

Eötvös Loránd Tudományegyetem  
Informatikai Kar  
Térképtudományi és Geoinformatikai Tanszék

# **DIGITÁLIS TÉRKÉPTÁR TERVEZÉSE ÉS KIVITELEZÉSE**

**UNGVÁRI ZSUZSANNA**

térképész szakos hallgató

Témavezető:

**DR. GEDE MÁTYÁS**

egyetemi adjunktus



Budapest

2012

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
2.1 A rendszerrel szemben támasztott követelmények	4
2.2 Az oldal működésének ismertetése	5
2.3 A fejlesztéshez használt eszközök	7
<b>3. A MySQL adatbázis</b>	<b>8</b>
3.1 MySQL Spatial Extensions	8
3.2 Az adattáblák létrehozása	11
3.3 Adatok felvétele és lekérdezése	12
<b>4. A PHP szkriptnyelv</b>	<b>14</b>
4.1 A PHP használata	14
<b>5. A JavaScript nyelv</b>	<b>22</b>
5.1 A Google Maps	22
5.2 Az OpenLayers és az OpenStreetMap	28
5.3 A Google Earth	30
<b>6. A honlap megjelenése</b>	<b>32</b>
<b>7. A térképek</b>	<b>34</b>
7.1 A térképek feldolgozása	34
7.2 Az első térképek	35
<b>8. Kitekintés</b>	<b>37</b>
<b>9. Összefoglalás</b>	<b>38</b>
<b>10. Köszönetnyilvánítás</b>	<b>39</b>
<b>11. Irodalomjegyzék</b>	<b>40</b>
<b>12. Melléklet</b>	<b>43</b>

# 1. Bevezetés

Felgyorsult világunkban szinte természetessé vált, hogy a könyvtárakban a régi cédulakatalógusok helyett digitális keresőket, adatbázisokat alkalmazunk. Egy térképtár esetében hasonló a helyzet, de itt nemcsak az alap keresőfunkciókat szeretnénk használni, mint a cím, szerző, vagy jelzet alapján történő keresés, hanem felmerülhet az igény, további információszerezésre magáról a kiadványról. Egy olyan térképtárban, ahol sok régi térkép van, állagmegóvásuk végett, érdemes létrehozni egy digitális térképtárat, amelyben a térképeket a számítógépen is meg lehet tekinteni. A régi térképek tanulmányozása során érdekes lehet összevetni a korabeli állapotokat a mostani földrajzi környezettel, ezért a digitális térképtárban létrehozok egy földrajzi megjelenítő és kereső rendszert is.

Munkám egy lehetséges alternatívaként szolgál az Országos Széchényi Könyvtár Térképtára számára.

Egy térképtár bővítése sok munkát igényel, és folyamatosan történik, néha a már meglévő információk módosításával, növelésével. Ebből a megfontolásból egy feltöltő oldal is elérhető lesz, amelynek használata egyszerű, nem igényel adatbázis-kezelési ismereteket.

A következő fejezetekben részletesen ismertem a rendszerrel szemben támasztott követelményeket, majd az adatbázis-építés folyamatát, a programozást és az adatokkal való benépesítést.

## **2. Alapok**

### **2.1 A rendszerrel szemben támasztott követelmények**

Munkám célja, hogy egy olyan digitális térképtárat hozzak létre, amelyben nemcsak a keresőrendszer működik, hanem emellett a megjelenítés is elérhető a honlapon. Továbbá szeretném megvalósítani az új térképek felvételét egy könnyen kezelhető oldalon keresztül.

Ahhoz, hogy működőképes rendszert hozzak létre, fel kell állítanom bizonyos elvárásokat a honlappal szemben. Mielőtt azonban részletesen ismertetném a követelményeket, röviden bemutatom a tervezett honlapot.

Az oldal alapvetően két részből fog állni: a keresés funkció publikus, bárki használhatja, ezzel szemben a feltöltő oldal megtekintéséhez már felhasználónév szükséges. Keresni többféle attribútum alapján lehessen, pl. térkép címe, szerzője, vagy kiadója, ezen kívül valós cím vagy földrajzi koordináták alapján is. Az egyes térképekhez tartozzon rövid leírás, valamint képfájlként, továbbá a Google Earth-ben vagy Maps-ben fedvényként, vagyis földrajzi környezetben is meg lehessen jeleníteni a térképet.

A feltöltő oldal használata ezzel szemben már korlátozott. Bejelentkezés után különböző jogosultságokat különítünk el. Az első szinten csak új térképet lehet létrehozni, illetve a saját rekordokat javítani, a következő szinten már mások által létrehozott adatlapokat is lehet módosítani. Végül az adminisztrátorok elfogadhatják, illetve törölhetik az egyes rekordokat, illetve új felhasználót adhatnak hozzá.

