

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR

Cartactual folyóirat térképi adatbázis szerkezetének és webes megjelenítő felületének létrehozása

DIPLOMAMUNKA
TÉRKÉPÉSZ MESTERSZAK

Készítette:
Magyar Zoltán

Témavezető:
Dr. Gede Mátyás
egyetemi docens
ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2018



EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR
TÉRKÉPTUDOMÁNYI ÉS GEOINFORMATIKAI TANSZÉK

DIPLOMAMUNKA-TÉMA BEJELENTŐ

Név: **Magyar Zoltán**

Neptun kód: **DZM5YL**

Szak: **térképész MSc**

Témavezető neve: **Dr. Gede Mátyás**

munkahelyének neve és címe: **ELTE IK Térképtudományi és Geoinformatikai Tanszék**
beosztása és iskolai végzettsége: **egyetemi docens, PhD**

A dolgozat címe: **Cartactual folyóirat térképi adatbázis szerkezetének és webes megjelenítő felületének létrehozása**

A dolgozat témája:

Cél a Cartactualban megjelent térképekből – reprezentatív minta alapján – egy megfelelő adatbázis megalkotása, valamint egy interaktív, megjelenítő weboldal létrehozása, aminek az adminisztrációs felülete biztosítja az adatbázis online bővíthetőségét.

A témavezetést vállalom.

.....
(a témavezető aláírása)

Kérem a diplomamunka témájának jóváhagyását.

Budapest, 2018. december 1.

.....
(a hallgató aláírása)

A diplomamunka-témát az Informatikai Kar jóváhagyta.

Budapest, 2018. december 1.

.....
(témát engedélyező tanszék vezetője)

Tartalomjegyzék

1. Bevezetés	2
2. Cartactual	3
3. Tervezés	4
3.1. Tervezett funkciók.....	4
3.2. Adatbázis tervezés	4
3.3. Weboldal-szerkezet tervezés.....	5
3.3.1. Keresés, találat lista	5
3.3.2. Megjelenítés – OpenLayers API	5
3.3.3. Regisztráció, bejelentkezés.....	6
3.3.4. Georeferálás, adatbázisba feltöltés.....	6
4. Programozási környezet.....	6
4.1. PHP	6
4.2. JavaScript	8
4.3. HTML, CSS	10
5. Megvalósítás	10
5.1. Weboldal szerkezete	10
5.2. Adatbázis.....	12
5.3. Regisztáció.....	14
5.4. Bejelentkezés.....	15
5.5. Georeferálás.....	16
5.6. Térkép feltöltése az adatbázisba	19
5.7. Főoldal	21
5.7.1. Keresés	21
5.7.2. Keresés eredménye.....	23
5.7.3. Letöltés	24
6. Tesztelés, bővítési lehetőségek.....	25
7. Összegzés.....	26
8. Irodalomjegyzék.....	27
9. Köszönetnyilvánítás	28
10. Melléklet.....	29

1. Bevezetés

A magyar térképészet nemzetközileg egyik legjelentősebb terméke a *Cartactual* folyóirat volt. A világon egyedülálló módon tematikus térképeket közölt a Földön bárhol bekövetkező térképi tartalmat módosító eseményekről. 1965—1993 között a korszakhoz képest igen gyorsan tudta megjeleníteni ezeket. Fennállása során több ezer térkép készült, többnyire a nyugati országokat érintő változásokról. Értékét nagyban növeli a tudat, hogy mindezt akkor a kommunista blokkba tartozó Magyarországról végezték.

Napjainkban egy ilyen jellegű folyóirat minden bizonnyal nem lenne képes nyereséget termelni. Az internet fejlődésével és a távérzékelés nyújtotta minden eddiginél pontosabb globális térképezés megvalósulásával az analóg információszerzés és folyóirat terjesztés már nem veheti fel a versenyt.

Dolgozatom céljának egy webes térképkatalógus és megjelenítő rendszer létrehozását, tűztem ki. Kritérium volt, hogy az oldalon lehessen georeferálni a feltöltött térképet és az adatbázisban kereséskor az megjelenjen egy online térképfelületen. A katalógus alanyának a *Cartactualt* választottam, mert méltó lenne ehhez a jelentős folyóirathoz, hogy térképei egy helyen, a pontos földrajzi helyzetükben böngészhetők és az utókor számára megőrizhetők legyenek.

2. Cartactual

A Cartactual a Kartográfia Vállalat szakfolyóirata volt 1965 és 1993 között. A világon ez volt az egyetlen folyóirat, amely az egész Földre kiterjedően közölte térképes formában a térképi tartalmat módosító változásokat. Tematikájának köszönhetően a magyar térképészet egyik legismertebb termékévé vált külföldön (*Klinghammer & Papp-Váry, 1983*).

A folyóirat elődének tekinthető a Radó Sándor által alapított 1957—1988 között működő Terra térképszolgálat, mely főként sajtótérképeket állított elő az aktuális világpolitikai eseményekről. A kezdeti sikeres eredmények hatására Radó kezdeményezésével 1965-ben megalapították a Cartactualt, a következő alapelvekkel:

- A térképek egységes kivitelűek A-3-as méretű lapokon, amiket puha kötésben kell kiadni 12 fekete-fehér, egyoldalas nyomtatásban.
- A térképek témája kizárólag földrajzi jellegű, témájának pártatlannak kell lennie.
- Közölni kell a Földön bárhol bekövetkezett földrajzi változásokat.
- Olyan objektumokat (vonalas és felületi egyaránt) kell bemutatni, amelyek kis- és közepes méretarányú térképeken még nem jelentek meg (pl.: nemzetközi-, közigazgatási határok változása, új utak, légifolyosók, vezetékek, vasutak, hidak stb.)
- Pontszerű adatokat is közölni kell, mint például népszámlálási eredmények, ipari termelési adatok (pl.: erőművek, bányák, olajfinomítók)
- A térképeket a könnyebb kereshetőség érdekében egyedi számmal kell ellátni.

Negyedéves periodikaként indult angol, francia és német nyelven, azonban 1968-ban a magyarral kiegészülve négynyelvűvé vált. 1966-tól a fekete-fehér térképen a tematikát vörös színnel emelték ki. A kezdeti évek során kialakult egy újabb alapelv, miszerint csak első kézből származó információkat közölhet a Cartactual (*Gercsák, Márton & Tóth, 2016*).

Radónak köszönhető a folyóirathoz a Földről mindenhonnan érkező információk sokasága, amit a két világháború közötti térképészeti tevékenysége és kialakított kapcsolatai alapoztak meg. Eleinte az alapanyag a Terra sajtótérkép-szolgálat adataiból származott. Később, egészen a folyóirat megszüntetéséig, kiterjedt levelezési hálózat és kialakított cserekapcsolatok adták az alapanyagot. Ezek jelentőségét mutatja, hogy a szerkesztőség az évek folyamán 1100 intézménnyel tartotta a kapcsolatot és adattárában szerepelt nemzetközi szaknévsorozat, állami hivatalok címjegyzéke és egyéb intézményi adatok, továbbá évente átlagosan 600 db információ bekérő levél került postázásra. (*Gercsák, Márton & Tóth, 2016*).

1965—1969 között negyedévenként jelent meg számonként tizenkettő térképlappal. 1970-től a kiterjedt kapcsolatoknak köszönhetően beérkező, hatalmas mennyiségű információ hatására kéthavi megjelenésűvé vált és már tizennégy térképlapot tartalmazott. Ezeken minden esetben feltüntették, hogy melyik intézmény volt az adatok forrása. „A Cartactual 1983-ban elnyerte a Nemzetközi Térképészeti Társulás (ICA) erkölcsi támogatását: engedélyezték, hogy a Cartactual a fedőlapján feltüntesse az ICA logóját, illetve jelezze, hogy a folyóirat az ICA védnöksége alatt jelenik meg.” (Gercsák, Márton & Tóth, 2016)

A gazdasági helyzet romlásával, a színvonal megtartása érdekében 1992-től évente csak négy alkalommal jelent meg. A Kartográfia Vállalat jogutódja 1993-ban megszüntette a Cartactualt, ami huszonkilenc éves pályafutása alatt 158 számot élt meg.

