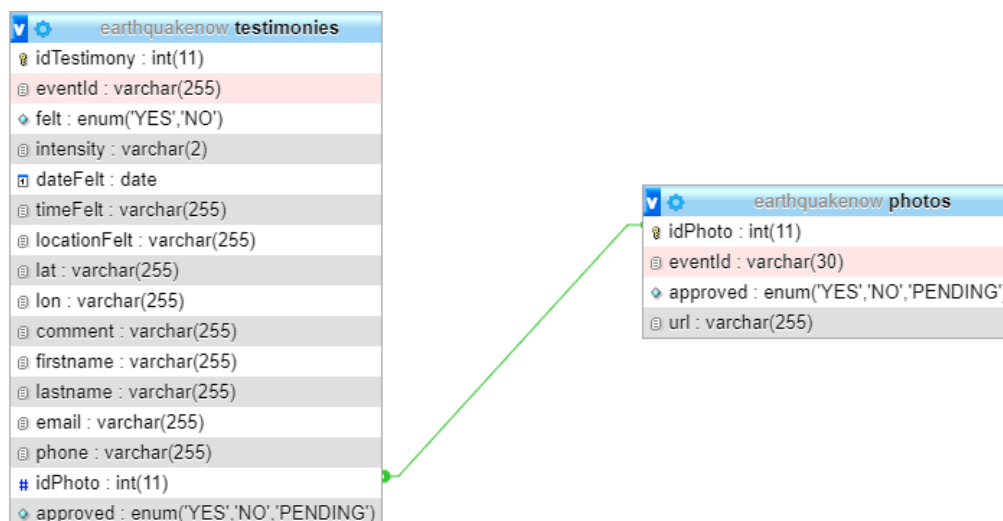
Znotraj XML dokumenta je vsak potres določen znotraj `<event>` značke, ta je pa sestavljena iz več gnezdenih elementov ki vsebujejo podatke o potresu kot so čas, lokacija, globina, magnituda, itd. Skripta 1 prikazuje strukturo elementa za magnitudo znotraj značke `<event>`.

```
<event>
  <magnitude publicID="quakeml:eu.emsc/event/20210405_0000103/YJTUO">
    <mag>
      <value>3.6</value>
    </mag>
    <type>ml</type>
    <originID>quakeml:eu.emsc/event/20210405_0000103/origin/YJTUO</originID>
    <stationCount>26</stationCount>
    <creationInfo>
      <agencyURI>smi:smi-registry/organization/EMSC</agencyURI>
      <authorURI>smi:smi-registry/organization/GNS</authorURI>
    </creationInfo>
  </magnitude>
</event>
```

*Skripta 1: Struktura deleža XML dokumenta*

### 4.3 Podatkovna baza

Zaradi varnosti osebnih podatkov in zasebnosti, spletne storitve ne vsebujejo komentarjev in slik uporabnikov, zato bomo jih shranjevali lokalno v MySQL podatkovni bazi. Podatkovna baza bo vsebovala tabelo za podatke uporabnikov o potresu kot so komentarji, lokacija kje so čutili potres, čas, itd. ter tabelo za slike, ki bo shranjevala povezavo do slike. Da bi preprečili neprimerne in napačne podatke, vsaka tabela vsebuje atribut `approved` s katerim bomo regulirali kaj bo spletna aplikacija izpisala. Slika 3 prikazuje relacijski model podatkovne baze za izjave in slike.



Slika 3: Relacijski model podatkovne baze

Atributi tabele za izjave (testimonies):

- **idTestimony** – Enolični id vsake izjave
- **eventId** – Enolični id vsakega potresa
- **felt** – Če je uporabnik čutil potres
- **intensity** – Moč potresa, ki ga je čutil uporabnik
- **dateFelt, timeFelt** – Datum in čas kdaj je uporabnik čutil potres
- **lat, lon** – Latitude in Longitude – lokacija za kartiranje
- **comment** – Komentar uporabnika
- **firstname, lastname** – Ime in priimek uporabnika (Opcijsko)
- **email, phone** – E-pošta in telefon uporabnika (Opcijsko)
- **idPhoto** – Tuji ključ za povezovanje slik
- **approved** – Če je izjava odobrena, DA ali NE

Atributi tabele za slike (photos):

- **idPhoto** – Enolični id vsake slike
- **eventId** – Enolični id posameznega potresa
- **approved** – Če je slika odobrena, DA ali NE
- **url** – Ker shranjevanje slik v podatkovni bazi nebi bilo učinkovito, bomo shranjevali samo povezave URL do posamezne slike, ki bo shranjena na spletnih storitvah npr. Imgur.

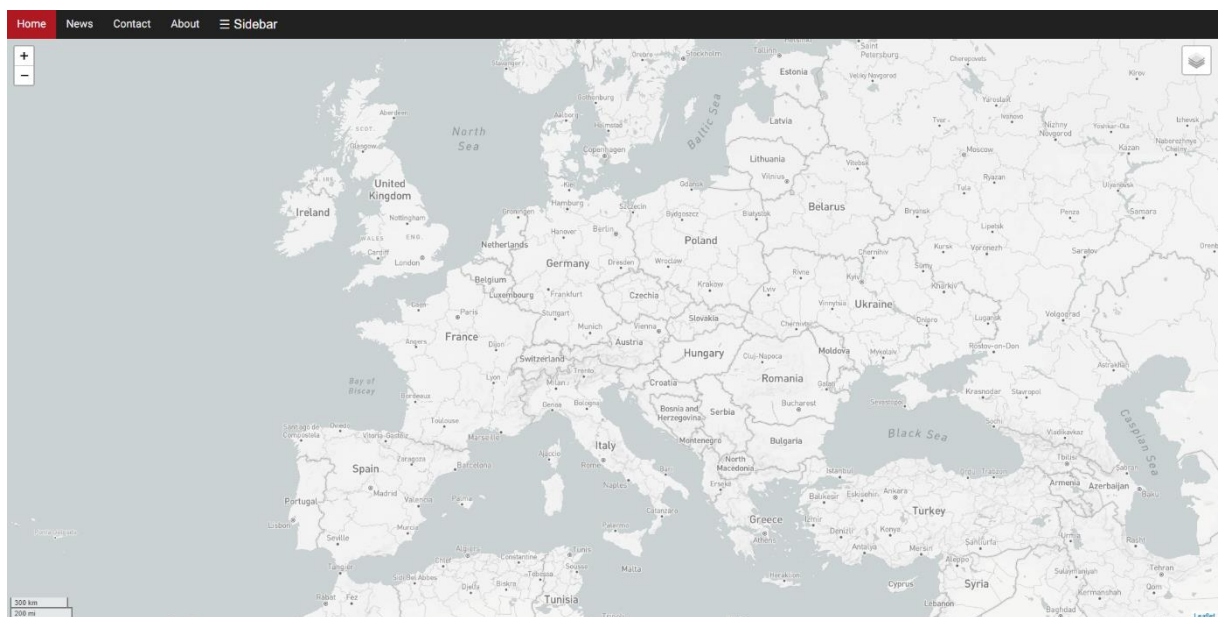
## 5 Razvoj

### 5.1 Zemljevid

Za uporabo Leaflet zemljevida, moramo najprej vključiti CDN za CSS in JS v glavi HTML dokumenta in v `<body>` dodati `<div id="mapid">` s poljubnim ID-jem v katerem bo kasneje zemljevid. Leaflet zemljevidi so sestavljeni iz več slik, katerim rečemo ploščice (angl. tiles). Da lahko prikažemo zemljevid, moramo najprej definirati ploščice v spremenljivki `darkMap`, kot je prikazano na Skripti 2. Nato lahko ustvarimo zemljevid `mymap`, ki vsebuje ploščice iz `darkMap` in se navezuje na element "mapid". Ker so ploščice večje, kot jih podpira Leaflet, moramo uporabiti `zoomOffset: -1` za prilagoditev. Slika 4 prikazuje zemljevid s ploščicam iz `lightMap`.

```
var darkMap = L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}', {
  maxZoom: 18,
  id: 'mapbox/dark-v10', // Type of map
  tileSize: 512,
  zoomOffset: -1,
  accessToken: 'pk.eyJ1IjoieYXNoaWwYXV5IiwiaW50IiwiaWF0Ij0iYWFwdWlqYjB1dHYzMnM5ZmhybXoweX' });
var mymap = L.map("mapid", {
  center: [earthquakes[0][1], earthquakes[0][2]], //Lat, Lon from earthquakes array
  zoom: 3,
  layers: [darkMap] }); // default layer
```

Skripta 2: Definiranje zemljevida



Slika 4: Prva verzija aplikacije

## 5.2 Podatki

Zajemanje podatkov bomo izvedli s pomočjo funkcije `getEarthquakes`, ki prejme kot parameter `$url` povezavo do API. Funkcija najprej poišče in shrani vse značke z imenom "event", ki predstavljajo potrese (prikazano na Skripta 1) v spremenljivko `$events`. Nato iritira skozi vsak potres, in iz njega shrani iskane lastnosti. Na koncu funkcija vrne 2D polje `$items`, ki vsebuje vse podatke. To funkcijo bomo večkrat uporabili v spletni aplikaciji za pridobivanje podatkov o posameznih potresih.

```
function getEarthquakes($url) {
    $xml = new DOMDocument();
    //@ error suppression
    if (@$xml->load($url) === false) {
        return "Error"; }
    $xml->load($url);
    $items = array();

    $events = $xml->getElementsByTagName('event');
    foreach ($events as $event) {
        $preferredOriginID = $event->getElementsByTagName('preferredOriginID')->item(0)->nodeValue;
        $eventid = explode("/", $preferredOriginID);
        $eventid = $eventid[2];
        $lat = $event->getElementsByTagName('latitude')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;
        $lon = $event->getElementsByTagName('longitude')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;
        $description = $event->getElementsByTagName('description')->item(0);
        $location = $description->getElementsByTagName('text')->item(0)->nodeValue;
        $date1 = date_create($event->getElementsByTagName('time')->item(0)->getElementsByTagName('value')->item(0)->nodeValue);
        $date = $date1->format('d-m-Y H:i:s');
        $depth = $event->getElementsByTagName('depth')->item(0)->nodeValue;
        $mag = $event->getElementsByTagName('magnitude')->item(0)->getElementsByTagName('mag')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;

        $earthquake = array();
        array_push($earthquake, $eventid, $lat, $lon, $location, $date, $depth, $mag);
        $items[] = $earthquake;
    }
    return $items;
}
```