A következőkben pontokba szedtem a legfontosabb elvárásokat:

1. A térképek digitalizálása és georeferálása után az állományokat könyvtárakba szervezzük.
2. A térképek és felhasználók adatait táblákban tároljuk.
3. A honlap képes kapcsolódni az adatbázishoz, lehessen az adattáblákban lekérdezéseket végrehajtani, és az így kapott eredményt megjeleníteni, valamint egyidejűleg hozzáférjen a könyvtárakban található fájlokhoz is.
4. Az adattáblák szerkeszthetőek legyenek a honlapon keresztül.
5. A feltöltő oldal kezelni tudja a felhasználók be- és kijelentkezését.
6. Elegáns megjelenés, könnyű használat.

## **2.2 Az oldal működésének ismertetése**

### **A keresés**

A keresést rögtön a kezdőoldalon lehet végrehajtani. A keresésre több lehetőség is van, melyek közül az első a térkép attribútumai alapján történik: a szöveges mezőbe kell beírni a keresendő kifejezést, és a legördülős listából választható az oszlop, amiben keresünk, a térkép címe, jelzete, kiadója, szerzője, méretaránya, vetülete, a kiadás helye vagy ideje, és egyéb kulcsszavak. A kereső gombot megnyomva, megkapjuk az eredménytáblát. Ha a térkép címére kattintunk, megtekinthetjük a részletes adatlapját. A részletes leíráson kívül a térkép nagyfelbontású képét, valamint a Google Earth-re vagy Maps-re vetített réteggént is megtekinthetjük.

A másik keresési lehetőség a koordináták alapján történő keresés. Ezt többféleképpen oldottam meg. Az első opció, hogy beírunk egy címet, ehhez a geokódoló földrajzi koordinátát rendel, ami alapján elindul a keresés az adatbázisban. A másik alternatíva, hogy mi visszük be a koordinátákat. Ezt megtehetjük kézzel, vagy egy egérgomb klikkel a Google Maps térképen. A kereső mindhárom esetben azt vizsgálja, hogy a megjelölt pont, melyik térképeken tűnik fel. Az eredmény itt is egy táblázatban tér vissza, valamint a Google Maps-es térképen sokszögek reprezentálják a térképek által lefedett területet. Ezek is érzékenyek a kattintásra, vagyis ez alapján is megtekinthető a részletes adatlap.

Mivel a Google Maps térképszolgáltatás a nagyfelhasználóknak fizetőssé vált, és felkészülve arra, hogy mindenki másnak is az lehet egyszer, az egész kereső rendszert létrehoztam az OpenStreetMap térképén is.

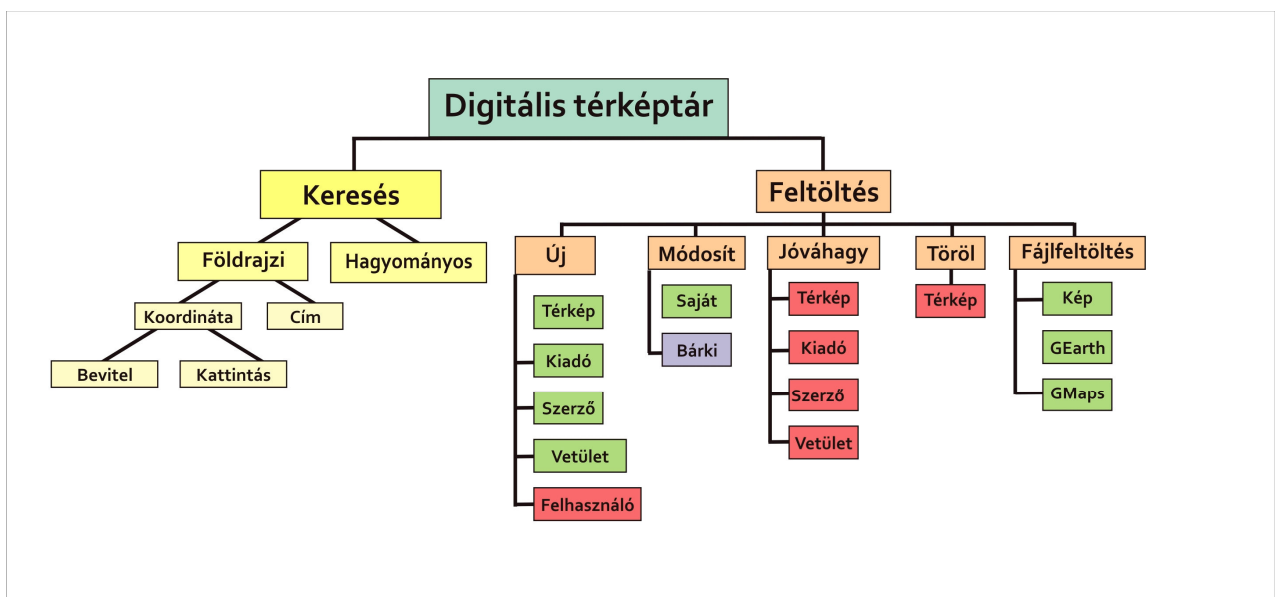
A keresőn kívül a menüsorban lehet elérni a súgót, valamint egyéb hasznos linkeket, és a feltöltő oldalt.