3. Tervezés

3.1. Tervezett funkciók

A webes alkalmazás fejlesztése előtt szükséges volt meghatározni a rendszerrel szemben elvárt teljesítményt, kitalálni az oldal szerkezetét és arculatát, továbbá megállapítani a tervezett funkciókat. Mivel a munka során a Cartactual ún. térképkatalógusa és ennek megjelenítő felülete kerül kialakításra, ezért nélkülözhetetlen, hogy az térképek keresetők legyenek a megfelelő mezők kitöltésével. Az eredmények megjelenítésének kétféle módja szükséges: táblázatos formában és online térképen. Biztosítandó a beszkenelt térképek, és a hozzájuk tartozó földrajzi pozíciót tartalmazó szöveges fájl letölthetősége. A weboldalon keresztüli adatbázis bővítési lehetőség érdekében kialakítandó egy georeferáló/feltöltő felület, azonban nem célszerű ennek a teljeskörű hozzáférhetősége. Ezt egy létrehozandó regisztráló/bejelentkező rendszernek kell szolgáltatnia. Továbbá cél a könnyen kezelhetőség.

3.2. Adatbázis tervezés

Adatbázis alatt az adatok rendezett formában tárolt gyűjteményét értjük. Egy adatbázis táblákból épül fel, léteznek egy- és több táblás, illetve relációs adatbázisok. Egy tábla sorait rekordoknak, oszlopait a tábla mezőinek hívjuk. A rekordok és mezők keresztszétét attribútumnak hívjuk. Az oszlopokat attribútum osztálynak is nevezhetjük, mert a mező típusa definiálja az adott mezőben tárolható attribútumok típusát (pl.: character, string, int, float, boolean stb.) (Elek, 2011).

Az adatbázis tervezésénél fő szempont volt a könnyű kezelhetőség és az egyszerűsége való törekvés. Legalább kettő táblára van szükség a weboldal megfelelő működéséhez: egy a felhasználók- és egy a térképek adatainak tárolására.

Adatbázis-kezelő szervernek a MySQL-t választottam, ami egy többfelhasználós, többszálú, SQL-alapú adatbázis-rendszer. Az SQL a Structured Query Language kifejezés mozaikszava, amely strukturált lekérdező nyelvet jelent. Ennek használatával végezhető adatbázisok, táblák létrehozása, lekérdezése.

Az adatbázisomat és tábláit phpMyAdmin felületen hoztam létre, ami jelenleg az egyik legnépszerűbb nyílt forráskódú, platformfüggetlen, PHP nyelven írt MySQL adatbázis-kezelő rendszer.

3.3. Weboldal-szerkezet tervezés

3.3.1. Keresés, találat lista

Az attribútumtábla kereshetőségét az alapkoncepció alapján egy HTML form biztosítja. Ez több input mezőt tartalmaz, melyek kitöltésével lehet szűrni a kívánt térképekre. A keresés folyamán cél, hogy ne legyen kötelező az összes mezőt kitölteni. Az üresen maradtak a kategóriájukban az összes elemre vonatkoznak, amennyiben van kitöltött mező, akkor azok is a keresés részeivé válnak. Így megvalósítható, hogy csak a kitöltött feltételeknek megfelelő rekordok kerülnek kiválasztásra.

Az eredményeket egy dinamikusan változó táblázatban és egy online térképen kell megjelenítenie. A táblázat sorai a MySQL tábla publikálásra szánt mezőit tartalmazzák, valamint egy gombot, ami a térképi állományok letöltésére szolgál. A kurzort a táblázat egy rekordja fölé mozgatva, az OpenLayers API felület az adott térképet határoló poligonra közelít, annak stílusa megváltozik, továbbá kattintásra betölti a térképet.

3.3.2. Megjelenítés – OpenLayers API

A OpenLayers egy nyílt forráskódú kliens oldali JavaScript függvénykönyvtár. Használatával többféle forrásból származó (pl.: KML, GeoJSON, WMS/WFS, raszteres stb.) térképek, térképi adatok vizualizálhatók meg. Az OpenLayers felületen az oldal betöltésekor az Open Street Map alaptérkép jelenik meg. A korábban említett keresés eredményessége esetén megjelennek a találatokhoz tartozó térképek poligonjai.

3.3.3. Regisztráció, bejelentkezés

A főoldal tetején a navigációs sávban helyet kapnak a bejelentkezéshez szükséges input mezők és gomb, valamint egy gomb a regisztráció elindításához. Az utóbbira kattintva a weboldal átirányítja felhasználót a regisztrációs felületre, ahol a felhasználónév, az email-cím és a jelszó kétszeri megadása után sikeres a regisztráció amennyiben a felhasználónév még nem foglalt. Bejelentkezés után a navigációs sávban megjelenik a georeferáló felületre irányító és a kijelentkezéshez szükséges gomb.

3.3.4. Georeferálás, adatbázisba feltöltés

Bejelentkezés után a felhasználónak lehetősége nyílik új térkép georeferálására és adatbázisba feltöltésére. Ennek a felületnek a nagy részét kettő OpenLayers API elem alkotja, egy a feltöltendő térkép, egy a referencia térkép megjelenítésére. Ezeken, kattintásra a felhasználónak képesnek kell lennie illesztőpontok elhelyezésére. Megfelelő mennyiség után elvégezhető a georeferálás. Egy, a térkép adatait tartalmazó form kitöltése után a georeferált térkép és a hozzá tartozó adatok feltölthetők az adatbázisba.

4. Programozási környezet

4.1. PHP

A PHP az első scriptnyelvek egyike, kezdetben még csak egy script gyűjtemény volt személyes honlapok karbantartására. Innen ered a neve, azaz Personal Home Page Tools (Személyes honlap eszközök). Később a PHP 3. verziójának kiadásával teljesen átalakult a nyelv és a mozaikszó jelentése PHP: Hypertext Preprocessor (PHP: hipertext-feldolgozó) lett. Az újabb kiadásokkal a nyelv már képes nagyméretű webes adatbázisalapú alkalmazások működtetésére.

A PHP mára széles körben elterjedt, nyílt forráskódú, szerver oldali programozási nyelv. Kifejezetten alkalmas dinamikus, adatbázissal kapcsolatban álló weboldalak fejlesztésére könnyű HTML-be ágyazhatósági képességével. A PHP alapú weboldalak elkészítésénél először a PHP kód fut le, alkalmat biztosítva arra, hogy a HTML tartalma még a szerver oldalon módosuljon, mielőtt az a böngészőbe kerülne. Ennek köszönhetően a kliens oldalról nem látható a programkód, csak az elkészült HTML. A nyelv előnye, hogy alkalmas olyan feladatok végrehajtására, amiket a kliensoldali scriptek nem vagy korlátozottan tudnak elvégezni, mint például bejelentkezés, adatbáziskezelés, fájlkezelés, kódolás stb. (Nixon, 2014).

Az alkalmazáshoz a böngésző felől elsősorban POST vagy GET módszerrel küldött, valamint sütikben tárolt adatok érkehetnek. Ezeket célszerű megbízhatatlan forrásnak

tekinteni, mert a kliens oldali ellenőrzéseket a felhasználó akármikor módosíthatja, törölheti. A beérkező adatokat végrehajtás előtt célszerű a szerveroldalon ellenőrizni, mert erre a kliensnek nincs rálátása.

A PHP hátránya, hogy nem megfelelően megírva nem tekinthető biztonságosnak. Adatbázis alapú weboldalak esetében például biztonsági kockázat az SQL injection (SQL injekció), mely során a lekérdezésbe ágyazott SQL parancsokkal feltérképezhető az adatbázis szerkezete, tartalma, ezek megsemmisíthetővé válnak stb. Ez a támadási forma könnyen elhárítható. A PHP MySQLi bővítmény felhasználása során ún. előkészített utasítással (prepared statement) biztosítható, hogy a kliens felől érkező adatokat a szerver megfelelő formában értelmezze (1.ábra), ezek lehetnek:

- i: integer (egész szám)
- d: double (lebegőpontos szám)
- s: string (karakter sorozat)
- b: boolean (logikai érték)

```

                                                                    dbh.inc.php
...
$conn = mysqli_connect($dbServername, $dbUsername, $dbPassword, $dbName);
...

                                                                    upload.inc.php
...
require 'dbh.inc.php';
...
$sql = "SELECT mapid FROM maps WHERE mapid=?";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt, $sql)) {
    echo 'SQL_ERROR';
    exit();
} else {
    mysqli_stmt_bind_param($stmt, "s", $mapid);
    mysqli_stmt_execute($stmt);
    ...
    mysqli_stmt_close($stmt);
    ...
}
```