*Skripta 3: Funkcija za pridobivanje podatkov*



### 5.3 Izpisovanje in kartiranje

Podatke generiramo glede na filter, ki ga je uporabnik izbral in nato rezultat shranimo v polje `earthquakesArr`. Polje bomo nato posredovali v JavaScript in ga uporabili, da izpišemo potrese v stranski vrstici.

```
<script> var earthquakes = <?php if (isset($_GET['submitCustomFilter'])) {
    $earthquakesArr = getEarthquakes(generateQuery());
    echo json_encode($earthquakesArr);
} else {
    if (isset($_GET['quickFilter'])) {
        $earthquakesArr = getEarthquakes(returnUrl($_GET['quickFilter'])[0]);
        echo json_encode($earthquakesArr);
    } else {
        $earthquakesArr = getEarthquakes($urlQuakeToday);
        echo json_encode($earthquakesArr);}}?>; </script>

<?php // Fill in sidebar with cards containing earthquake data
foreach ($earthquakesArr as $earthquake) {
    echo '<div class="card w-100 mt-1 item darkItem border-
0" id=' . $earthquake[0] . ' "> <div class="card-body darkItem"> <h6 class="card-
title"><span class="badge ' . getColor($earthquake[6])[1] . '">'
. $earthquake[6] . '</span>' . " " . $earthquake[3] . '</h6><p class="card-text"></p> <p class="card-
text">' . $earthquake[4] . '</p></div></div>'; } ?>
```

*Skripta 4: Posredovanje JS in izpisovanje stranske vrstice*

Skripta 5 prikazuje primer strukture zapisa v polju `earthquakesArr`. Potresi so ločeni v posamezna polja znotraj enega velikega polja. Do lastnosti potresov dostopamo z indeksi.

```
Array (
[0] => Array ( [0] => 20210417_0000084 [1] => 5.19 [2] => 127.57 [3] => PHILIPPINE
ISLANDS REGION [4] => 17-04-2021 09:57:43 [5] => 150000.0 [6] => 4.8 )
[1] => Array ( [0] => 20210417_0000078 [1] => 6.8 [2] => -73.02 [3] => NORTHERN
COLOMBIA [4] => 17-04-2021 09:17:13 [5] => 149000.0 [6] => 4.3 )
[2] => Array ( [0] => 20210417_0000062 [1] => -4.98 [2] => 129.7 [3] => BANDA
SEA [4] => 17-04-2021 06:48:02 [5] => 213000.0 [6] => 4.4 )
[3] => Array ( [0] => 20210417_0000028 [1] => -30.42 [2] => -72.22 [3] =>
OFFSHORE COQUIMBO, CHILE [4] => 17-04-2021 03:50:55 [5] => 2000.0 [6] => 4.9 ))
)
```

*Skripta 5: Struktura polja*

Nato funkcija `mapEarthquakes` (Skripta 6) gre skozi polje definirano v Skripta 4, in za vsakega doda točko na mapi (circle) ter podatke, ki se prikažejo v pojavnem oknu ob kliku na točko. Barva točke je določena v funkciji `getColor` glede na moč potresa. Za iteracijo smo uporabili jQuery, ker bi z navadnim JS kasneje naleteli na asinhrono težavo.

```
function mapEarthquakes(earthquakes) {
    var onLoad = false;
    function getColor(mag) {
        return mag >= 8 ? ["#FF0000", "eqGreat"]:
            mag >= 7 ? ["#FF6000", "eqMajor"]:
            mag >= 6 ? ["#FFA000", "eqStrong"]:
            mag >= 5 ? ["#FFF071", "eqModerate"]:
            mag >= 4 ? ["#FFE500", "eqLight"]:
            mag >= 2 ? ["#1b8700f1", "eqMinor"]:
            ["#A2A2A2", "eqMicro"];
    }
    $(earthquakes).each(function () {
        var earthquake = $(this);
        var eventid = earthquake[0];
        var lat = earthquake[1];
        var lon = earthquake[2];
        var location = earthquake[3];
        var date = earthquake[4];
        var depth = earthquake[5] / 1000;
        var mag = earthquake[6];
        var markers = [];
        var circle = L.circle([lat, lon], {
            color: getColor(mag)[0],
            fillColor: getColor(mag)[0],
            fillOpacity: 0.7,
            radius: 3000 * mag,
            title: eventid
        }).addTo(markerGroup);
        circle.bindPopup('<p class="mb-1"><span class="fw-
bolder">' + location + '</span><br>Magnitude: ' + mag + '<br>Date: ' + date +
        '<br>Depth: ' + depth + ' km <div class="text-center"><a class="btn btn-
primary" style="color:white;" role="button" href="earthquake.php?id=' + eventid + '">Details</a></div>
</p>');
        markers.push(circle);
    });
}
```

```

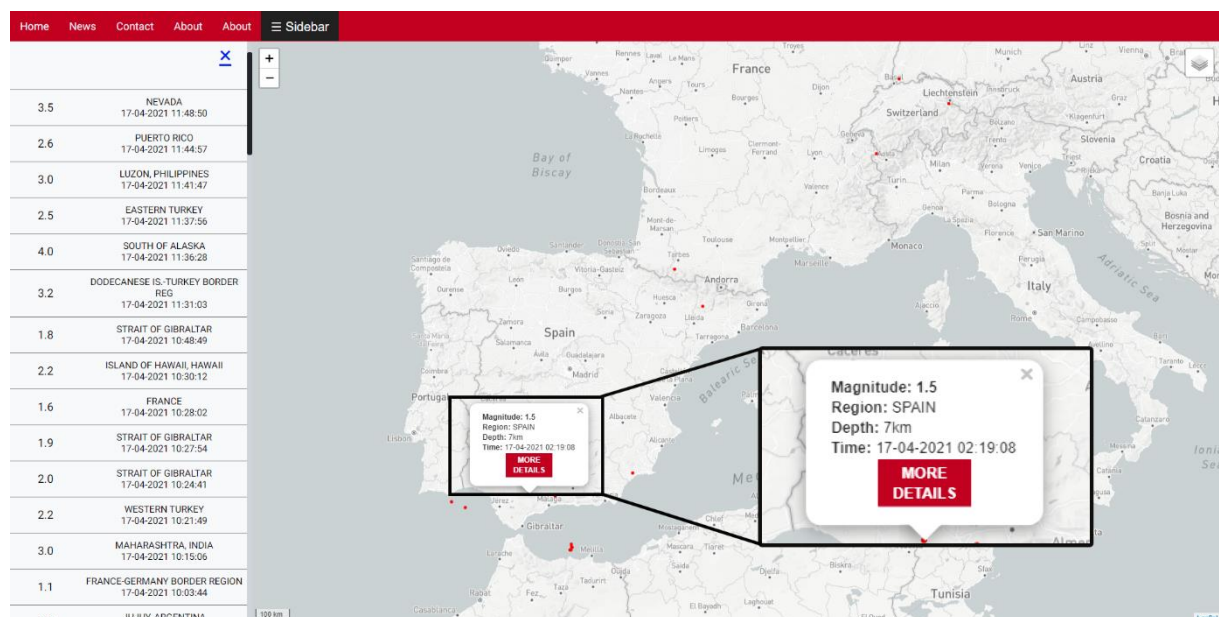
//Fly map to marker with X id
function gotoMarker(id) {
  for (var i in markers) {
    var markerID = markers[i].options.title;
    if (markerID == id) {
      mymap.flyTo([lat, lon], 6, {
        animate: true,
        duration: 1 });
      markers[i].openPopup();
    }
  }
}

//on click item from sidebar, go to it on map // remove and set active class
$('.item').click(function () {
  gotoMarker($(this)[0].id);
  $(this).siblings('.item').removeClass('activeItem');
  $(this).addClass('activeItem');
});
if (!onLoad) { //On load fly to first marker and open the popup
  markers[0].openPopup();
  onLoad = true;
}
}); //end for each //end function

```

*Skripta 6: Funkcija za kartiranje potresov*

Aplikacija prikazuje točke na lokacijah potresov in tektonske plošče (oranžna črta) ter vsebuje seznam potresov v stranski vrstici. Če uporabnik klikne na potres v stranski vrstici, bo zemljevid "letel" do točke potresa in se bo odprlo pojavno okno kot je vidno na Slika 5.



*Slika 5: Druga verzija aplikacije*

Pojavno okno vsebuje gumb, ki nam bo odprl novo stran `earthquake.php` za podrobnosti potresa. Značka `<a>` vsebuje href atribut z enoličnim ID-jem s katerim bomo ločili in prikazali potrese. ID bomo vrnili v URL in ga v `earthquake.php` izvlekli s pomočjo GET ter uporabili za poizvedbo. Poizvedba bo vsebovala parameter 'eventid', ki bo enak ID-ju.

Primer href: `earthquake.php?id=20210401_0000126`

Funkcijo `getEarthquakes` lahko ponovno uporabimo za posamezne potrese, saj je struktura dokumenta skoraj enaka. Ker želimo prikazati vse podrobnosti o potresu, bomo uporabili še funkcije za pridobivanje momentnih tenzorjev in izvorov dogodka. Skripta 7 prikazuje funkcijo za momentne tenzorje, ker odgovor API ni v XML, je funkcija nekoliko drugačna.

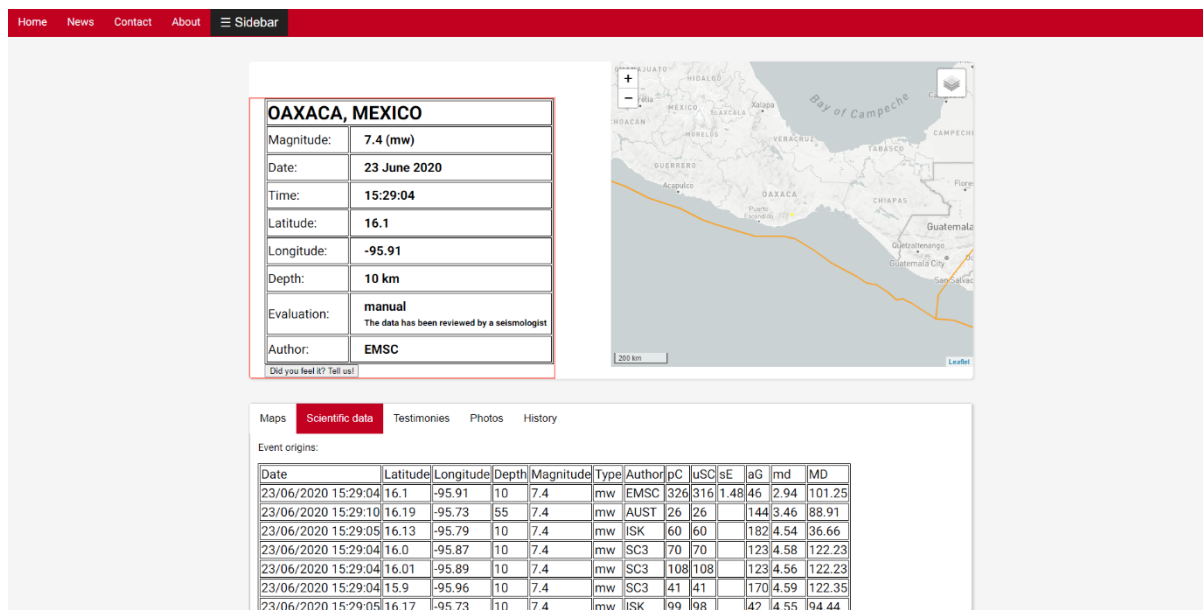
```
function getMomentTensors($url)
{
    $pieces = explode("|", file_get_contents($url));
    //check if empty by checking index beyond header
    if (array_key_exists('40', $pieces)) {
        $mt = array_slice($pieces, 40); //delete first 40 elements
        $mt = array_chunk($mt, 39); // split array into pieces 39 size
        return $mt;
    } else {
        return null;
    }
}
```