### **A térkép feltöltése**

A feltöltő oldal eléréséhez felhasználónévvel és jelszóval kell rendelkezni, ezek megadása után a program különböző jogosultságok alapján átirányít a megfelelő aloldalra. Három jogosultsági szintet léptettem életbe. A legszigorúbb korlátozás alá eső felhasználók, csak az alapfunkciókat érhetik el. Létrehozhatnak új térképet, kiadót, szerzőt és vetületet, valamint ezeket módosíthatják is mindaddig, amíg nem történt meg jóváhagyás. Természetesen fájlokat is tölthetnek fel. A második szint erre épül: az előző

lehetőségek annyiban enyhülnek, hogy nemcsak saját rekordot lehet módosítani, amíg jóvá nem hagyták, hanem bárki által létrehozott és már jóváhagyott adatlapot is, ekkor ezek bekerülnek az ellenőrizetlen sorok listájába. Végül, lesznek, akikre nem érvényes semmilyen korlátozás sem. Ők szabadon módosíthatnak, vagy törölhetnek rekordokat, és rájuk marad az ellenőrzés is. Mivel ők ismerik legjobban az adatbázist, ezért hozzáadhatnak új felhasználókat a rendszerhez, ezért nevezhetjük őket egyszerűen adminisztrátoroknak is.

Felmerülhet a kérdés, mért van szükség a három szint elkülönítésére. A cél alapvetően az, hogy ne csak egy ember módosíthassa a térképtárat, hiszen a munkája hiányában megszűnhet működni, fejlődni. A feltöltő oldal létrehozása tulajdonképpen azért szükséges, mert nem biztos, hogy minden felhasználó rendelkezik adatbázis-kezelési ismeretekkel, ezen ismeretek hiánya véletlen adatvesztéshez is vezethet, törölhetnek, módosíthatnak dolgokat, miközben egyszerűbb műveletet akarnak végrehajtani, ezzel tönkreteszhetik hetek, hónapok munkáját is. Az egyes szintek elkülönítése szintén az adatok védelmét szolgálja. A felhasználók gyakorlatot szerezhetnek a térképek feltöltésében, mielőtt megkapnák az adminisztrátori jogosultságot, amely a fejlesztő részéről magas szintű bizalommal és bizonyos felelősséggel is jár. Továbbá lehetővé tesszük azt, hogy a bővítésbe minél többen bekapcsolódjanak, értékes információkkal gazdagíthatják az adatbázist. A felhasználóbarát megjelenés és könnyű kezelés ezért volt fő szempont a fejlesztés során.



1. ábra: A digitális térképtár aloldalai

Az első ábra szemlélteti a weboldal felépítését, és az egyes részek használatához szükséges jogosultságokat. A különböző színek az egyes jogosultságokat különítik el, a sárga a szabadon használható keresőoldalt mutatja. A rózsaszínnel foglaltam össze a feltöltést, mely megtekintése csak bejelentkezés után lehetséges. Ezen belül a zöld mutatja a bárki által elérhető funkciókat, a lila a középső szintet, végül a pirossal jelzett oldalakat csak az adminisztrátorok érhetik el.

## 2.3 A fejlesztéshez használt eszközök

Ebben az alfejezetben a honlap fejlesztéséhez használt programokat írom össze, a következő részekben pedig részletezem jellemzőiket.

A honlap tervezése során több eszközt is felhasználtam. A térképek georeferálását *Global Mapper*-rel végeztem, valamint ennek a programnak a kitűnő formátumkezelési előnyeit is kihasználtam, ezzel hoztam létre a tárhelyre feltöltendő fájlokat.

A honlap fejlesztését *Mozilla Firefox 11*-es verziójában végeztem, kihasználva a böngészőbe épített hibakeresési lehetőségeket, vagyis a Webfejlesztő eszközöket, úgymint a Webkonzolt és az Elemvizsgálatot.

Az adatbázist a *MySQL 5.0*-ás verziójában építettem fel. Az ELTE Térképtudományi és Geoinformatikai Tanszék mercator.elte.hu szerverén a *MySQL 5.0.51a* verzió van telepítve *PHPMysqlAdmin 2.11.8.1*. kezelőfelülettel. Ugyan mindkét eszköznek létezik későbbi verziója a *MySQL* esetében már megjelent az 5.6-os és a *PHPMysqlAdmin*-nál is a 3.5-ös verzió, ugyanakkor ezek biztonsági elemei még nem teljesen teszteltek, illetve működnek, ezért nem futtathatók a tanszék szerverén, amely Debian Stable disztribúción alapszik. Szerencsére, az *MySQL 5.0* tartalmazza a *Spatial Extensions*-t, vagyis a térbeli kiterjesztést, amelyet felhasználtam és később részletesen ismertetem.