1. ábra: MySQLi prepared statement példa

4.2. JavaScript

A JavaScript (JS) nevéből jól látható, hogy ez egy script nyelv, azaz futás közben értelmezett nyelv. Kis erőforrás-igényű, értelmezett programozási nyelv, aminek legelterjedtebb felhasználási területe a weblapok dinamikussá, interaktívvá tétele. A JS fájlok a többi HTML dokumentummal együtt töltődik be a böngészőbe és kliens oldalon fut, ahol a böngésző JavaScript-motorja értelmezi és futtatja a kódot. A nyelv számára a HTML dokumentumot a DOM (Document Object Model – Dokumentum Objektum Modell) reprezentálja, amiben minden egyes HTML elem, egy hierarchikus rendszerbe szerveződött objektumnak felel meg. Így futás közben a DOM módosíthatók, ezáltal a weboldal dinamikusan megváltoztatható. Eseménykezelői többféleképp és egyszerűen beágyazhatók a HTML dokumentumba (2. ábra).

```

                                                                    georef.php
...
<button onclick="addPoint()">Add point</button>
...
<script src="georef.js"></script>
...
                                                                    georef.js
...
window.addEventListener('load', function() {
    document.querySelector('input[type="file"]').addEventListener('change',
function() {
    ...
    img.onload = imageIsLoaded;
    ...
});
function imageIsLoaded() { ...};
...

```

2. ábra: Eseménykezelők a HTML-ben, illetve eseménykezelők objektumokhoz rendelése JavaScript-ben

A JavaScript, mint látható egy kliens oldali programozási nyelv, a weboldalakkal történő interakcióban és azok módosításában kiválóan működik. Bár képes XMLHttpRequest függvénnyel kapcsolatot teremteni, de léteznek a nyelvnek kiegészítői, melyekkel kényelmesebben lehet elvégezni a műveletet. Ilyen például a jQuery JS függvénykönyvtár, ami azért készült, hogy egyszerűsítse a DOM objektumok azonosítását, manipulációját, megkönnyítse az eseménykezelést stb. (3. ábra) A jQuery továbbá hangsúlyozza a szerver-kliens összekapcsolást AJAX alkalmazások létrehozásának támogatásával. Ekkor a felhasználó interakcióba tud lépni a szerverrel kérések és utasítások küldésén keresztül, ami ezeket feldolgozza és választ küld vissza a böngészőnek megjelenítésre.

Az Ajax nem egy programozási nyelv, hanem egy interaktív alkalmazások létrehozására szolgáló webfejlesztési technika. Neve az Asynchronous JavaScript and XML rövidítése. Használatával a webalkalmazások képesek a háttérben kis mennyiségű adatot gyorsan cserélni a szerverrel, így az oldalnak nem kell minden egyes alkalommal újratölteni, amikor a felhasználó módosít valamit (Garrett, 2005). Az Ajax a következő technikák kombinációja:

- XHTML (vagy HTML) és CSS: Tartalom leírása és formázása.
- JavaScript, DOM: A kliens oldali script nyelvek alkalmazásával az interaktív megjelenítés és módosítás lehetővé tétele
- XML: Ez a formátum használt legtöbbször, de nem kizárólagosan a kliens és a server között. A kommunikáció történhet formázott HTML-ként vagy sima szöveggént is. Alternatívaként egyre elterjedtebb a JSON (JavaScript Object Notation – JS Objektum Jelölés).
- XMLHttpRequest: Ez az objektum alkalmas az adatok aszinkron kezelésére a szerver és a kliens között. Használatával az oldal újratöltése nélkül lehet tartalmat frissíteni

```
...
$.ajax({
  type: "POST",
  enctype: 'multipart/form-data',
  url: url,      // URL of the destination php file
  ...
  success: function (data) {
    console.log("SUCCESS : ", data);
    $("#bttmOK").prop("disabled", false);
  },
  error: function (e) {
    console.log("ERROR : ", e);
    $("#bttmOK").prop("disabled", false);
  }
});
...
```

georef.js

3. ábra: Példa az Ajax alkalmazására jQuery használatával

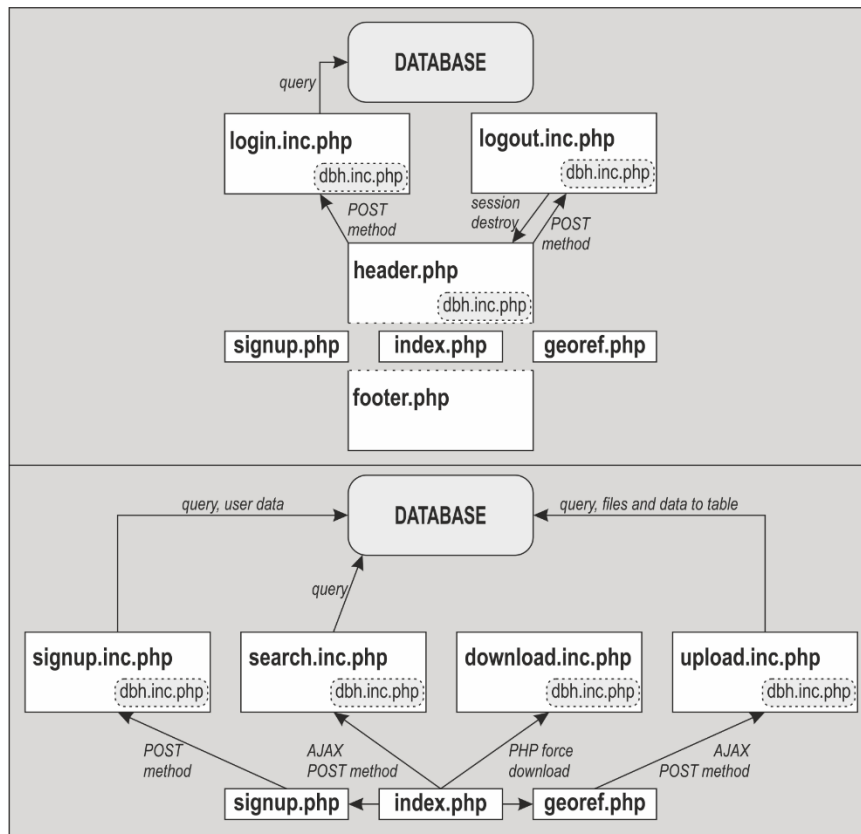
4.3. HTML, CSS

A HTML az angol HyperText Markup Language angol kifejezés rövidítése, jelentése magyarul: hiperszöveges leíró nyelv. A hiperszöveg biztosítja a dokumentum egyes elemei és más dokumentumok közötti kapcsolatot. Fontos megemlíteni, hogy a HTML nem egy programozási nyelv és megjelenítési képességei korlátozottak. A felhasználói felület megjelenésének definiálásához egy egyszerű módszer a CSS használata. Ez a Cascading Style Sheets mozaikszava, magyar jelentése egymásba ágyazott stíluslapok. A CSS egy stíusleíró nyelv és általa stíluslapok készíthetők és ágyazhatók a HTML-be. Ezekben a fájlokban a weblaphoz teljes körű stílusbeállításokat lehet megadni. Beállíthatók a méretek, határok, térközök, animációk, betűtípusok, pozíciók és még sok minden (Nixon, 2014).

5. Megvalósítás

5.1. Weboldal szerkezete

A weboldal fejlesztése során elsődleges szempont volt, hogy az adatbázissal közvetlen kapcsolatot létrehozó programkódok külön fájlokba kerüljenek és ezek a főoldalhoz képest egy alkönyvtárban helyezkedjenek el. Ezek közül szinte az összes egy egész fájlban átívelő `if else` elágazáson belül található, ami meggátolja a felhasználót, hogy URL-en keresztül elérje tartalmukat. Szintén cél volt a programozás során, hogy a felhasználói felületen a *header* navigációs sáv és a *footer* állandóan elérhető legyen. A *footer* tartalma dinamikusan változik, számon tartja a weboldal létrehozásának az évét és az aktuális évszámot. Ebből következik, hogy a HTML kód három részből alakul ki a szerveren, de a kliens csak egy egészet lát. Az oldal fájljainak egymáshoz képesti viszonyrendszere a korábbi feltételek alapján a következőképp alakult ki:



4. ábra: A weboldal fájljainak viszonyrendszere

A felhasználói felület arculatának kialakításakor az egyszerűsége, a letisztultságra és a könnyen kezelhetőségre törekedtem. A weblap megnyitásakor a főoldal jelenik meg, ahol alapesetben a keresés, letöltés és regisztráció funkciók érhetők el. Bejelentkezés után a felhasználó jogosultságot kap a georeferáló felület megnyitására, ezáltal térképek georeferálására és adatbázisba feltöltésére.