*Skripta 7: Funkcija za momentne tenzorje*

Skripta 8 prikazuje primer zapisa momentnih tenzorjev.

```
#ev_unid|ev_region|ev_event_time|ev_latitude|ev_longitude|ev_mag_type|ev_mag_value|ev_depth|mt_source_catalog|mt_region|mt_centroid_time|mt_latitude|mt_longitude|mt_mw|mt_depth|mt_strike_1|mt_dip_1|mt_rake_1|mt_strike_2|mt_dip_2|mt_rake_2|mt_tval|mt_tplung|mt_taz|mt_pval|mt_pplung|mt_paz|mt_nval|mt_nplung|mt_naz|mt_mrr|mt_mtt|mt_mpp|mt_mrt|mt_mrp|mt_mtp|mt_per_dc|mt_per_clvd|mt_per_iso|mt_m0
20170919_0000091|PUEBLA, MEXICO|2017-09-19 18:14:38.5 UTC|18.59|-98.47|mw|7.1|50|GCMT|PUEBLA, MEXICO|2017-09-19 18:14:47.1 UTC|18.51|-98.62|7.1|53|108|44|-98|300|47|-82|6.276e19|1|23|-6.506e19|84|281|0.231e19|5|114|-6.43e19|5.28e19|1.15e19|-0.006e19|-0.742e19|-2.25e19|92.91|7.09|0|6.39e19
```

*Skripta 8: Struktura zapisa momentnih tenzorjev*



Slika 6: Druga verzija podrobnosti

Slika 6 prikazuje izpis podrobnosti o potresu iz funkcije `getEarthquakes` poleg zemljevida ki označuje epicenter. Tabela v zavihku "Scientific data" je izpis funkcije `getOrigins` prikazana na Skripta 9. Funkcija deluje na podoben način kot `getEarthquakes` ampak je bolj obsežna, pod tabelo so še podatki momentnega tenzorja.

```
function getOrigins($url, $onlyLocation = False) {
    $xmlOrigin = new DOMDocument();
    $xmlOrigin->load($url);
    $items = array();
    $origins = $xmlOrigin->getElementsByTagName("origin");
    //if true only return location of origins
    if ($onlyLocation) {
        foreach ($origins as $origin) {
            $latOrigin = $origin->getElementsByTagName('latitude')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;
            $lonOrigin = $origin->getElementsByTagName('longitude')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;
            $item = array();
            array_push($item, $latOrigin, $lonOrigin);
            $items[] = $item;
        }
        return $items;
    }
}
```

```

//gets child elements of parent only if they exist
function getChildValue($parent, $child) {
    if (isset($parent->getElementsByTagName($child)->item(0)->nodeValue)) {
        return $parent->getElementsByTagName($child)->item(0)->nodeValue;
    } else { return null; }}

foreach ($origins as $origin) {
    $latOrigin = $origin->getElementsByTagName('latitude')->item(0)->nodeValue;
    $lonOrigin = $origin->getElementsByTagName('longitude')->item(0)->nodeValue;
    $date = date_create($origin->getElementsByTagName('time')->item(0)->getElementsByTagName('value')->item(0)->nodeValue);
    $description = $xmlOrigin->getElementsByTagName('description')->item(0);
    $region = $description->getElementsByTagName('text')->item(0)->nodeValue;
    $regionType = $description->getElementsByTagName('type')->item(0)->nodeValue;
    $date = date_create($origin->getElementsByTagName('time')->item(0)->getElementsByTagName('value')->item(0)->nodeValue);
    $date = $date->format('d/m/Y H:i:s');
    $depth = $origin->getElementsByTagName('depth')->item(0)->nodeValue;
    $mag = $xmlOrigin->getElementsByTagName('mag')->item(0)->getElementsByTagName('value')->item(0)->nodeValue;
    $magType = $xmlOrigin->getElementsByTagName('magnitude')->item(0)->getElementsByTagName('type')->item(0)->nodeValue;
    $author = $origin->getElementsByTagName('creationInfo')->item(0)->getElementsByTagName('authorURI')->item(0)->nodeValue;
    $author = explode("/", $author)[2];
    $quality = $origin->getElementsByTagName('quality')->item(0);
    $usedPhaseCount = getChildValue($quality, 'usedPhaseCount');
    $usedStationCount = getChildValue($quality, 'usedStationCount');
    $standardError = getChildValue($quality, 'standardError');
    $azimuthalGap = getChildValue($quality, 'azimuthalGap');
    $minimumDistance = getChildValue($quality, 'minimumDistance');
    $maximumDistance = getChildValue($quality, 'maximumDistance');

    $item = array();
    array_push($item, $date, $latOrigin, $lonOrigin, $depth, $mag, $magType, $author, $usedPhaseCount, $usedStationCount, $standardError, $azimuthalGap, $minimumDistance, $maximumDistance);
    $items[] = $item;
}
return $items;
}

```

*Skripta 9: Funkcija za izvor dogodka*

Odzive uporabnikov, ki so čutili potres dobimo s pomočjo storitve testimonies-ws. Ta nam vrne stisnjeno datoteko v kateri je besedilni dokument z podatki. Funkcija prikazana na Skripta 10 najprej stisnjeno datoteko shrani v začasno mapo, jo ekstrahira<sup>1</sup> in shrani vsebino besedilnega dokumenta. Nato izbriše vse datoteke in nam vrne rezultat.

```
function downloadUnzipGetContents($url, $id) {
    $data = file_get_contents($url);
    $path = tempnam(sys_get_temp_dir(), 'prefix');
    $temp = fopen($path, 'w');
    fwrite($temp, $data);
    fseek($temp, 0);
    fclose($temp);
    $pathExtracted = tempnam(sys_get_temp_dir(), 'prefix');
    $filenameInsideZip = $id . '.txt';
    $success = @copy("zip://" . $path . "#." . $filenameInsideZip, $pathExtracted);
    if (!$success) {
        return false;
    } else {
        $data = file_get_contents($pathExtracted);
        unlink($path);
        unlink($pathExtracted);
        return $data;
    }
};
```

*Skripta 10: Funkcija za pridobivanje odzivov*

Odzive nato posredujemo v JavaScript in jih kartiramo kar prikazuje Skripta 11.

```
if (testimonies !== false && typeof testimonies !== 'undefined' && testimonies !== null) {
    var testimonies = testimonies.trim().split(/\n/);
    testimonies = testimonies.slice(4);
    for (var testimony of testimonies) {
        testimony = testimony.split(',');
        circle = L.circle([testimony[1], testimony[0]], {
            color: '#4DD7FF',
            fillColor: '#4DD7FF',
            fillOpacity: 0.5,
            radius: 500,
        }).addTo(testimonyGroup);
    }
}
```

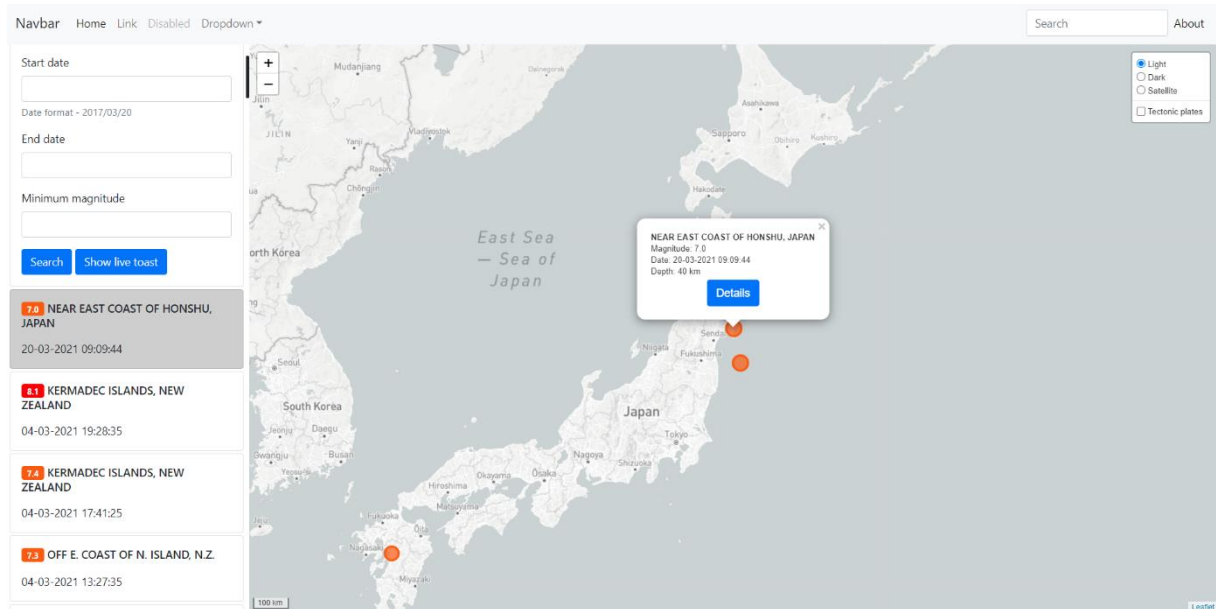
*Skripta 11: Kartiranje odzivov*

---

<sup>1</sup> Postopek pri katerem se z raztezanjem podatkov obnovita izvorna oblika in velikost datoteke.

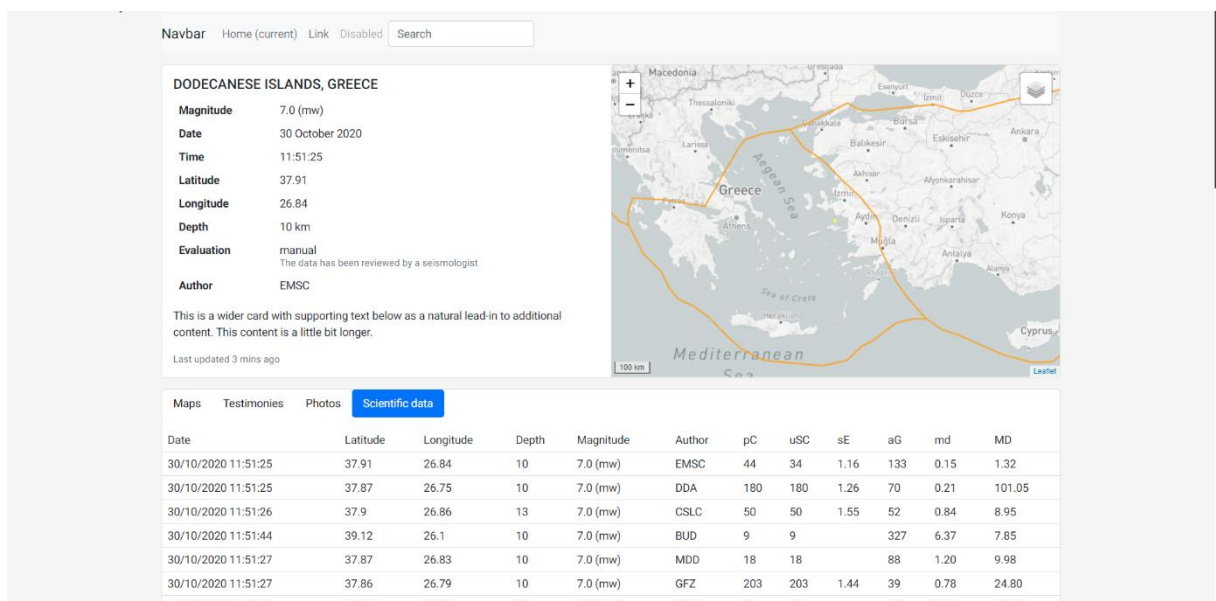
## 5.4 Bootstrap

Zaradi estetskih in funkcionalnih razlogov smo dodali še Bootstrap. Elementi stranske vrstice so iz razreda "card", ki zglejajo kot karte na podlagi in dajejo elementom globino ter občutek izstopanja s pomočjo barv, senc, obrob in odzivnosti.