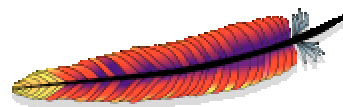
A *HTML* oldal dinamizmusának fejlesztéséhez a *PHP* és *JavaScript* szkriptnyelveket használtam fel, az oldal a külalakját a *CSS3* stíluslapokkal nyerte el.

A következő fejezetekben az egyes rendszerek, illetve verziók sajátosságait ismertetem, valamint az adatbázis-építés, a honlap fejlesztésével kapcsolatos tapasztalataimat osztom meg az olvasóval.

### 3. A MySQL adatbázis

„A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. A szoftver eredeti fejlesztője a svéd MySQL AB cég volt, 2008 januárjában a Sun felvásárolta 800 millió dollárért a céget. 2010. január 27-én a Sun felvásárolta az Oracle Corporation, így a MySQL is az Oracle tulajdonba került. A MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.” (Wikipédia-MySQL, 2012)

A tanszéki Mercator szerveren is Linux operációs rendszer alatt fut egy Apache HTTP Webserver, amelyre telepítették a MySQL-t. „Az Apache sok szabványt támogat, melyeknek nagy része lefordított modulok formájában áll rendelkezésre a mag kiegészítéseként. Ezek a modulok sok területet lefednek a kiszolgáló oldali programnyelv támogatástól kezdve a hitelesítési sémákig.” (Wikipédia-Apache, 2012) Ilyen támogatott programnyelv a PHP is. A MySQL használatát szintén PHP-ban megírt PHPMyAdmin kezelőfelület teszi egyszerűbbé. MySQL-ben az admintól saját kezelőfelületet kaptam, melyen jogosultságom van új adatbázis és az adatbázishoz tartozó műveletek végrehajtására. A kezelőfelület ugyan megkönnyíti a MySQL használatát, de a PHPMyAdmin telepített verziója még nem képes kezelni a Spatial Extensions-t, az SQL parancsokat, ezért csak szövegesen tudom a Lekérési ablakban megadni, ezért dolgozatomban inkább az SQL parancsok említésén keresztül mutatom be az egyes adattábla-műveleteket.



*2. ábra: Az Apache Webszerver logója*



*3. ábra: A MySQL logója*

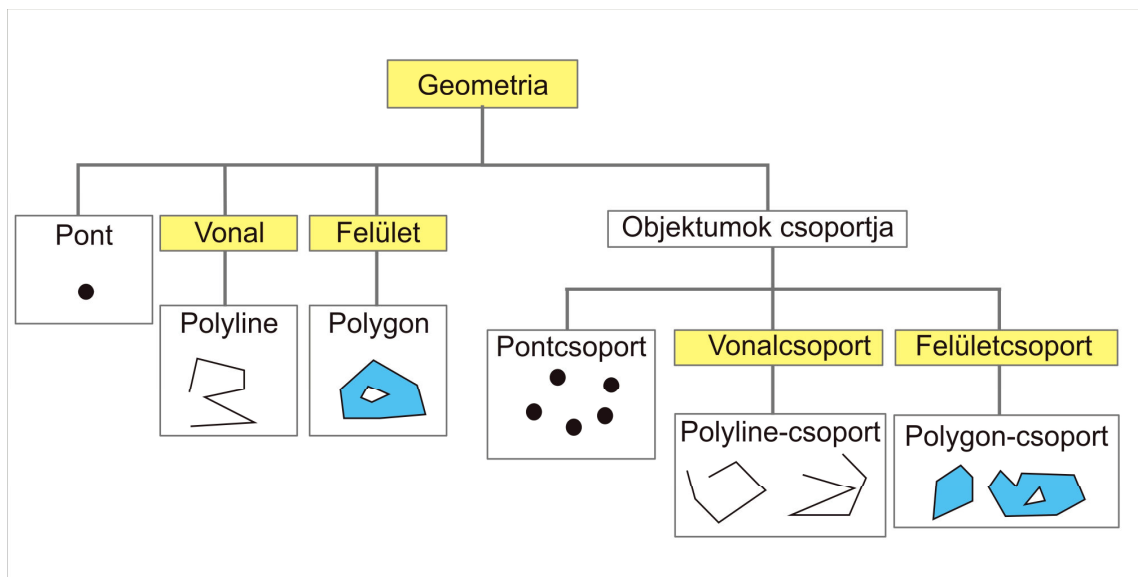
#### 3.1 MySQL Spatial Extensions

Az Open Geospatial Consortium (OGC) több mint 250 cég és egyetem közreműködésével létrejött nemzetközi szervezet, amelynek célja olyan alkalmazások



fejlesztése, amely a térbeli adatok feldolgozását és elemzését egyaránt elvégzi. A MySQL is csatlakozott ehhez a kezdeményezéshez, ennek keretében hozták létre a MySQL Spatial Extensions-t. A Spatial Extensions nemcsak a térbeli adattípus létrehozását jelenti, hanem az elemző funkciók beiktatását is a lekérdező nyelvbe. Mivel az SQL ezen kiterjesztése kevésbé ismert, ezért ennek elméleti háttérével részletesebben foglalkozom.