5.2. Adatbázis

Az adatbázist a rendelkezésre álló Cartactual példányok térképei alapján SQL parancsokkal alkottam meg phpMyadmin felületen keresztül (5.ábra). Az adatbázis a *cartactual* nevet kapta és utf8_hungarian_ci karakter-kódolású, továbbá kettő táblát tartalmaz egyet a térképek adatai, egyet a felhasználók adatai tárolására.

```
CREATE TABLE users (  
  idUsers int(11) AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  uidUsers TINYTEXT NOT NULL,  
  emailUsers TINYTEXT NOT NULL,  
  pwdUsers LONGTEXT NOT NULL  
);  
CREATE TABLE maps (  
  id INT(11) AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  filename TINYTEXT NOT NULL,  
  nbr INT NOT NULL,  
  vol INT NOT NULL,  
  yr INT NOT NULL,  
  mapid TINYTEXT NOT NULL,  
  ttl TEXT NOT NULL,  
  ttlhu TEXT NOT NULL,  
  ctry TEXT NOT NULL,  
  ctryhu TEXT NOT NULL,  
  ol TINYTEXT NOT NULL,  
  g GEOMETRY NOT NULL  
);
```

5. ábra: Az adatbázis tábláinak létrehozása SQL parancsokkal

A regisztrált felhasználók adatainak tárolására a *users* nevű táblát hoztam létre négy mezővel:

- *idUsers*: Integer típusú, elsődleges kulcs- és automatikus növekedésű.
- *uidUsers*: String mező a felhasználónevek tárolására.
- *emailUsers*: String mező az email címek tárolására.
- *pwdUsers*: String mező a jelszavak tárolására.

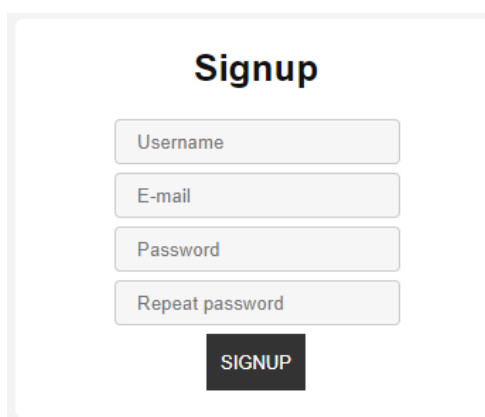
A szkennelt térképek *maps* nevű táblájához 12 attribútum osztályt definiáltam:

- Integer típusú mezők:
 - *id*: Elsődleges kulcs, automatikus növekedésű.
 - *nbr*: A térképet tartalmazó folyóirat száma.
 - *vol*: A térképet tartalmazó folyóirat évfolyama.
 - *yr*: A térkép készítésének éve.
- String típusú mezők:
 - *filename*: Az adatbázisba feltöltött fájl neve kiterjesztéssel együtt.
 - *mapid*: A térkép egyedi azonosítója. A folyóiratban az összes térképet egyedi azonosító kóddal láttak el.
 - *ttl*: A térkép angol címe.
 - *ttlhu*: A térkép magyar címe.
 - *ctry*: Angol országnév.
 - *ctryhu*: Magyar országnév.
 - *ol*: Térképhez tartozó adatok az OpenLayers API-ban megjelenítéshez.
- Geometry típusú mező:
 - *g*: A georeferált térképek által lefedett területet tartalmazó poligon. Keresésnél ebből a mezőből generálódik a GeoJSON réteg, ami az OpenLayers API-ban jelenik meg.

A webes alkalmazás működése során az adatbázissal interakcióba kerülő összes php fájl meghívja a *dbh.inc.php* csatlakozási adatokat tartalmazó dokumentumot, csatlakozik az adatbázishoz és a szerverrel történő kommunikáció után a `mysqli_stmt_close()` és `mysqli_close()` függvényekkel zárja a kapcsolatot. Továbbá az összes SQL parancs prepared statement alkalmazással kerül végrehajtásra.

5.3. Regisztráció

A **SIGNUP** gomb kattintás eseménye átirányítja a felhasználót a *signup.php* oldalra, ami egy regisztrációs űrlapot tartalmaz (6. ábra). Az ebben a form-ban található **SIGNUP** gombra kattintva a beviteli mezők adatait egy POST metódussal a *signup.inc.php* kapja meg. A beviteli értékek formájának vizsgálatát egymás utáni `if else` elágazások végzik. Először ellenőrzi a program, hogy vannak-e üres mezők. A PHP-ban megtalálható `preg_match()` függvényben `"/^[a-zA-Z0-9]*$/"` RegEx (Regular Expression) kifejezés felhasználásával biztosított, hogy a felhasználó név csak az abc kis és nagy betűit, illetve számokat tartalmazza.



6. ábra: Regisztrációs form

Egy SQL lekérdezés a *users* táblában megnézi, hogy foglalt-e már a felhasználó által beállítani kívánt felhasználónév. Amennyiben nincsenek hibák, egy prepared statement által összeállított SQL- parancs a jelszó hash-elése után eltárolja az adatbázisban a felhasználónevet, email címet, a hash-elt jelszót és generál egy egyedi azonosítószámot. Ezután megjelenik egy 'Signup success!' felirat a képernyőn. A folyamat során a hibák típusát a *signup.inc.php* a *signup.php* URL-je után írja ki (7. ábra). Az adott hiba fajtája alapján megjelenik egy megfelelő üzenet az oldalon.


```

                                                                    signup.inc.php
...
else if (!preg_match("/^[a-zA-Z0-9]*$/", $username)) {
    header("Location: ../signup.php?error=invaliduid&mail=".$email);
    exit();
} ...

                                                                    signup.php
...
if (isset($_GET["error"])) { ...
    else if ($_GET["error"] == "invaliduid") {
        echo '<p class="signuperror">Invalid username!</p>';
    } ...
} ...

```

7. ábra: Példa nem megfelelő karaktereket tartalmazó felhasználónév esetén

5.4. Bejelentkezés

A regisztrált felhasználó érvényes a felhasználónév és jelszó megadása után bejelentkezhets az oldalra, ezáltal plusz funkciók válnak számára elérhetővé. A **LOGIN** gomb megnyomásával az oldal elküldi a *login.inc.php* fájlban a bejelentkezési adatokat. Amennyiben maradt üresen mező vagy tévesek az adatok, akkor az oldal URL-jében megjelenik a hiba típusa, ezek lehetnek:

```

                                                                    login.inc.php
...
header("Location: ../index.php?error=emptyfields");
...
header("Location: ../index.php?error=sqlerror");
...
header("Location: ../index.php?error=wrongpwd");
...
header("Location: ../index.php?login=wronguidpwd");
...

```

8. ábra: Bejelentkezés során fellépő hibalehetőségek: üresen hagyott mezők, SQL hiba, rossz jelszó, rossz felhasználónév és jelszó

5.5. Georeferálás

A szkennelt raszteres képformátumú térképek csak a saját síkkoordináta-rendszerükben érvényes pixelkoordinátákkal rendelkeznek. Georeferálás során illesztőpontok (GCP – Ground Control Point) felhasználásával kapcsolatot hozunk létre a térkép síkkoordináta-rendszere és a választott földrajzi koordináta-rendszer között. A GCP-k kiválasztásánál meghatározzuk ezek pixelkoordinátáit és térbeli koordinátáit egyaránt. Az illesztőpontok megadása mellett, a számításokhoz egy kívánt eljárást is választani kell. Gyakran használt módszer a polinomiális illesztés, melynek fajtái lehetnek: lineáris, kvadratikus (másodrendű) vagy köbös (harmadrendű). A lineáris módszernél az illesztőpontokra egy elforgatott, egyenközü négyzethálót illeszt a program. A másod- illetve harmadrendű illesztésnél az előzőhöz képest több GCP-re van szükség és a pontok között nagyobb hibalehetőség keletkezik (Timár & Molnár, 2013).