Slika 7: Tretja verzija aplikacije

Efekt izstopanja in globine je bolj opazen pri Slika 8, kjer smo tudi uporabili karte za poudarjanje elementov. Daje nam občutek zgradbe plasti in v ospredje postavi za uporabnika pomembne podatke. Bootstrap bo poleg tega še zagotovil pravilno in dinamično razporejanje elementov glede na velikost zaslona.



Slika 8: Tretja verzija - podrobnosti



## Napredni filter

Za generiranje poizvedb bomo uporabili funkcijo `generateQuery`, ki jo bomo poklicali če uporabnik s pomočjo obrazca vnese filter kot je prikazano na Skripta 4. Funkcija iz obrazca izvleče parametre s pomočjo GET in ustvari polje `$query`. Ker poizvedba potrebuje točno določena imena parametrov, bomo uporabili `compact`, ki nam omogoči, da v polje shranimo parametre s ključem ki je enak imenu spremenljivke in vsebuje vrednost spremenljivke. Primer: Če imamo spremenljivko `$maxmag = 8`, bo ta zapisana v polje s ključem "maxmag" in vrednostjo "8". Potem lahko s tem poljem ustvarimo poizvedbo, kot je prikazano na Skripta 12.

```
function generateQuery($onlyParamater = false) {
    $url = "https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&limit=100&";
    $start = $_GET['start'];
    $end = $_GET['end'];
    $minmag = $_GET['minmag'];
    $maxmag = $_GET['maxmag'];
    $magtype = $_GET['magtype'];
    $mindepth = $_GET['mindepth'];
    $maxdepth = $_GET['maxdepth'];
    $orderby = $_GET['orderby'];
    $minlat = $_GET['minlat'];
    $maxlat = $_GET['maxlat'];
    $minlon = $_GET['minlon'];
    $maxlon = $_GET['maxlon'];
    $lat = $_GET['lat'];
    $lon = $_GET['lon'];
    $minradius = $_GET['minradius'];
    $maxradius = $_GET['maxradius'];
    //compact , filter removes empty paramaters
    $query = compact("orderby", "start", "end", "minmag", "maxmag", "magtype", "mindepth", "maxdepth",
"minlat", "maxlat", "minlon", "maxlon", "lat", "lon", "minradius", "maxradius");
    $query = array_filter($query, fn ($value) => !empty($value));
    //if true returns only parameter and value, else build query
    if ($onlyParamater) {
        return $query;
    } else {
        $test = http_build_query($query) . "\n";
        return $url . $test;
    }
}
```

Skripta 12: Funkcija `generateQuery`

## Hitri filter

Za najpomembnejše in najpogostejše filtre bomo ustvarili še hitri filter, ki bo vseboval spustni seznam in na izbiro nekaj vnaprej definiranih poizvedb. Na ta način bo uporabnik lahko hitro prišel do podatkov kot so najmočnejši potresi in najnovejši potresi. Skripta 13 prikazuje funkcijo `returnUrl`, ki glede na prejet parameter vrne URL z vnaprej definirano poizvedbo.

```
function returnUrl($parameters)
{
    $dateNowStart = (new \DateTime())->modify('-24 hours')->format('Y-m-d');
    $url = "https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&start=" . $dateNowStart;
    $urlEnd = "&end=" . (new \DateTime())->modify('+24 hours')->format('Y-m-d'). "&limit=50";
    switch ($parameters) {
        case "tall":
            return [$url . $urlEnd . '&orderby=time', "Most recent (Today)"];
            break;
        case "stoday":
            return [$url . $urlEnd . '&orderby=magnitude', 'Strongest (Today)'];
            break;
        case "sweek":
            return ["https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&start=" . (new \DateTime())->modify('-1 week')->format('Y-m-d') . $urlEnd . '&orderby=magnitude', 'Strongest (Week)'];
            break;
        case "smoth":
            return ["https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&start=" . (new \DateTime())->modify('-1 month')->format('Y-m-d') . $urlEnd . '&orderby=magnitude', 'Strongest (Month)'];
            break;
        case "syear":
            return ["https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&start=" . (new \DateTime())->modify('-1 year')->format('Y-m-d') . $urlEnd . '&orderby=magnitude', 'Strongest (Year)'];
            break;
        case "sall":
            return ['https://www.seismicportal.eu/fdsnws/event/1/query?format=xml&minmag=7&orderby=magnitude&limit=50', 'Strongest (All time)'];
            break;
        case "reu":
            return [$url . $urlEnd . '&minlat=10&minlon=-30&maxlon=65', 'Region (Europe)'];
            break;
        default:
            echo "Error";
    }
}
```

Skripta 13: Funkcija za hitri filter

## 5.5 Izjave

Za shranjevanje izjav uporabnikov moramo najprej ustvariti podatkovno bazo in tabele s pomočjo MySQL ukazov prikazanih na Skripta 14.

```
CREATE DATABASE earthquakenow;
CREATE TABLE photos (
    idPhoto INT PRIMARY KEY AUTO_INCREMENT,
    eventId VARCHAR(255) NOT NULL,
    urlPhoto VARCHAR(255) NOT NULL,
    approved ENUM('YES', 'NO', 'PENDING')
);
CREATE TABLE testimonies (
    idTestimony INT PRIMARY KEY AUTO_INCREMENT,
    eventId VARCHAR(255) NOT NULL,
    felt ENUM('YES', 'NO'),
    approved ENUM('YES', 'NO', 'PENDING'),
    dateFelt DATE,
    timeFelt TIME,
    locationFelt VARCHAR(255),
    lat VARCHAR(255),
    lon VARCHAR(255),
    comment VARCHAR(255),
    firstname VARCHAR(255),
    lastname VARCHAR(255),
    email VARCHAR(255),
    phone VARCHAR(255),
    idPhoto INT,
    FOREIGN KEY (idPhoto) REFERENCES photos(idPhoto)
);
```

*Skripta 14: Ustvarjanje podatkovne baze*

Za povezavo do podatkovne baze bomo uporabili funkcijo `db_connect`, ki je prikazana na Skripta 15.

```
function db_connect(){
    $conn = mysqli_connect("localhost", "root", "", "earthquakenow");
    if(!$conn){
        echo "Can't connect database " . mysqli_connect_error($conn);
        exit;
    }
    return $conn;
}
```

*Skripta 15: Funkcija za povezovanje do podatkovne baze*

Za vnašanje podatkov bomo uporabili obrazec, ki je prikazan na Slika 9. Obrazec bo znotraj pojavnega okna in bo vseboval polja v katera bo uporabnik vpisal kdaj je potres čutil, kje, svoj komentar in če želi še kontaktne informacije. Skripta 16 najprej preveri če je uporabnik kliknil gumb za vpis podatkov, če ja potem generira SQL stavek, ki vsebuje vrednosti iz polj obrazca in jih vpiše v podatkovno bazo. Na koncu skripta še vrne če je vpis bil uspešen ali ne.

Slika 9: Obrazec za vpis podatkov

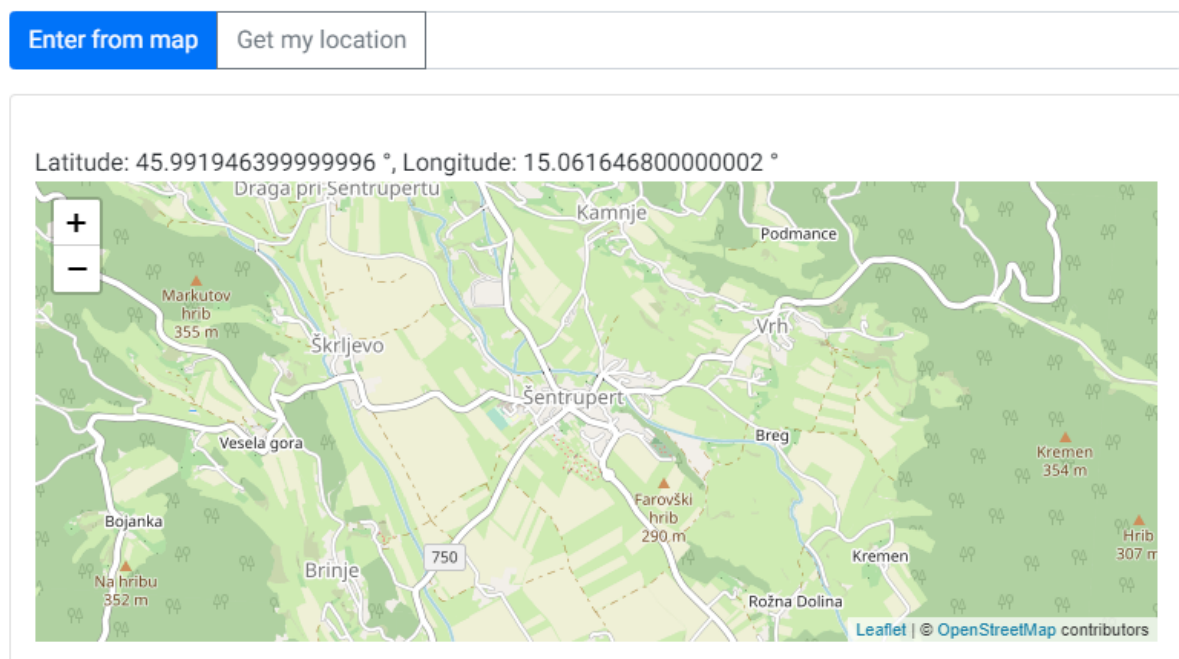
```
if (isset($_POST['submitTestimony'])) {
    $sql = 'INSERT INTO testimonies (eventId, felt, dateFelt, timeFelt, `location`, lat, lon, `message`,
    `name`, email, phone)
    VALUES (" . $eventid . "', "' . $_POST['feltRadio'] . "', "' . $_POST['dateFelt'] . "', "' . $_POST
    ['timeFelt'] . "', "' . $_POST['location'] . "', "' . $details[1] . "', "' . $details[2] . "', "' . $_
    POST['message'] . "', "' . $_POST['name'] . "', "' . $_POST['email'] . "', "' . $_POST['phone'] . "')'
    ;
    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
        header("Location: " . $_SERVER['REQUEST_URI']);
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}
```