Mindenekelőtt definiálni kell, mi számít térbeli tulajdonságnak. A válasz, hogy minden olyan entitás vagy tér, amely valamilyen helyhez, vagyis koordinátához köthető, illetve azzal leírható. Az alábbi ábra szemlélteti, hogy az Open GIS szerint létrehozott geometriai modellben milyen osztályokat találunk. A sárgával kiemelték nem tartoznak a MySQL térbeli adattípusaihoz.



**4. ábra: A MySQL térbeli adattípusai (Grimmer, 2003)**

A támogatott térbeli adatformátumok a következők: a Well-Known Text (WKT) és a Well-Known Binary (WKB) formátum. A MySQL által támogatott térbeli adattípusokat az előző ábrán mutattam be, az adatbázisomban a pontokra és polygonokra lesz szükségünk. Polygonok reprezentálják majd a térkép által lefedett területet. Ezeket az adatokat tartalmazza a „poly” mező. A honlapon, mint már említettem, lesz egy földrajzi adatok alapján működő kereső rendszer is, melynek lényege, hogy kiválasztunk egy földrajzi pontot, és megnézzük, hogy azt tartalmazza-e a polygon, vagyis a térképen rajta lesz-e a pont vagy sem.

Az előbbiből is következik, hogy térbeli elemző funkciók megléte nélkül értelmetlen lenne térbeli adatbázisokat készítenünk. Négy csoportba sorolhatjuk az elemző funkciókat:

- Konvertáló függvények.
- Minőségi és mennyiségi információkat elemző függvények.
- Térbeli kapcsolatok leírása két objektum között.
- Egy objektumból egy másik objektumot készítő függvények.

A konvertáló függvények azok, amelyek string-ből geometry típusú adatot készítenek vagy fordítva: `GeomFromText()`, a visszaalakító függvény pedig az `AsText()`. (Mivel dolgozatomban az adatbázisba nem kerültek bináris adatok, ezért az ilyen jellegű funkciókat most nem említem.) Számos olyan függvény létezik, amelyik magáról az objektumról, alakzatról szolgáltat információkat. Ezek tartoznak a második csoportba. A harmadik csoportban viszont több hasznos elemző függvényt találunk, amelyet egy térképtárban futó lekérdezéseknél egyszerűen használhatunk. Egyik ilyen függvény a `CONTAINS`. Ha meg akarjuk vizsgálni, hogy a polygon felszíne tartalmazza-e az adott pontot, ezt használjuk, ha az állítás igaz, az eredmény 1, ha hamis, akkor 0 értékkel tér vissza. Ezek a függvények egyébként nemcsak pont és polygon viszonyának vizsgálatára használhatók.

Sajnos, ezek a függvények az 5.0-ás verzióban működnek, de még nincsenek hitelesítve. Azok, amelyek hibátlanul működnek hasonló eredménnyel térnek vissza, mint az MBR-alapú függvények (MBR-Minimal Bounding Rectangle: Legkisebb befoglaló téglalap). A MySQL 5.6.1 verziójától kezdve bonyolultabb alakzatokat is képesek vizsgálni ezek a függvények. Ezek nevét egy „ST\_” előtaggal bővítették. Így a `CONTAINS()`-ből, amely eddig a legkisebb befoglaló téglalapot nézte meg, most már az objektum valós alakját vizsgáló `ST_CONTAINS()` használható.

Ezek alapján a térbeli elemző függvényeket két csoportra bonthatjuk:

- Függvények, amelyek két objektum közötti térbeli viszonyt tesztelnek
- Függvények, amelyek két objektum befoglaló téglalapja közti viszonyt tesztelik.

Ebből a példából is láthatjuk, ha majd egy olyan szerveren futtat az alkalmazás, amelyre a legújabb MySQL verzió lett telepítve, lényegesen pontosabb eredményt kapunk vissza egy térbeli lekérdezés esetén. (MySQL Reference, 2012)

A következő fejezetben a felhasznált SQL parancsokat mutatom be.