A polinomiális illesztés esetén az általános, m -ed fokú polinomiális összefüggést (Brovelli & Minghini, 2012) a pixelkoordináták (x, y) és vetületi-koordináták (E, N) között:

$$E = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + \dots + a_{m,0}x^m + a_{m-1,1}x^{m-1}y + a_{1,m-1}xy^{m-1} + a_{0,m}y^m \quad (1)$$

$$N = b_{00} + b_{10}x + b_{01}y + b_{20}x^2 + b_{11}xy + b_{02}y^2 + \dots + b_{m,0}x^m + b_{m-1,1}x^{m-1}y + b_{1,m-1}xy^{m-1} + b_{0,m}y^m \quad (2)$$

Ekkor az ismert paraméterek száma $(m+1)(m+2)$ és az illesztéshez minimálisan szükséges GCP-k száma (n_{GCP})

$$n_{GCP} = \frac{(m+1)(m+2)}{2} + 1 \quad (3)$$

Teoretikus és gyakorlati szempontok figyelembevételével (Timár & Molnár, 2013) a georeferálási algoritmus implementálásához a lineáris ($m=1$) eljárást választottam. Így az (1) és a (2) egyenlet az alábbi formára redukálódik:

$$E_{th} := a_{00} + a_{10}x + a_{01}y \quad (4)$$

$$N_{th} := b_{00} + b_{10}x + b_{01}y \quad (5)$$

ahol (E_{th}, N_{th}) az elméleti Easting és Northing értéke az (x, y) képi (pixel) koordinátáknak. A köztük lévő kapcsolatot az ($a_{00}, a_{10}, a_{01}, b_{00}, b_{10}, b_{01}$) paramétervektor írja le, ahol

- a_{00} : A bal felső sarok Easting koordinátája
- a_{10} : 1 pixel jobbra lépéskor az Easting növekménye
- a_{01} : 1 pixel lefelé lépéskor az Easting növekménye
- b_{00} : A bal felső sarok Northing koordinátája
- b_{10} : 1 pixel jobbra lépéskor az Northing növekménye
- b_{01} : 1 pixel lefelé lépéskor az Northing növekménye

Léteznek térinformatikai szoftverek által ismert kvázi szabványok, a koordináta-rendszerben érvényes pozíció leírására. Ezek egyik egyszerű formája az ún. „*World file*”. A georeferálás során kiszámított paramétervektor adatai a fájlba egymás alatti sorokba (a_{10} , a_{01} , b_{10} , b_{01} , a_{00} , b_{00}) sorrendben exportálhatók.

Belátható, hogy lineáris esetben ($m=1$) a (4) -egyenlet által meghatározott minimális illesztőpont szám $n_{GCP}=4$. A paramétervektor meghatározását Gauss-féle legkisebb négyzetek módszerével végeztem Molnár(2013) alapján. Ekkor a problémára felírt általános lineáris egyenletrendszer:

$$A \cdot x = b \quad (6)$$

ahol A alakmátrix a (4) és (5) egyenletekben bemutatott illesztés x paramétervektor szerinti deriváltja, azaz a Jacobi-mátrix, b vektor pedig a térképi koordináták megfigyelési vektora. A legkisebb négyzetek módszer során a reziduálok L_2 szerinti minimalizálására törekszünk az alábbi módon:

$$r = b - A \cdot x \quad (7)$$

$$r^T \cdot r := \min \quad (8)$$

ekkor a (8)-egyenlet:

$$(b - A \cdot x)^T (b - A \cdot x) := \min \quad (9)$$

átrendezve:

$$x = (A^T \cdot A)^{-1} \cdot A^T \cdot b \quad (10)$$

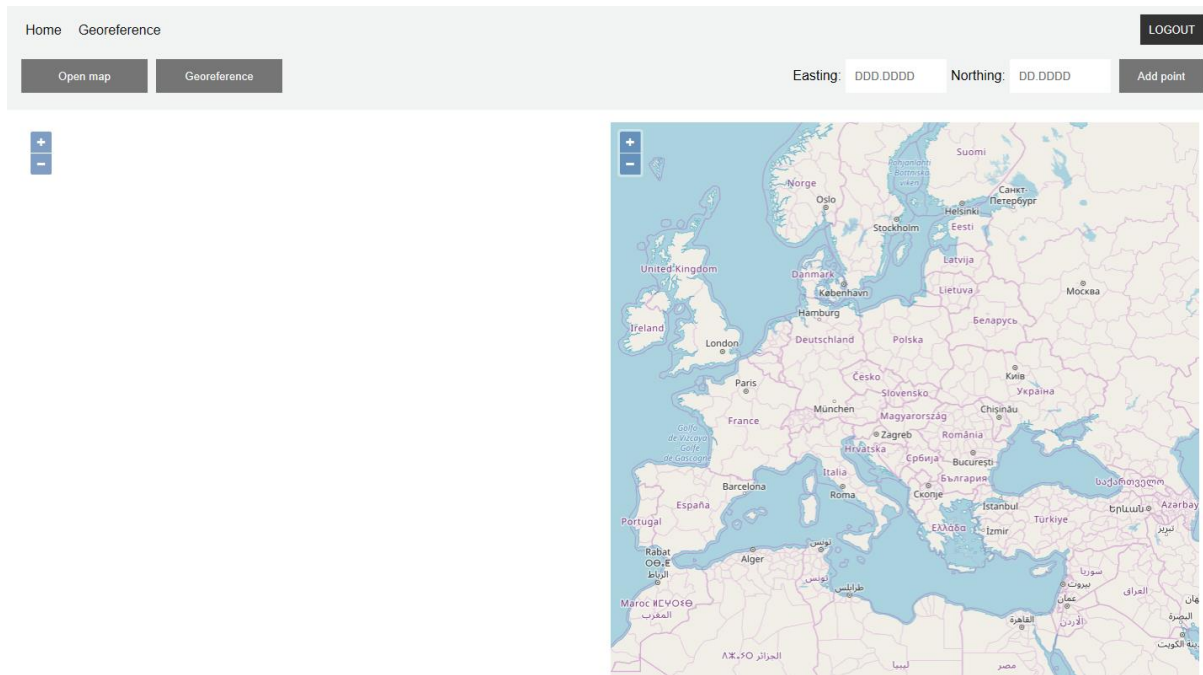
Mivel az illesztőpontok kiválasztásánál nem áll rendelkezésre az adott pixel hibája, megfigyelésüket egyenlő súlyúnak feltételezhetjük. Így a (10) -egyenlet felírható az alábbi alakban:

$$x = N^{-1} \cdot n \quad (11)$$

azaz a normál-egyenlet. Ahol N normálmátrix megegyezik a Jacobi-mátrix és a transzponáltjának szorzatával, n vektor pedig a Jacobi-mátrix transzponáltja és a megfigyelési (tisztatag) vektor szorzatával.

A (7)-es és (10)-es egyenletek alapján iteratív módon kerül meghatározásra az x paramétervektor. Az iteráció konvergencia küszöbértékei a maximális iterációs szám, ami implicit módon biztosítja a végtelen ciklus kizárását, valamint a reziduál küszöbérték, ami a legkiseb négyzetek alapján szolgáló ciklus-megszakítási kritérium.

A fenti eljárás implementációját JavaScriptben végeztem a Math.js könyvtár felhasználásával (de Jong & Mansfield, 2018). A térképek megjelenítéséhez és a GCP-k lerakásához az OpenLayers alkalmazást választottam (9. ábra). Az illesztőpontok elhelyezésére a felhasználónak a térképen történő kattintással nyílik lehetősége. Sok Cartactual térkép tartalmaz koordináta hálót, amit webes alaptérképen nem lehet beazonosítani. Így szükségszerűnek tűnt, hogy a földrajzi koordináták kinyerésére használt térképen input mezők használatával biztosított legyen a GCP-k meghatározott földrajzi pontra elhelyezése.



9. ábra: A georeferáló felület

A georeferálendő térkép a pixel-felbontása alapján egy dinamikusan változó kiterjedésű kép rétegre kerül statikus képként (`ol.source.ImageStatic()`). A fájl megnyitásánál a program ellenőrzi a fájlformátumot (csak .jpg, .jpeg és .png a megengedett), -méretet (50MB a felső határ) és törli az alkalmazásból a korábban hozzáadott vektoros és kép réteget (10. ábra). Hibák fellépésénél a böngésző figyelmeztetést küld. Elegendő mennyiségű GCP esetén lefut a georeferáló script és a szkennelt képet hozzáadja 50%os áttetszőséggel az OSM alaptérképhez (1.sz. melléklet). Ez annyiszor ismételhető meg a felhasználó által, amíg a lehető legpontosabb eredmény keletkezik.

```

...
w = document.querySelector("#img").naturalWidth;
h = document.querySelector("#img").naturalHeight;
extent.splice(2,2,w,h);
mapping.setView(
    new ol.View({
        projection: projection,
        center: ol.extent.getCenter(extent),
        zoom: 2,
        maxZoom: 20,
    })
);
var img = new ol.layer.Image({
    title: 'img',
    source: new ol.source.ImageStatic({
        url: srcurl,
        projection: projection,
        imageExtent: extent,
        zIndex: 0,
    })
});
mapping.addLayer(img);
...