Skripta 16: Vpisovanje podatkov v podatkovno bazo

Uporabnik lahko lokacijo vnese na več načinov, lahko jo vpiše ročno, poišče na zemljevidu ali pa vnese avtomatsko s pomočjo lokacije naprave. Če izbere avtomatski vnos, bo poklicana funkcija `geoFindMe`, ki je prikazana na Skripti 17. Funkcija izpiše Lat in Lon uporabnika, ki ga pridobi od lokacije naprave (Slika 10). V primeru, da uporabnik ima izklopljeno lokacijo, bo funkcija izpisala napako.

```
function geoFindMe() {  
  const status = document.querySelector('#status');  
  const mapLink = document.querySelector('#map-link');  
  function success(position) {  
    const latitude = position.coords.latitude;  
    const longitude = position.coords.longitude;  
    status.textContent = '';  
    mapLink.textContent = `Latitude: ${latitude} °, Longitude: ${longitude} °`;  
  }  
  function error() {  
    status.textContent = 'Unable to retrieve your location';  
  }  
  if(!navigator.geolocation) {  
    status.textContent = 'Geolocation is not supported by your browser';  
  } else {  
    status.textContent = 'Locating...';  
    navigator.geolocation.getCurrentPosition(success, error);  
  }  
}
```

Skripta 17: Funkcija za iskanje lokacije



Slika 10: Avtomatski vpis lokacije

Za nalaganje slik bomo uporabili obrazec, ki bo ob spremembi poklical funkcijo, ki bo sliko poslala na Imgur API, kot je prikazano na Skripta 18. V skripti je definiran URL do API na katerega bomo sliko poslali in ID odjemalca, ki ga bomo dodali v glavo za preverjanje dostopa. ID dobimo tako, da ustvarimo Imgur račun, brezplačni računi vsebujejo omejitev na število slik, ki jih lahko pošljemo na dan. Funkcija nam vrne povezavo do slike, ki jo bomo zapisali v podatkovno bazo in v obrazcu pokaže predogled slike za uporabnika.

```
var UPLOAD_URL = "https://api.imgur.com/3/image.json";
var CLIENT_ID_HARD = 'CLIENT ID';
let imgPrevDiv = document.querySelector(".img-preview");
let urlPrev = document.querySelector("#uploadedImgUrl");
var injectUrl = function() {
    console.log("[+] Success uploading!");
    let res = JSON.parse(this.responseText);
    imgPrevDiv.setAttribute("src", res.data.link);
    urlPrev.value = res.data.link;
};
var uploadToImgur = function() {
    if ('files' in this && this.files.length > 0)
        upload(this.files[0], injectUrl);
};
@param {File} file //Uploads a single file to imgur
var upload = function(file, cb) {
    urlPrev.value = "";
    // Get client id on change
    var CLIENT_ID = CLIENT_ID_HARD || document.querySelector("#client-id").value;
    if (CLIENT_ID.length < 1) {
        throw new Error("I need a client Id!");
    }
    let fd = new FormData();
    fd.append("image", file, file.name);
    // AJAX Request
    let xhr = new XMLHttpRequest();
    xhr.addEventListener("load", cb);
    xhr.open("POST", UPLOAD_URL);
    // Send authentication headers.
    xhr.setRequestHeader("Authorization", "Client-ID "+CLIENT_ID);
    // Send form data
    xhr.send(fd);
};
```

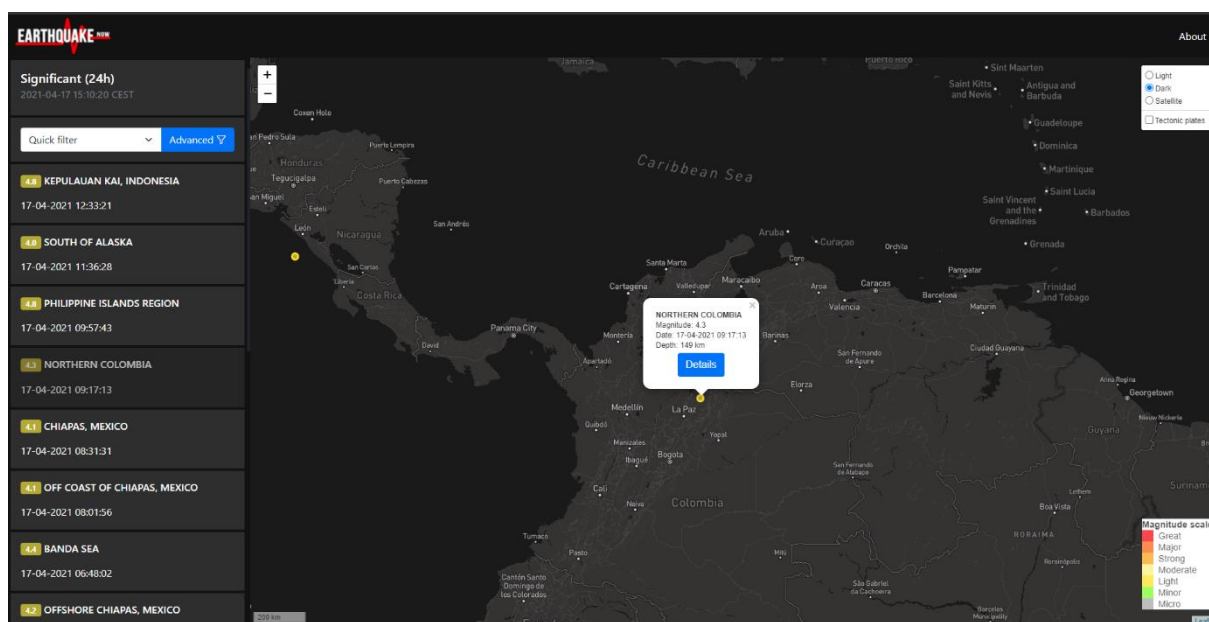
*Skripta 18: Funkcija za pošiljanje slik na Imgur API*

## 6 Predstavitev aplikacije

V tem poglavju je predstavljena uporaba razvite aplikacije s strani uporabnika. Najprej je predstavljena glavna stran aplikacije in uporaba filtrov, nato sledi predstavitev podrobnosti potresov in na koncu še mobilna odzivnost spletne aplikacije.

### 6.1 Glavna stran

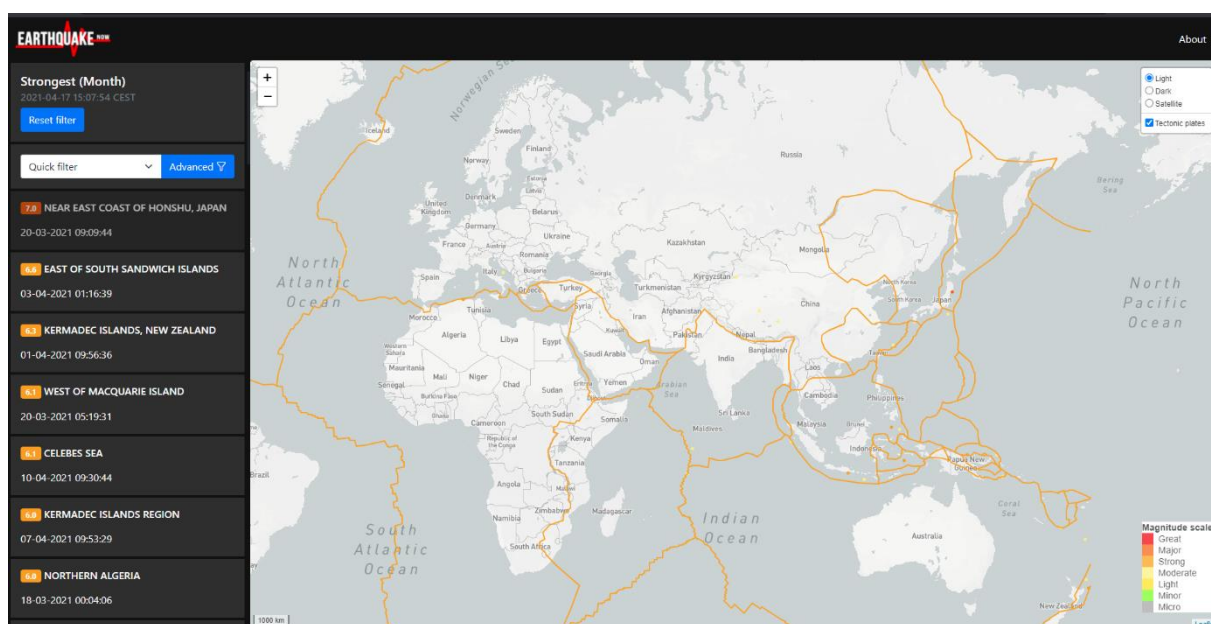
Ko uporabnik odpre glavno stran aplikacije, bo zemljevid prikazal najnovejši potres. Privzeto so v stranski vrstici prikazani vsi potresi z magnitudo več kot 4 v zadnjih 24 ur. Stranska vrstica vsebuje še vrsto trenutnih podatkov, trenutni čas, hitri in napredni filter. Potresi imajo zraven imena značko, ki predstavlja magnitudo in je pobarvana glede na moč. Legenda barv se nahaja v spodnjem desnem kotu, v zgornjem pa lahko uporabnik preklaplja med različnimi sloji zemljevida (svetel, temen, satelitski pogled in tektonske plošče). Navigacijski meni spletne aplikacije vsebuje logotip na levi strani, ki nas ob kliku preusmeri na domačo stran in na desni gumb za podatke o aplikaciji.