## 3.2 Az adattáblák létrehozása

Első lépésként az adatbázisomban adattáblákat kell létrehoznom. Két tábla mindenképpen szükséges: egyikben a térképek adatait tárolom, a másikban, pedig a felhasználókét. Azonban korábbi tapasztalatok azt mutatták (Egyetemi Digitális Térképtár) (Elek, Gercsák, 2012), hogy a többször előforduló, ismétlődő információkat nem érdemes ugyanabban az adattáblában sokszor eltárolni, mivel a többszörösen ismétlődő szöveg csak növelné az adatbázis méretét. Erre találták ki a relációs adatbázisokat. (Elek, 2006) A térképek táblájában definiálunk egy elsődleges kulcsot, amely az adott rekordot (térképet) egyértelműen azonosítja. Ez egy automatikusan növekvő számláló lesz, a mi esetünkben, nevezzük Id-nek. A legtöbbször ismétlődő attribútumok a térkép kiadója, szerzője, valamint vetülete, ezeknek külön adattáblákat készítek. Minden rekordhoz itt is hozzárendelődik egy elsődleges kulcs. A térképek és kiadók, szerzők, és vetületek táblákat valamilyen módon össze kell kapcsolni. Erre szolgál az idegen kulcs, amely egy másik tábla elsődleges kulcsára mutat. A térképek tábla kiadó mezőjébe az egyes rekordoknál nem szöveg, hanem egy szám fog kerülni, amely azonos a kiadók tábla elsődleges kulcsával, azaz Id-jével. Hasonlóan történt ez a vetületek és szerzők tábla esetében is.

Az adattáblákat a következő paranccsal hoztam létre. Az egyes táblák mezőit lásd részletesen a mellékletben.

```
CREATE TABLE oszk_maps
(
  ID int(10) NOT NULL PRIMARY KEY,
  title text,
  ...)
```

Ha egy táblát létrehoztam már, de módosítani szeretném a mezőit az ALTER TABLE parancsot használnom. Már említettem, hogy a PHPMyAdmin felületen nem lehet kezelni a Spatial kiterjesztést, viszont a MySQL már ismeri. Ezért csak Lekérési ablakon keresztül töltöm fel:

```
ALTER TABLE oszk_maps ADD poly geometry
```

Az eddigiekben ismertettem az adattáblák és a mezők létrehozásának módját, a következőkben pedig a táblák adatokkal való feltöltésének, módosításának, illetve törlésének lehetőségeit mutatom be.

### 3.3 Adatok felvétele és lekérdezése

A MySQL sok utasítást és függvényt tartalmaz, a következőkben csak azokra térek ki, amelyet a honlap elkészítése során felhasználtam.

Új rekord felvételéhez az INSERT INTO parancsot alkalmazom, melynek szintaxisát a következő példa mutatja:

```
INSERT INTO oszk_maps (title, author, pubdate,...)
VALUES („Budapest közlekedése”, „Kartográfia Vállalat”,
„1990”,...)
```

Ha már létezik egy rekord azt az UPDATE parancssal lehet módosítani:

```
UPDATE oszk_maps SET title="Magyarország domborzata",
pubdate=1989,... WHERE ID=3
```

Az adott sor törléséhez a DELETE-et használjuk:

```
DELETE FROM oszk_maps WHERE ID=3
```

Az adatbázisba azért töltöttünk fel adatokat, hogy abból később rendezetten, valamilyen szűrők alapján információkat nyerjünk. A SELECT utasítás ezt segíti, melynek szintaxisa a következő:

```
SELECT * FROM oszk_maps WHERE pubplace LIKE "%Buda%" AND
author="Kogutowicz Manó" ORDER BY title ASC
```

A SELECT után következik, hogy milyen mezők adatait szeretnénk vizsgálni az eredménytáblában, a csillag az adott tábla összes mezőjét kiírja, ha szűkíteni szeretnénk az eredménytáblán, egymástól vesszővel elválasztva sorolhatjuk fel az egyes attribútum-osztályokat. Az „AS *név*”, vagyis az alias név alkalmazásával átnevezhetjük az eredménytábla fejlécében található mezőneveket is. Ennek főleg a túl bonyolult oszlopnevezések esetében van szerepe, vagy ha az egyik mező tartalmát módosítjuk egy függvénnyel, ekkor a mező neve is megváltozik. Pl. AsText(poly) as poly. Ekkor a geometry típusú adatok átkonvertáljuk szöveges adattá, miközben a mező neve változatlan marad, amely segíthet nekünk egyszerűsíteni a PHP-ban az adatok kiíratását. A FROM megadja, hogy melyik táblában lévő adatokat vizsgálja. A WHERE után szűrő feltételeket következnek azért, hogy szűkítsük a találatok számát. Több feltétel összekapcsolásához és egyidejű figyelembe vételéhez az AND, vagyis az „és” logikai operátor használandó. Ha tökéletes egyezést keresünk az adattáblában, akkor a megfelelő mezőnév kívánt értékét kell beírni, pl. author="Kogutowicz Manó". Ha hasonló értékek után kutatunk, akkor a „mezőnév LIKE string-et” használjuk, pl. WHERE pubplace LIKE "%Buda%". A „%” tetszőleges számú karaktert helyettesít. A honlapon a

keresőnél nem törekszem tökéletes egyezésre keresésére, mert a felhasználó esetleg nem ismeri a teljes nevet, így ha elégel, vagy csak részleges ír be szavakat, minden olyan találat megjelenik, amelyik tartalmazza azokat a részleteket, így könnyebben válogathat a találatok között.

Az ORDER BY a rendezés utasítása, az ASC növekvő sorrendet, a DESC pedig csökkenő sorrendet készít.