```

10. ábra: Példa szkennelt térkép megnyitására, ahol:

projection: az új extent által módosított, korábban definiált pixel-koordinátarendszer; sourceurl: a megnyitott kép URL címe; mapping: korábban definiált, statikus kép tárolására alkalmas OpenLayers felület

5.6. Térkép feltöltése az adatbázisba

Az **Upload to database** gombra kattintva a térkép adatait kérő form kitöltése után a kép és a megadott információk feltölthetők az adatbázisba. Hibák fellépése esetén az oldal újra töltése és az adatok eltűnésének elkerülése érdekében ezt a folyamatot egy jQuery-ben található AJAX utasítás küldi el a szervernek.

```

...
$(document).ready(function () {
    $("#btnOK").click(function (event) {
        event.preventDefault();
        var form = $('#formUpload')[0];
        var data = new FormData(form);
        data.append("geom", geom);
        data.append("polygon", polygon);
        $("#btnOK").prop("disabled", true);
    });
});

```

```

$.ajax({
    type: "POST",
    enctype: 'multipart/form-data',
    url: "includes/upload.inc.php",
    data: data,
    processData: false,
    contentType: false,
    cache: false,
    timeout: 600000,
    success: function (data) {

        alert("SUCCESS : ", data);
        $("#btnOK").prop("disabled", false);
    },
    error: function (e) {
        alert("ERROR : ", e);
        $("#btnOK").prop("disabled", false);
    }
});
});
});

```

12. ábra: Feltöltésre használt AJAX utasítás, ahol:
*#btnOK: Upload gomb azonosítója; #formUpload: a térkép-adatokat tartalmazó form azonosítója;
 geom és polygon: georeferáló sriptben keletkezett adatok*

Szerver oldalon az adatokat az *upload.inc.php* fájl kapja meg. Ezeket a program további vizsgálatoknak veti alá, a biztonság érdekében, mert az itt definiált feltételrendszerre nem lát rá a kliens oldal. Ellenőrzésre kerülnek a fájladatok, a georeferáló scriptből küldött értékek, az üres mezők esetleges jelenléte, továbbá az adatbázis *maps* tábláján lefut egy keresés, hogy foglalt-e a megadott térképazonosító. Ha nincs hiba, akkor az előkészített adatok bekerülnek az említett táblázatba (11. ábra), a program feltölti a képet a szerverre a weblab főkönyvtárán belüli *images* mappába, valamint ugyanide létrehozza az azonos nevű „world file”-t (pl.: filename.jpg, filename.jpgw).

```

...
$mysql = "INSERT INTO maps (filename, nbr, vol, yr, mapid, ttl, ttlhu, ctry, ctryhu, ol, g)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, PolygonFromText(?, 3857));";
$stmt = mysqli_stmt_init($conn);
...
mysqli_stmt_bind_param($stmt, "ssssssssss", $fileNameNew, $nbr, $vol, $yr, $mapid, $ttl,
$ttlhu, $ctry, $ctryhu, $geom, $polygon);
...

```

upload.inc.php

11. ábra: SQL parancs összeállítása (részlet), ahol:
*\$polygon: scriptből érkező adat, általánosan felírva:
 POLYGON((E₀₁ N₀₁, E₁₁ N₁₁, E₀₁ N₀₁, N₀₀ N₀₀, E₀₁ N₀₁),)*

A PHP `uniqid(prefix, more_entropy)` funkciójával egyedi, 23 karakter hosszúságú fájlneveket generál program (13. ábra). Így az alkalmazás működése függetlenné válik a felhasználó által feltöltött kép nevétől és biztosított, hogy az adatbázis bővítése során nem lesz a fájlnevek között összeférhetetlenség. Az említett függvény `more_entropy` paramétere alapesetben FALSE, azonban TRUE-ra állítva egyedibb és 13 helyett 23 karakter hosszúságú azonosítót hoz létre.

```
upload.inc.php
...
$fileExt = explode('.', $fileName);
$fileActualExt = strtolower(end($fileExt));
$fileNameNew = uniqid('', true)." ".$fileActualExt;
...
```

13. ábra: Egyedi fájlnev generálás:

\$fileName: a fájl eredeti neve; \$fileExt: pont karaktereknél szétdarabolt fájlnev tömbje; \$fileActualExt: a fájl kisbetűsített kiterjesztése; \$fileNameNew: újonnan generált fájlnev

5.7. Főoldal

5.7.1. Keresés

A honlap főoldalának meghatározó funkciója a Search form (14. ábra). Ennek kitöltésével és elküldésével tud a felhasználó keresni a térképi adatbázisban több paraméter alapján. A hat input mező lehetőséget ad, a szám, évfolyam, év, térképazonosító, cím és ország alapján szűrni a tábla rekordjait. A **Search** gomb kattintás eseményre AJAX technikával elküldi a megadott értékeket szerverre a `search.inc.php` fájlnek.

```
index.js
...
$('#btnSearch').prop("disabled", true);
$.ajax({
    type: "POST",
    enctype: 'multipart/form-data',
    url: "includes/search.inc.php",
    data: data,
    processData: false,
    contentType: false,
    cache: false,
    timeout: 600000,
    success: function (data) { ... $('#btnSearch').prop("disabled", false); },
    error: function (e) { ... $('#btnSearch').prop("disabled", false); }}); ...
```

14. ábra: Keresési adatok küldése a szerverre, ahol:

data: A Search form input mezőinek értékei; prop() függvény az AJAX kérés idejére letiltja a Search gombot

Az SQL lekérdezések megírásánál az egyszerűséget és hatékonyságot tartottam szem előtt. Ennek okán az alap szövege változatlan, mindig ugyanazokat a változókat kapja a programból, de ezek tartalma a kereső form-ba beírtak alapján változik. Természetesen ebben a programrészben is átesnek a kliens oldali adatok egy karakterszűrésen a `preg_match()` függvény által, hogy a felhasználó felé lehessen visszajelzést küldeni, ha nem megfelelő karaktereket is használ keresés közben. Az alap SQL lekérdezés a következőképp alakul:

```
search.inc.php
...
$sql = "SELECT filename, nbr, vol, yr, mapid, ttl, ttlhu, ctry, ctryhu, ol, ST_AsGeoJSON(g)
AS p FROM maps WHERE nbr LIKE ? AND vol LIKE ? AND yr LIKE ? AND mapid LIKE ? AND
(ttl LIKE ? OR ttlhu LIKE ?) AND (ctry LIKE ? OR ctryhu LIKE ?)";
$stmt = mysqli_stmt_init($conn);
...
mysqli_stmt_bind_param($stmt, "sssssss", $nbrSQL, $volSQL, $yrSQL, $mapidSQL, $ttlSQL,
$ttlSQL, $ctrySQL, $ctrySQL);
mysqli_stmt_execute($stmt);
...
```

15. ábra: SQL lekérdezés prepared statement alkalmazásával

Látható, hogy a keresés két bemeneti értéke ugyanúgy lefut az angol, illetve magyar szöveget tartalmazó mezőkön is. Az `$nbrSQL`, `$volSQL` stb. változók nem közvetlen a felhasználó által származó adatok, hanem azok kiegészítései. Mindegyik értéke az elején és végén egy-egy százalékjellel egészül ki. Ha a felhasználó egy beviteli mezőt üresen hagy, akkor az az SQL parancsba `'%%'`-ként kerül be. Ezt a MySQL úgy érzékeli, mintha az adott attribútumosztályában az összes elemre keresnénk. Belátható, hogy csak üres bemeneti értékek mellett az eredmény az adatbázis összes rekordját tartalmazza.