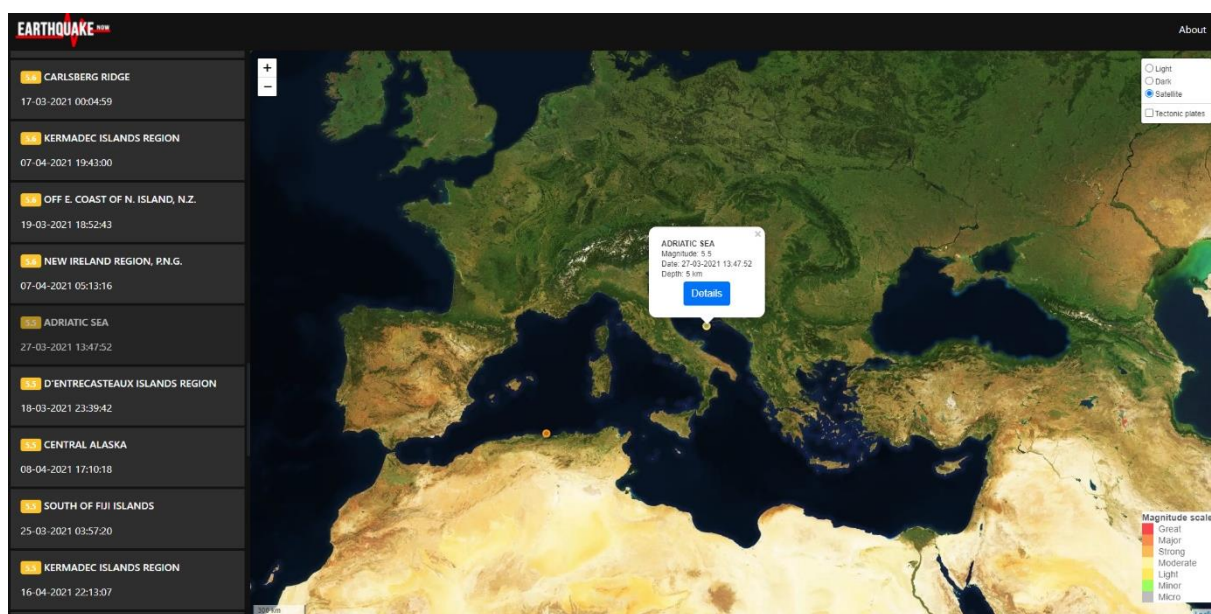
Slika 11: Glavna stran aplikacije



Spletna aplikacija omogoča spreminjanje vrste zemljevida med temnim, svetlim in satelitskim načinom ter vkapljanje pogleda tektonskih plošč. Slika 12 prikazuje posnetek zaslona s svetlim zemljevidom in vklopljenim tektonskim ploščam, Slika 13 pa satelitski zemljevid. Svetel zemljevid je bolj pregleden in uporaben za iskanje in prikazovanje lokacij potresov, medtem ko satelitski ponuja boljši pregled geografskega reliefa in gostote prebivalstva na območju.



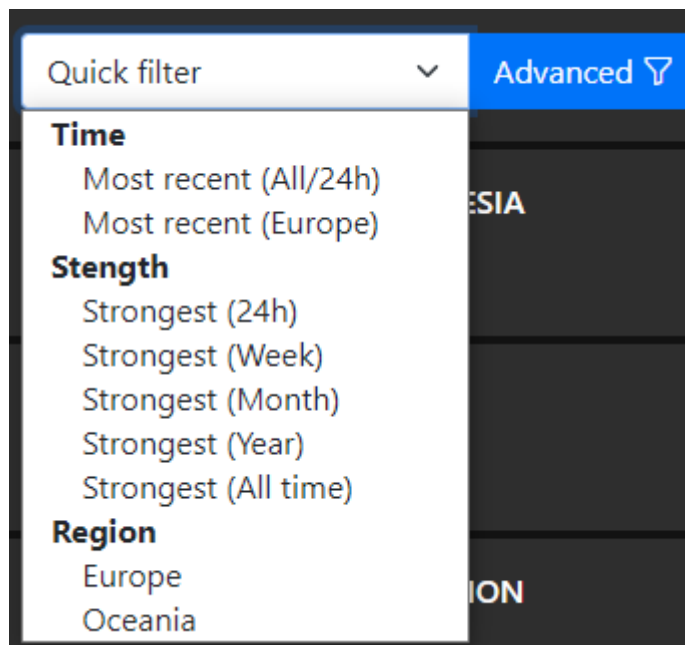
Slika 12: Svetel zemljevid



Slika 13: Satelitski zemljevid

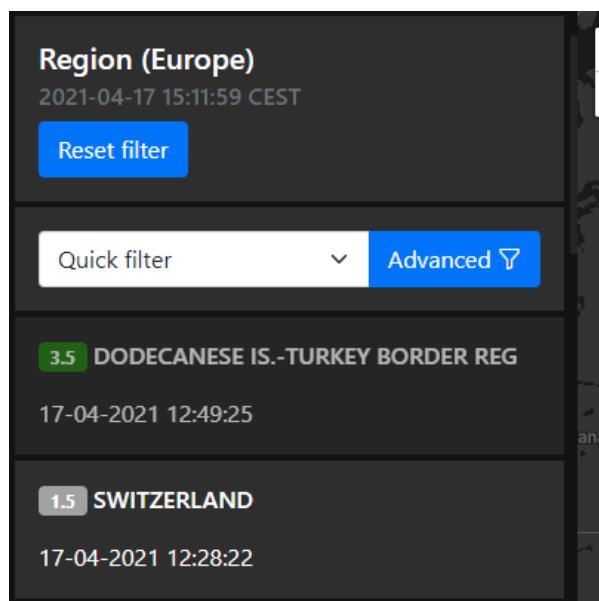


Hitri filter omogoča filtriranje potresov po vnaprej nastavljenih parametrih kot so najnovejši in najmočnejši potresi v določenem časovnem obdobju ali pa po regijah v katerih so se zgodili potresi, kot je razvidno iz Slika 14.



Slika 14: Hitri filter

Če uporabnik izbere npr. Europe filter, se bojo prikazali samo potresi v Evropi. Stranska vrstica dinamično posodobi ime trenutnega filtra in doda gumb za hitro ponastavljanje filtrov, kot je prikazano na Slika 15.



Slika 15: Rezultati hitrega filtra

Za iskanje bolj specifičnih rezultatov, nam to omogoča obrazec za napredno filtriranje. Obrazec omogoča vnos parametrov za iskanje glede na datum, magnitudo, globino in geografske omejitve za latitudo in longitudo, kot je razvidno iz Slika 16. Po iskanju se stranska vrstica dinamično posodobi, da pokaže še parametre, ki smo jih uporabili pri iskanju, kar prikazuje Slika 17.

Advanced search

Start date: 08 Apr 2021

End date: 16 Apr 2021

Minimum magnitude: 1

Maximum magnitude: 5

Magnitude type: mf

Minimum depth: 50

Maximum depth: 100

Order by: magnitude

**Geographic constraints**

**Box area constraints**

Minimum latitude	Maximum longitude	Minimum latitude	Maximum longitude
Default: -90	Default: -90	Default: -180	Default: -180
Degrees: -90.0 - 90.0	Degrees: -90.0 - 90.0	Degrees: -180.0 - 180.0	Degrees: -180.0 - 180.0

**Circuit constraints**

Latitude	Longitude	Minimum radius	Maximum radius
		Default: 0	Default: 180
Degrees: -90.0 - 90.0	Degrees: -180.0 - 180.0	Degrees: 0 - 180.0	Degrees: 0 - 180.0

Close Search

Slika 16: Napredni filter

Results for

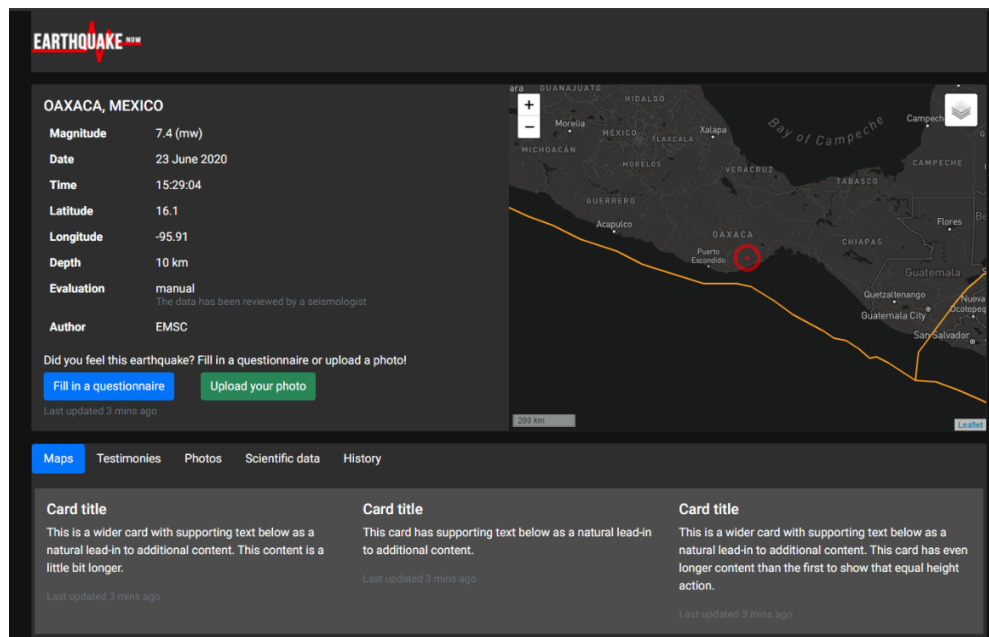
orderby	magnitude
start	2021-03-31
end	2021-04-16
minmag	5

Reset filter

Slika 17: Okno stranske vrstice

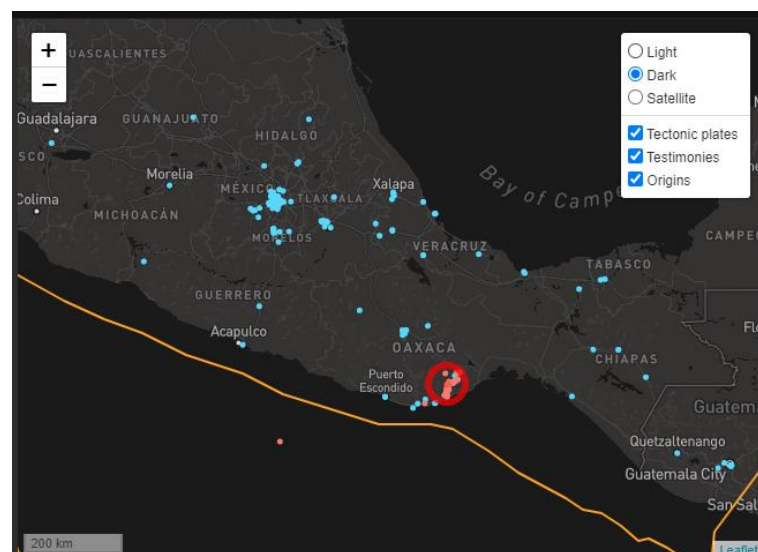
## 6.2 Podrobnosti potresa

Ob kliku na "More details" v pojavnem oknu se nam odpre nova stran, ki vsebuje podrobnosti o potresu, kot je prikazano na Slika 18. Zgornji del prikazuje osnovne podatke o potresu na levi in zemljevid žarišča potresa in drugih podatkov na desni. Spodaj se pa nahaja navigacijski meni za dodatne podatke o potresu.