A következő példa a táblák összekapcsolására vonatkozik. A szerzők azonosítója egyezzen a térkép szerzőjével, ez esetben ez egy szám, eredményként kiíratjuk az összes térkép címét és a térkép szerzőjének nevét.

```
SELECT title, au_author FROM oszk_maps, oszk_authors WHERE
oszk_authors.authorid = oszk_maps.author
```

Az előző fejezetben részletesebben írtam már a Spatial Extensions-ről, most ennek a gyakorlati alkalmazását mutatom be a weboldal példáján. A Spatial Extensions-t használva a feltöltési és módosítási parancsok kissé módosulnak, ugyancsak szükség lesz konvertáló függvény meghívására, amely szöveges adatból (text) geometry típusú adatot állít elő. Ez a függvény a `GeomFromText()`. Lekérdezéskor a geometry típusú adatot kell „olvashatóvá” tennünk, ezt az `AsText()` függvény garantálja.

```
SELECT title, AsText(poly) as poly FROM oszk_maps WHERE
poly=GeomFromText('POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))')
```

Ebben az esetben azokra a polygonokra vagyunk kíváncsiak, amelynek a koordinátái a fentebb felsoroltak. A következő példában azokat a polygonokat keresem, amelyek tartalmazzák az x=0,5, y=0,5 pontot.

```
SELECT title, AsText(poly) as poly FROM oszk_maps WHERE
CONTAINS(poly,POINT(0.5,0.5))
```

Az eddig említett példák a honlapon is szerepelnek majd. A következő fejezetben az oldalt működtető szkriptnyelvet mutatom be.

## 4. A PHP szkriptnyelv

„A PHP (Hypertext Preprocessor) nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerver-oldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele. A nyelvet eredetileg Rasmus Lerdorf alkotta meg 1994-ben, de a ma létező egyetlen (és hivatalos specifikáció híján de facto szabvánnyá vált). PHP implementációt már a PHP Group tartja karban és fejleszti. A PHP a legtöbb webszerverre, operációs rendszerre és platformra ingyenesen telepíthető. A PHP emellett az Apache webszerver egyik legnépszerűbb beépülő modulja.” (Wikipédia-PHP, 2012)

Azért választottam a PHP-t, mert egy olyan szerveroldali programozási nyelv, amely amellet, hogy a MySQL-el jól „együttműködik”, a nyelv szintaxisa egyszerű, könnyen tanulható és programozható, a beépített függvények segítik a fejlesztést, továbbá egyszerű szövegszerkesztőben tudunk programozni. A kód futtatása mindig egy böngészőben történt, erre a célra a *Mozilla Firefox*-ot találtam a legalkalmasabbnak.

### 4.1 A PHP használata

A PHP kódot a felhasználó nem látja, hiszen a szerveroldalon fut. Kimenete viszont HTML oldal. A PHP kódot tartalmazó fájlok kiterjesztése „.php”, maga a kód a `<?php ?>` tag-ek közé kerül. Programom első és legfontosabb eleme az adatbázishoz való kapcsolódás, amelyet minden PHP fájlban végrehajtok. Először is meghívom a kapcsolódási-függvényt, melynek argumentumaiba a szerver neve (ez esetben localhost, vagyis a webszerver maga a kiszolgáló), majd felhasználónév és jelszó kerül.

```
$db=mysql_connect($servername,$username,$pwd) or die ("Nem sikerült a kapcsolódás.".mysql_error());
```

A következő függvénnyel definiálom, melyik adatbázison futtatom a lekérdezéseket és, hogy ez melyik connection-re vonatkozik. Végül a karakterkódolást állítom be. Ha mindezek közben hiba lépne fel, kiíratom a hibaüzenetet.

```
mysql_select_db($db_name, $db) or die('Error while selecting db');  
mysql_query('set names latin2');
```

Mivel az adatbázis kapcsolat felállítás minden egyes fájlban meghívódik, ezért célszerű egy külön fájlban definiálni a változókat, úgymint felhasználónév, jelszó, a szerver és az adatbázis neve, azért ha változás történne, egy helyen kelljen csak módosítani. A külső változókat tartalmazó fájlt az `INCLUDE 'config.php';` paranccsal olvastatom be.

A következő lépés az adatok lekérdezése. A változó tartalmazni fogja az eredménytáblát. A `mysql_query()` függvénynek két paramétere van, magának a lekérdezésnek a szövege és melyik kapcsolatra vonatkozik. A lekérdezés szövege egy szöveges változó tartalmaként is elküldhető.

```
$lekerdezes=mysql_query($query, $db);
```

Az eredményt egy asszociációs tömbbé kell alakítani a `mysql_fetch_assoc($lekerdezes)` segítségével ahhoz, hogy utána kinyerhessünk belőle adatokat. Korántsem biztos, hogy a lekérdezésünknek lesz eredménye, ekkor érdemes a `mysql_num_rows()` függvényt használni, amely megszámolja az eredménytábla sorait, ha ez zérus, akkor egy kiíratási paranccsal kiírjuk, hogy nincs találat. A kiíratáshoz két utasítást lehet használni: `echo` vagy `print`.