5.7.2. Keresés eredménye

A keresés eredménye megjelenik egy dinamikusan változó táblázatban a főoldal alsó felén és egy vektoros, illetve raszteres rétegen az OpenLayers felületen (2.sz. melléklet). Ezek létrehozásához szükséges adatot a *search.inc.php* végzi (16. ábra). A megjelenő táblázatban az adatbázisból származó mezőkhöz egy *Download* nevű oszlop csatlakozik, ami a térkép és a hozzá tartozó „world file” letöltésére szolgál. Továbbá az eredmények vektor réteggént megjelennek a térképes felületen, aminek az egyes elemeihez rendelendő geometriát a `ST_AsGeoJSON(g)` SQL parancs adja. A kurzor a táblázat fölé mozgatva 'pointer' stílusúvá változik, az adott sor szövege kiemelt lesz és a hozzá tartozó elemre közelít a OpenLayers felület. Kattintásra az adott térkép is megjelenik a térképen. Az említett egér-funkciók a *Download* mezőre nem érvényesek, hogy a letöltés és a térképi interakció ne futhasson egyszerre.

```
search.inc.php
...
if ($resultCount > 0) {
    $output = '<h1 align="center">Search results</h1>';
    $output .= '<table id="tableresults"><tr><th>Map ID</th><th>Year</th>
        <th>Volume</th><th>Number</th><th>Title</th><th>Cím</th>
        <th>Country</th><th>Ország</th><th>Download</th></tr>';
    $gjson = '{ "type": "FeatureCollection", "crs": { "type": "name", "properties": { "name":
        "EPSG:3857"}}, "features": [';
    while ($row = mysqli_fetch_assoc($result)) {
        $output .= '<tr><td>'.$row["mapid"].'</td><td>'.$row["yr"].'</td>
            <td>'.$row["vol"].'</td><td>'.$row["nbr"].'</td><td>'.$row["ttl"].'</td>
            <td>'.$row["ttlhu"].'</td><td>'.$row["ctry"].'</td><td>'.$row["ctryhu"].'</td>
            <td><a href="includes/download.inc.php?file='.$row["filename"].'">Map</a>,
            <a href="includes/download.inc.php?file='.$row["filename"].'w">
            Word file</a></td></tr>';
        $gjson .= '{"type": "Feature", "properties": { "Mapid": "'.$row["mapid"].'", "Fname":
            "'.$row["filename"].'", "geom": "'.$row["ol"].'" }, "geometry": '.$row["p"].',';
    }
    $gjson .= ']';
    echo $output;
    echo $gjson;
    exit();
}
...
```

16. ábra: Adatok generálása a megjelenítéshez az SQL lekérdezés eredményéből, ahol: *\$resultCount*: egyezés esetén a keresett sorok száma; *\$row*: az adatbázis egy sorának tartalma; *\$output*: táblázat létrehozásához szükséges kód *\$gjson*: az eredményekből létrehozott GeoJSON

5.7.3. Letöltés

A keresés eredményének táblázatában a *Map* és *World file* linkekre (ezek illesztik az URL-be a fájlneveket) kattintva letölthető a szerverről az adott térkép, valamint annak földrajzi helyzetét tartalmazó szöveges fájl. Ezt a műveletet a *download.inc.php* végzi, a szükséges információt GET metódussal szerzi. A letöltésért a definiált `header()` és `readfile()` utasítások felelősek (17. ábra). Amennyiben minden rendben megy vagy hiba lép fel, a program visszairányítja a felhasználót a főoldalra.

```
...
header("Cache-Control: public");
header("Content-Description: File Transfer");
header("Content-Disposition: attachment; filename=$filename");
header("Content-Type: application/octet-stream");
header("Content-Transfer-Encoding: binary");

readfile($filepath);
...
download.inc.php
```

17. ábra: A letöltést végző utasítások, ahol:
\$filepath: a letöltendő fájl elérési útja a szerveren

6. Tesztelés, bővítési lehetőségek

A weboldal fejlesztése során elengedhetetlen volt a folyamatos tesztelés, a felmerülő hibák felderítésére és kiküszöbölésére. Erre a programozási környezettől függően többféle módszert használtam.

JavaScript-ben programozás közben az egyes műveletek eredményét, változók típusát a `console.log()` paranccsal írtam ki a böngésző konzoljára. Így a műveletek részeredményei könnyen megismerhetők, és változók összeférhetetlensége esetén a hiba forrása hamar felderíthető. Ez a hibakeresési és ellenőrzési módszer a végleges állományokból törlésre került. A működés közben felmerülő problémák kezelésére és a felhasználó felé történő kommunikációra is szükség van. Az `if else` elágazásokba épített programmal és az `alert()` funkcióval lehetőség nyílik a jelentkező hiba értesítésére a böngészőn keresztül.

A szerver oldali fejlesztés során és működés közbeni interakcióra, hibakeresésre szintén `if else` elágazásokat használtam a PHP `echo` parancsával (változók értékeinek, illetve szövegek kiírására használt), valamint a kód futását leállító `exit()` és `die()` (a kettő lényegében ugyanaz) függvények kombinálásával.

A weboldal gördülékeny működéséhez nagyban hozzájárult folyamatos tesztelés, mert munkarészenként kiderültek a felmerülő hibák és hibalehetőségek, amiket rögtön javítani lehetett.

Az elkészült térképkatalógus és georeferáló rendszer a gyakorlatban is megálja a helyét, de még rengeteg potenciál és lehetőség van benne. Célszerű lehet jövőben egy jelszó újra igénylő funkció kialakítása, ha egy felhasználó elfelejti azt. Hasznos lehetne egy pdf megjelenítő és böngésző modul kialakítása, hogy a beszkenelt folyóiratszámok egyben hozzáférhetővé váljanak. Ez a weboldal információ szolgáltatásának képességét jelentősen kibővítené, mert így a *Cartactualok* feltöltése mellett lehetőség nyílna az 1971-ban indult *Cartinform* katalogizálásra is.

7. Összegzés

Dolgozatom célja egy térképi adatbázis és a hozzá tartozó webes megjelenítő felület létrehozása volt, amelyen keresztül lehetőség nyílik térképek georeferálására és adatbázisba feltöltésére.

A dolgozat elején igyekeztem megfelelő módon ismertetni katalogizálandó *Cartactual* folyóiratot, megfogalmazni a követelményeket a weboldallal szemben és ismertetni a tervezési folyamatot.

Utána röviden bemutattam a fejlesztés során használt programozási környezeteket, valamint a következő fejezetekben igyekeztem minél részletesebben ismertetni a weboldal felépítését, és a létrehozott funkciókat.

A dolgozat elkészítése során hasznos webfejlesztési ismeretekkel gyarapodtam mind szerver-, mind kliens oldalról.

8. Irodalomjegyzék

- BROVELLI, Maria, A. & MINGHINI, Marco (2012): Georeferencing old maps: a polynomial-based approach for Como historical cadastres. *e-Perimetron*, 7(3):97-110
- DE JONG, Jos & Mansfield, Eric (2018): Math.js: An Advanced Mathematics Library For JavaScript. *Computing in Science & Engineering*, 20(1):20-32.
- ELEK, István (2011): Adatbázisok, térképek, információs rendszerek. ELTE Eötvös Kiadó, Budapest, 181p.
- GARRETT, Jesse, James *et al.* (2005): Ajax: A new approach to web application
https://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf
(Utolsó elérés: 2019. 05.15.)
- GERCSÁK, Gábor, MÁRTON, Mátyás, TÓTH, Mária (2016): Ötven éve indult a Cartactual (1965—1993). *Geodézia és Kartográfia* 68(5-6):11-17
- KLINGHAMMER, István & PAPP-VÁRY, Árpád (1983): Földünk tükre a térkép in: Schiller Jánosné (ed.): A Térképészet nemzetközi szervezetei, folyóiratai. Gondolat Kiadó, Budapest, pp. 354-356
- MOLNÁR, Gábor (2013): Geofizikai inverzió. ELTE Geofizika Tanszék, Budapest, 111p
- NIXON, Robin (2014): Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition. O'Reilly Media Inc., Sebastopol, 729p
- TIMÁR, Gábor, MOLNÁR, Gábor (2013): Térképi vetületek és alapfelületek. Eötvös Loránd Tudományegyetem, Budapest, 87p.