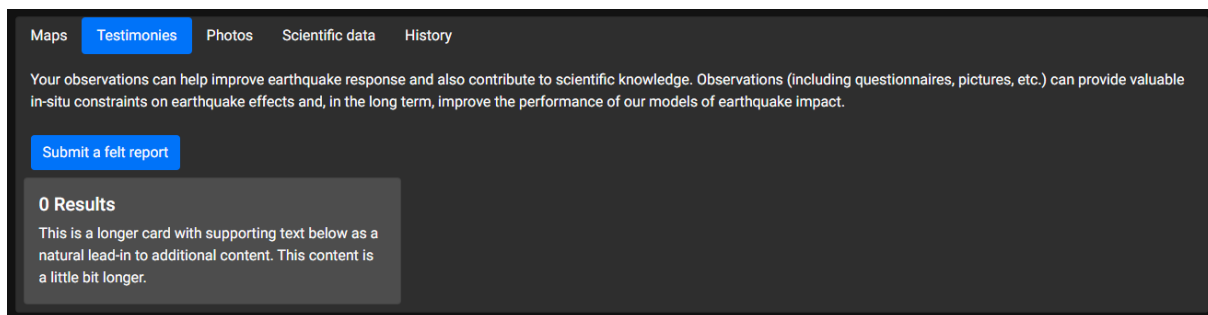
Slika 18: Podrobnosti potresa

Uporabnik lahko izbere vrsto zemljevida in podatke, ki jih želi videti. Modre pike prikazujejo izjave (ang. Testimonies), kjer so uporabniki čutili potres in rdeče prikazujejo žarišča potresa (ang. Origins), kar je razvidno iz Slika 19.



Slika 19: Sloji potresa

V spodnjem navigacijskem meniju lahko preklapljam med različnimi vrstami podatkov, kot so zemljevidi, izjave, slike, znanstveni podatki in zgodovina potresov. Slika 19 prikazuje zavihek za izjave uporabnikov, ki so čutili potres. V kolikor še ni izjav, nam aplikacija pokaže samo privzeto prazno karto.

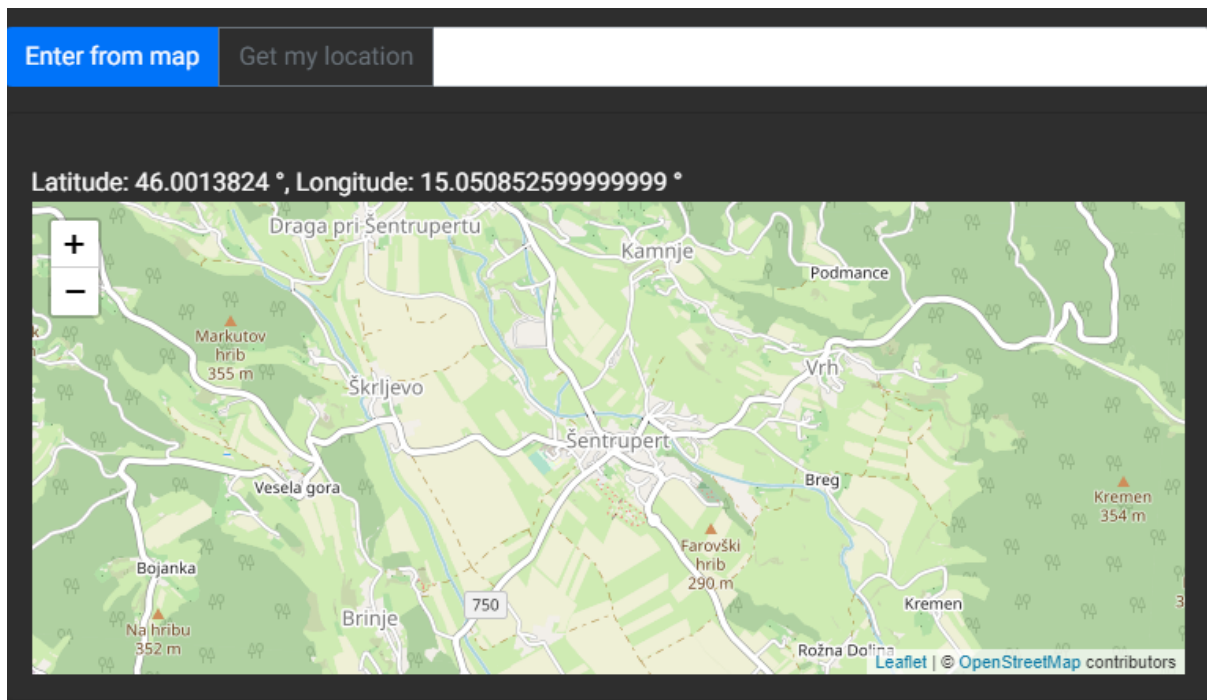


Skripta 19: Izjave brez rezultatov

V primeru, da je uporabnik čutil potres, lahko doda novo izjavo s pomočjo gumba "Submit a felt report", ki odpre pojavno okno z obrazcem za dodajanje izjav, kar prikazuje Slika 20.

Slika 20: Obrazec za izjavo

Lokacijo vnesemo z gumbom "Get my location", ki pridobi lokacijo naprave in nam vrne koordinate, kot je prikazano na Slika 21. Lokacijo tudi pokaže na zemljevidu, v kolikor jo uporabnik želi spremeniti.



Slika 21: Vnašanje lokacije

Če uporabnik želi vnesti še svoje kontaktne podatke, to lahko stori v spustnem meniju za "Optional contact information". Meni vsebuje obrazec za vnos imena, elektronske pošte in telefona, kot je prikazano na Slika 22.

**Optional contact information**

This information is not public and will only be used to contact you if we need your help with the data.

**Name**

Napoleon Bonaparte

**Email**

example@earthquakenow.eu

**Phone number**

Please include the country code

Slika 22: Obrazec za vnos kontaktnih informacij

Če potres že vsebuje izjave od prejšnjih uporabnikov, nam jih bo izpisal v posameznih kartah, kot je prikazano na Slika 23.

Maps
Testimonies
Photos
Scientific data
History

Your observations can help improve earthquake response and also contribute to scientific knowledge. Observations (including questionnaires, pictures, etc.) can provide valuable in-situ constraints on earthquake effects and, in the long term, improve the performance of our models of earthquake impact.

Submit a felt report

1

Date: 2021-03-18 Time: 16:31 CEST

Location: Athens Lat: 15.02 Lon -93.81

Message: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

2

Date: 2021-03-18 Time: 16:31 CEST

Location: Athens Lat: 15.02 Lon -93.81

Message: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Slika 23: Izjave

Zavihek za znanstvene podatke vsebuje tabelo za izvore dogodkov (Slika 24) in tabelo za momentne tenzorje (Slika 25). Vsaka tabela vsebuje podatke o času dogodka, lokaciji, globini, moči, agenciji in vrednosti o natančnosti podatkov.

Date	Latitude	Longitude	Depth	Magnitude	Author	pC	uSC	sE	aG	md	MD
23/06/2020 15:29:04	16.1	-95.91	10	7.4 (mw)	EMSC	326	316	1.48	46	2.94	101.25
23/06/2020 15:29:10	16.19	-95.73	55	7.4 (mw)	AUST	26	26		144	3.46	88.91
23/06/2020 15:29:05	16.13	-95.79	10	7.4 (mw)	ISK	60	60		182	4.54	36.66
23/06/2020 15:29:04	16.0	-95.87	10	7.4 (mw)	SC3	70	70		123	4.58	122.23
23/06/2020 15:29:04	16.01	-95.89	10	7.4 (mw)	SC3	108	108		123	4.56	122.23
23/06/2020 15:29:04	15.9	-95.96	10	7.4 (mw)	SC3	41	41		170	4.59	122.35
23/06/2020 15:29:05	16.17	-95.73	10	7.4 (mw)	ISK	99	98		42	4.55	94.44

Slika 24: Tabela za izvore dogodkov

region	time	lat/lon	mag(type)	ev_depth	mt_source_catalog	mt_region	mt_centroid_time	mt_latitude	mt_longitude	mt_mw	mt_depth	mt_strike_1
OAXACA, MEXICO	2020-06-23 15:29:04.2 UTC	16.1/-95.91	7.4mw	10	USGS	OAXACA, MEXICO	2020-06-23 15:29:27.5 UTC	15.9292	-95.9012	7.4	22	270
OAXACA, MEXICO	2020-06-23 15:29:04.2 UTC	16.1/-95.91	7.4mw	10	GCMT	OAXACA, MEXICO	2020-06-23 15:29:12.8 UTC	16.04	-96.06	7.4	20	270

Slika 25: Tabela za momentne tenzorje

5.0 NEAR S COAST OF PAPUA, INDONESIA	18-03-2021 07:54:25
5.3 KEPULAUAN KAI, INDONESIA	31-01-2021 09:44:41
5.1 NEAR S COAST OF PAPUA, INDONESIA	20-07-2020 06:00:38
5.0 KEPULAUAN KAI, INDONESIA	04-10-2019 13:19:27
5.0 KEPULAUAN ARU REGION, INDONESIA	12-08-2019 13:57:54
5.8 KEPULAUAN KAI, INDONESIA	26-01-2019 08:12:48
5.0 NEAR S COAST OF PAPUA, INDONESIA	17-01-2019 03:35:02
5.4 KEP. TANIMBAR REGION, INDONESIA	03-08-2018 05:42:14
5.0 KEPULAUAN KAI, INDONESIA	05-05-2018 16:22:41