9. Köszönetnyilvánítás

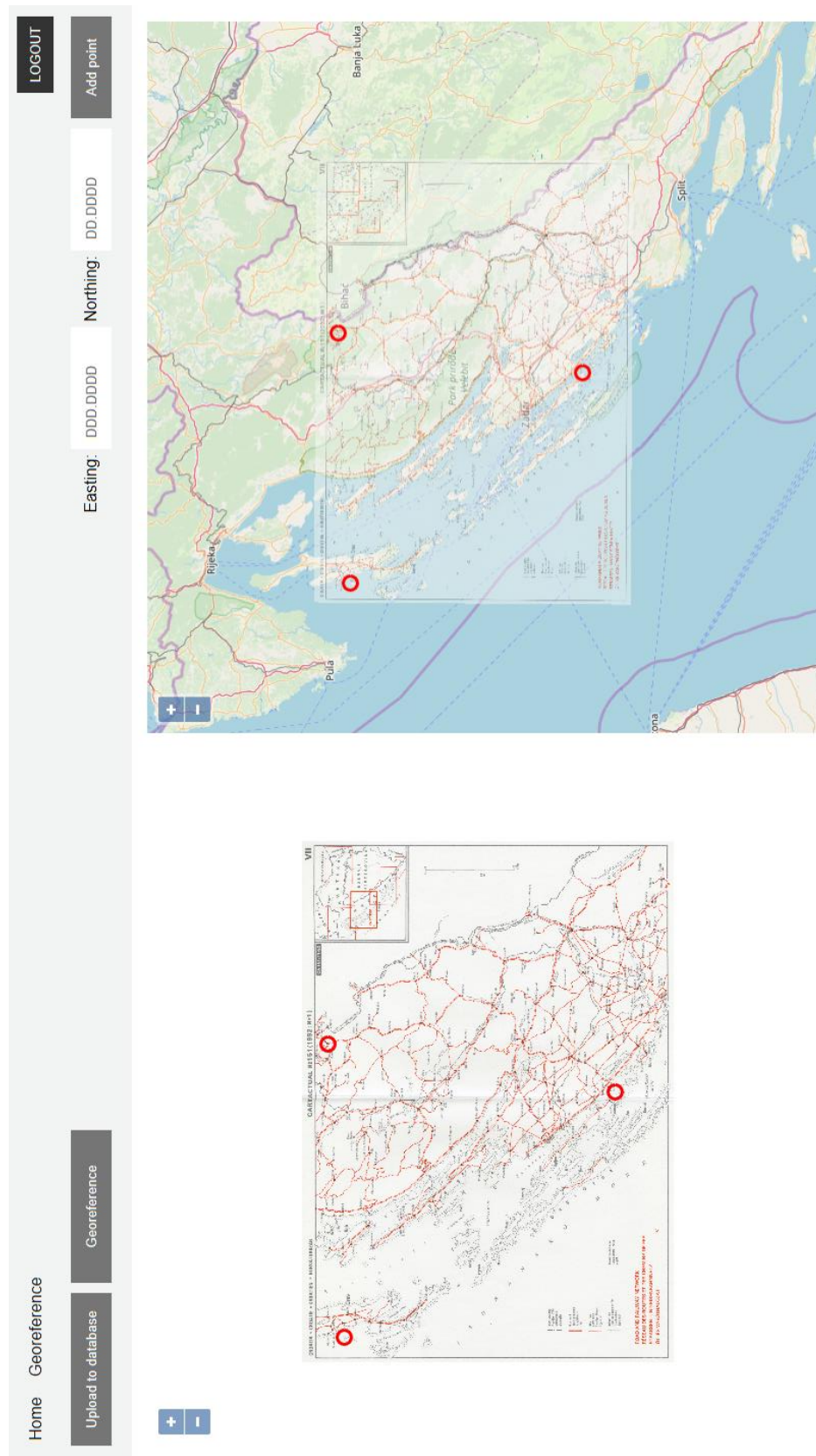
Elsősorban szeretném megköszönni témavezetőmnek, Gede Mátyásnak, segítőkészségét, türelmét, valamint hasznos tanácsaival segítette, hogy a dolgozatom időben elkészüljön.

Továbbá köszönettel tartozom Zentai Lászlónak a téma ajánlásáért és Gercsák Gábornak, hogy rendelkezésemre bocsátott forrást a *Cartactuallal* kapcsolatban.

Végül szeretném megköszönni családomnak a támogatásukat és türelmüket.

10. Melléklet

1.sz. melléklet: Georeferálás eredménye hozzáadva az OSM térképhez:



2.sz. melléklet: Keresés eredménye táblázatban és poligonokként az OpenLayers felületen:

Home

Georeference

Search

Number:

Volume:

Year:

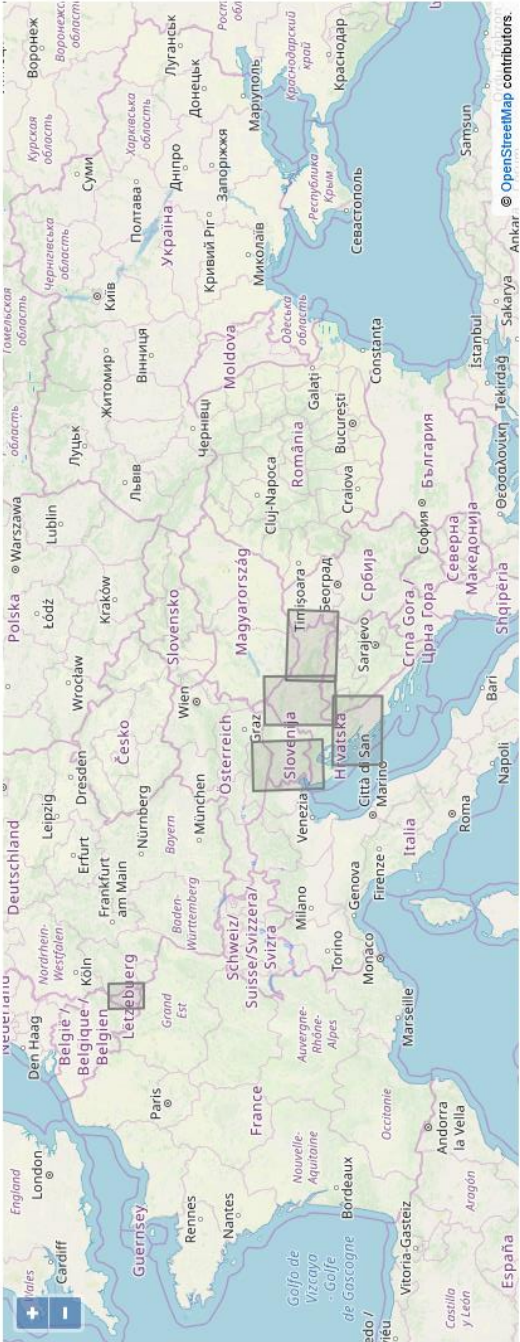
Map ID:

Title:

Country:

Search

LOGOUT



Map ID	Year	Volume	Number	Title	Cim	Country	Ország	Download
CA 151/11-92	1992	28	151	Eoad and railway network	Út- és vasúthálózat	Croatia	Horvátország	Map , World file
CA 151/10-92	1992	28	151	Eoad and railway network	Út- és vasúthálózat	Croatia	Horvátország	Map , World file
CA 151/9-92	1992	28	151	Eoad and railway network	Út- és vasúthálózat	Croatia	Horvátország	Map , World file
CA151/2-92	1992	28	151	Administrative division and population	Közigazgatási beosztás és népesség	Luxemburg	Luxemburg	Map , World file
CA 151/4-92	1992	28	151	Eoad and railway network	Út- és vasúthálózat	Slovenia	Szlovénia	Map , World file

Magyar Zoltán

Eötvös Loránd University, Department of Cartography and Geoinformatics

©2019

30

DIPLOMAMUNKA LEADÁSI és EREDETISÉG NYILATKOZAT

Alulírott **Magyar Zoltán**

Neptun-kód: **DZM5YL**

az Eötvös Loránd Tudományegyetem Informatikai Karának, Térképtudományi és
Geoinformatikai Tanszékén

**Cartactual folyóirat térképi adatbázis szerkezetének és webes megjelenítő felületének
létrehozása**
című diplomamunkámat a mai napon leadtam.

Témavezetőm neve: **Dr. Gede Mátyás**

CD-t / DVD-t mellékelek (aláhúzendő): igen nem

Büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom, hogy jelen
szakdolgozatom/diplomamunkám saját, önálló szellemi termékem; az abban hivatkozott
szakirodalom felhasználása a szerzői jogok általános szabályainak megfelelően történt.

Tudomásul veszem, hogy szakdolgozat/diplomamunka esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

A témavezető által benyújtásra elfogadott szakdolgozat PDF formátumban való elektronikus
publikálásához a tanszéki honlapon

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2018. május 15.

.....
hallgató aláírása