Scientific data   Testimonies   Photos   History   **Seismicity**

### Seismicity in the area

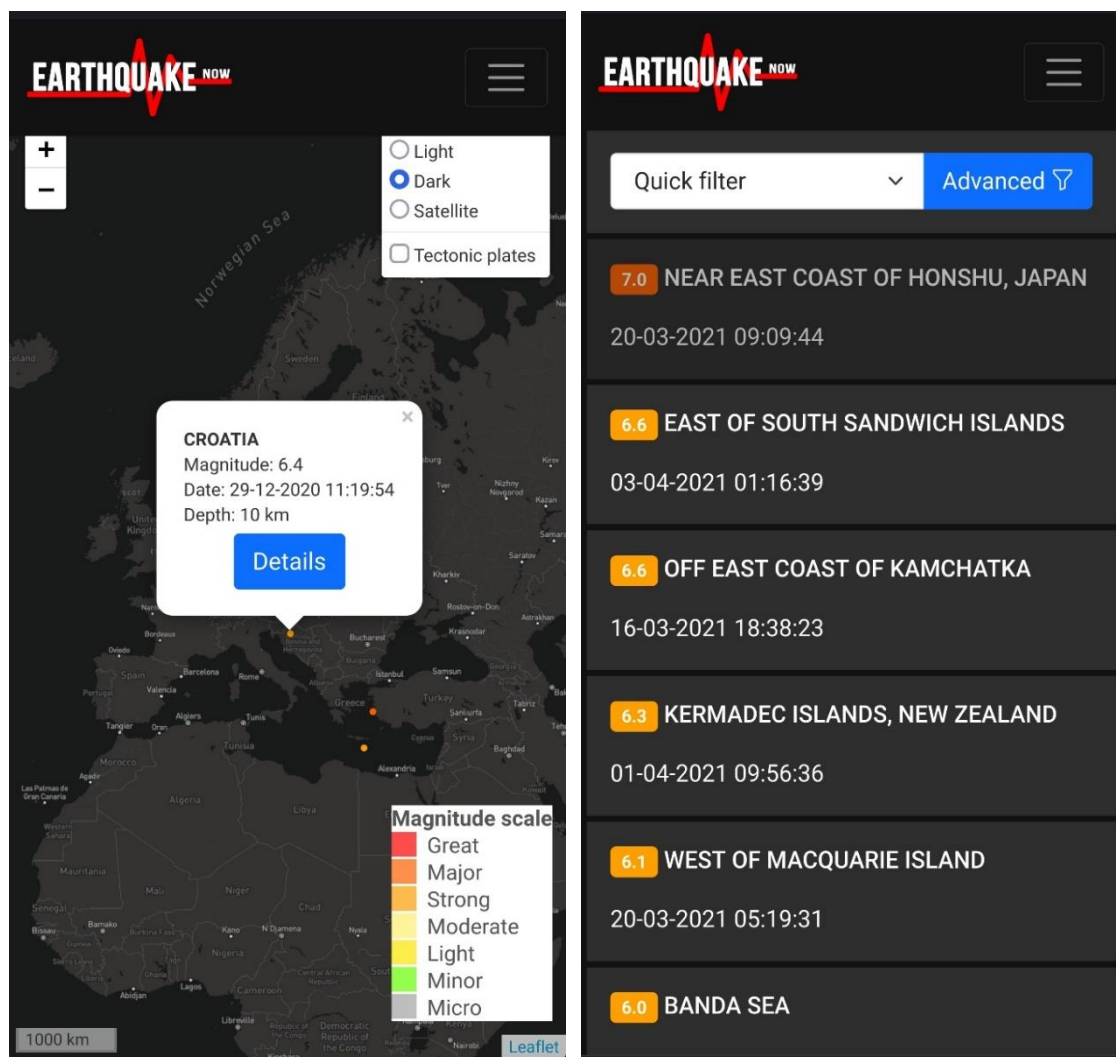
The map displays a high density of green dots, indicating seismic activity, primarily concentrated in the Italian Peninsula and the surrounding regions. The dots are scattered across the Alps, the Apennines, and the coastal areas. The map also shows major cities and geographical features, providing a broader context for the seismic data.

33



### 6.3 Mobilna odzivnost

Celotna spletna aplikacija je tudi popolnoma odzivna glede na velikost zaslona, kar pomeni, da jo lahko uporabljamo tudi na mobilnih telefonih, tablicah, prenosnih računalnikih, itd. Navigacijski meni in stranska vrstica se dinamično skrčita v skriti meni, do katerega lahko dostopamo z gumbom v zgornjem desnem kotu. Meni se tudi avtomatsko skrije ob kliku na potres, za bolj prijazno uporabniško izkušnjo. Slika 28 prikazuje odzivnost spletne aplikacije na Android napravi.



Slika 28: Odzivnost glavne strani in stranske vrstice



Slika 29 prikazuje odzivnost strani za podrobnosti o potresu in obrazca za dodajanje izjav na Android mobilni napravi.

The image shows two side-by-side screenshots of a mobile application interface for earthquakes.

**Left Screenshot: Earthquake Details**

- Header:** EARTHQUAKE NOW
- Title:** DODECANESE ISLANDS, GREECE
- Details:**
  - Magnitude: 7.0 (mw)
  - Date: 30 October 2020
  - Time: 11:51:25
  - Latitude: 37.91
  - Longitude: 26.84
  - Depth: 10 km
  - Evaluation: manual  
The data has been reviewed by a seismologist
  - Author: EMSC
- Map:** A map of the Aegean Sea region with a red circle indicating the earthquake epicenter. Labels include Athens, Aydin, Denizli, Isparta, Antalya, Rhodes, and Mugla. A 100 km scale bar is present.
- Bottom Navigation:** Maps (selected), Testimonies, Photos, Scientific data, History.

**Right Screenshot: Felt report form**

- Header:** Felt report (with a close button 'X')
- Form Fields:**
  - Did you feel it? ☐ Yes ☐ No
  - How strong was the felt earthquake? (A horizontal slider bar with a blue dot in the middle.)
  - Date felt: 10/30/2020
  - Time felt: 11:27 AM
  - Buttons: Enter from map, Get my location
  - Comment: (A large text input area)
  - Optional contact information: (A blue button)
- Bottom Buttons:** Cancel, Submit report

Slika 29: Odzivnost podrobnosti potresa in obrazca za izjave

## 7 Zaključek

V okviru naloge sem razvil dinamično spletno aplikacijo, ki omogoča kartiranje potresov na zemljevidu, filtriranje potresov, vpogled v podrobnosti, dodajanje izjav in slik. Cilj projektne naloge je bil dosežen, ampak so možne številne izboljšave in nadgradnje. Pri izdelovanju aplikacije sem pridobil ogromno znanja s področja obdelave XML dokumentov, REST poizvedb in uporabe Leaflet zemljevidov. Aplikacijo so hitro pobrale in delile številne evropske tiskovne agencije, kar je zelo pripomoglo pri pridobivanju prometa. Aplikacija ima trenutno več kot 4 tisoč uporabnikov, največ iz Argentine, Španije in ZDA.

### 7.1 Nadaljnje delo

Opisana aplikacija predstavlja le eno izmed prvih različic. Celoten model aplikacije je prilagodljiv in omogoča razširitve. Za nadaljnje delo razvite aplikacije bi lahko naredili naslednje nadgradnje oziroma izboljšave:

**Optimizacija** – glavna ovira aplikacije pred produkcijsko izdajo je počasna obdelava podatkov in potreba po obdelovanju istih poizvedb večkrat. To bi lahko rešili z implementacijo predpomnilnika na strani strežnika, ki bi hranil podatke o najbolj pogostih poizvedbah.

**Slike** – trenutna različica aplikacije shranjuje slike s pomočjo Imgur API, ki pa vsebuje omejitve na število slik, ki jih lahko naložimo na dan. Da bi se izognili temu bi lahko shranjevali slike lokalno na strežniku, ampak to prinese še druge izzive in težave.

**Večjezični vmesnik** – aplikacija trenutno podpira samo angleški jezik. Da bi dosegli večje število uporabnikov bi morali postopoma dodati še jezike ostalih evropskih držav, iz katerih bi beležili zadostno število uporabnikov.

**Dodatni podatki** – na spletu je na voljo še ogromno seizmološki podatkov. Lahko bi dodali še posamezne seizmološke postaje, pogled seizmoloških grafov v živo, iskanje po regijah, bolj podrobne izjave, itd.

### 7.2 Zahvala

Zahvaljujem se vsem, ki so mi kakorkoli pomagali pri izdelavi izdelka in dokumentacije, še posebej se zahvaljujem profesorju Robertu Sconocchiniju, ki je samoiniciativno delil spletno stran na svojem blogu in mi s tem pripomogel dobiti promet.

## 8 Literatura

1. **Ministrstvo za obrambo Republike Slovenije.** Ocena ogroženosti Republike Slovenije zaradi potresov. *Uprava Republike Slovenije za zaščito in reševanje*. [Elektronski] 7. June 2018. [Navedeno: 14. April 2021.] [http://www.sos112.si/slo/tdocs/ogrozenost\\_potres.pdf](http://www.sos112.si/slo/tdocs/ogrozenost_potres.pdf). 842-9/2012-73 - DGZR.
2. **Wikipedia.** HTML. *Wikimedia Foundation*. [Elektronski] 9. April 2021. [Navedeno: 9. April 2021.] <https://en.wikipedia.org/w/index.php?title=HTML&oldid=1016919858>.
3. **MDN Web Docs.** CSS: Cascading Style Sheets. *Mozilla*. [Elektronski] February. 10 2021. [Navedeno: 12. April 2021.] <https://developer.mozilla.org/en-US/docs/Web/CSS>.
4. **PHP Group.** *PHP*. [Elektronski] [Navedeno: 12. April 2021.] <https://www.php.net/>.
5. **The PHP Group.** News Archive. *PHP*. [Elektronski] 4. March 2021. [Navedeno: 12. April 2021.] <https://www.php.net/archive/2021.php#2021-03-04-2>.
6. **MDN Web Docs.** About JavaScript. *Mozilla*. [Elektronski] 29. March 2021. [Navedeno: 12. April 2021.] [https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript).
7. **Ecma International.** ECMAScript 2020 Language Specification. *Ecma International*. [Elektronski] 2020. [Navedeno: 12. April 2021.] <https://262.ecma-international.org/11.0/#sec-copyright-and-software-license>.
8. **Wikipedia.** Bootstrap. *Wikimedia Foundation*. [Elektronski] 9. April 2021. [Navedeno: 12. April 2021.] [https://en.wikipedia.org/w/index.php?title=Bootstrap\\_\(front-end\\_framework\)&oldid=1016854982](https://en.wikipedia.org/w/index.php?title=Bootstrap_(front-end_framework)&oldid=1016854982).
9. **Otto, Mark.** Blog. *Bootstrap*. [Elektronski] 23. March 2021. [Navedeno: 12. April 2021.] <https://blog.getbootstrap.com/2021/03/23/bootstrap-5-beta-3/>.
10. **Agafonkin, Vladimir**. *Leaflet.js*. [Elektronski] [Navedeno: 12. April 2021.] <https://leafletjs.com/>.
11. **Ortega, Iván Sánchez.** Leaflet 1.7 released. *Leaflet.js*. [Elektronski] 4. September 2020. [Navedeno: 12. April 2021.] <https://leafletjs.com/2020/09/04/leaflet-1.7.1.html>.
12. **Wikipedia.** MySQL. *Wikimedia Foundation*. [Elektronski] 9. April 2021. [Navedeno: 12. April 2021.] <https://en.wikipedia.org/w/index.php?title=MySQL&oldid=1016928480>.
13. **MySQL.** Release notes. *MySQL*. [Elektronski] 18. January 2021. [Navedeno: 12. April 2021.] <https://dev.mysql.com/doc/relnotes/mysql/8.0/en/news-8-0-23.html>.

14. **MDN Web Docs**. XML introduction. *Mozilla*. [Elektronski] 17. March 2021. [Navedeno: 12. April 2021.] [https://developer.mozilla.org/en-US/docs/Web/XML/XML\\_introduction](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction).

15. **Richardson, Leonard in Ruby, Sam**. *RESTful Web Services*. s.l. : O'Reilly Media, Inc., 2007. 9780596529260.

16. **VMware**. About. *Apache Friends*. [Elektronski] [Navedeno: 12. April 2021.] <https://www.apachefriends.org/about.html>.

17. **phpMyAdmin**. About. *phpMyAdmin*. [Elektronski] [Navedeno: 12. April 2021.] <https://www.phpmyadmin.net/>.

Spletna aplikacija se nahaja na: <https://earthquakenow.cf/>

Izvorna koda je dostopna na: <https://github.com/adnansmlatic/earthquake-now>