Az adatok, mivel maguk is egy táblázatból érkeznek célszerű egy táblázatba feltölteni. Ekkor egy iterációt kell létrehozunk, melynek a kilépési feltétele az, hogy végigolvasta és ki is írta a sorokat.

A keresés során a user-nek kell eldönteni, mit és hol szeretne keresni, vagyis milyen szöveges információt melyik mezőben, ehhez egy textbox-ba lehet beírni a keresendő kifejezést, és egy legördülős listából választhatjuk ki az aktuális mezőt. A lista, a `SELECT` tag a HTML form egyik eleme, melynek működése a következő: az egyes lehetőségeket `OPTION` tag-ek közé írjuk, és a „value” tárolja az értéket pl. `<option value=oszk_maps.title>Térkép címe</option>`. Erre az aktuális értékre lesz szükségünk.

Ezeket az adatokat beolvassuk, a tartalmukat egy változóba töltjük, amely aztán a lekérdezésbe is belekerül pl. `$hol=$_GET['lista'];`. A form-ok feldolgozásához alapvetően két metódust lehet használni. Az egyik ezek közül a GET. A böngésző címsorában küldi el az információkat az oldal címéhez hozzáfűzve. A küldött szöveg mennyisége 1024 karakterben korlátozott, ezért több, hosszabb változó küldésére alkalmatlan, valamint bináris adatokat sem tud továbbítani. Ezzel a módszerrel nem szabad olyan érzékeny információkat elküldeni, mint a jelszó, arra a POST-ot ajánlják, ami a HTTP header-ben küldi el a kódolt információt, így azok nem láthatóak, és nem

hozzáférhető. Mint mondtam a jelszót beírásakor sem szabad láttatni, ezért egy sima text input helyett a password inputra váltottam, amely a begépelte karaktereket pontokkal helyettesíti.

Az adatok feltöltése, törlése illetve módosítása szintén a `mysql_query` paranccsal érhető el, a küldendő lekérdezés változik meg. Feltöltéskor a legegyszerűbb módja, hogy megbizonyosodjunk arról, hogy feltöltődött-e az adott rekord, hogy a feltöltő parancs meghívása előtt és után is megszámoljuk az adatbázis összes sorát, és ha egyel növekedett, akkor sikerült. Törléskor ennek ellenkezője érvényesül.

Mielőtt rátérnék a fájlfeltöltésre és a session-kezelésre, előbb pár szót ejtenék a HTML form-okról, amivel már fentebb foglalkoztam. (W3C, 2012). A DOM (Document Object Model) szerint a form-ok a HTML dokumentum részei, amelyek különleges elemeket, control-okat tartalmaznak, amelyekkel interaktívvá tehetjük a weboldalt. A form-on található input elemek a következők:

- Text: szöveg beviteléhez használjuk. A `id` vagy a `name` attribútumával egyedi azonosítót rendelhetek hozzá, melyre később hivatkozhatok egy beolvasás során. A `value`, vagyis az „érték” attribútumot csak a módosító oldalon használom, mivel ezzel a tulajdonsággal megadható, milyen szöveges információ kerüljön bele.
- Password: Jelszó titkosításához használjuk, `name` értékkel azonosítható válik adatfeldolgozáshoz.
- Submit: Egy speciális gomb, melynek megnyomásával elfogadjuk az adott form-on lévő elemeket és elküldjük feldolgozásra. A feldolgozás metódusát a `form` tag `method` elemével állíthatjuk be, az `action` pedig a kezelendő fájlt írja le. Ez lehet önmaga (`$PHP_SELF`), vagy egy külső fájl is. A `value` attribútum tartalmazza, mi legyen a gomb felirata.
- Button: Egy adatfeldolgozásra utasítást adó bemeneti elem, előre definiált viselkedés nélkül, ezért az `onclick` eseménnyel határozzuk meg, hogy mit hajtson végre. A `value` attribútum tartalmazza, mi legyen a gomb felirata.
- A `select` tag, vagy más néven legördülős lista, nem az input elemek közé tartozik ugyan, de a form része. Az `option` tag-ek közé irandó a megjelenítendő szöveg, a `value` a feldolgozás értékét küldi el.



- A fájlfeltöltéshez két elem szükséges. A file select és file input. A file select akkor jön létre, ha egy file input-ot is létrehoztunk. A kettővel együtt tudjuk kiválasztani a gépünkről a feltöltendő fájlt.

A fájlfeltöltéshez ezt a file input-ot használtam. A form feldolgozási módszere ilyenkor csak POST lehet. A PHP-ban fájlkezeléshez kapcsolódó függvények is be vannak építve. A fájlok tömbként viselkednek. Mielőtt a fájl a végleges helyére kerülne a szerveren általában egy ideiglenes mappába vándorol, ha megérkezett az összes eleme, akkor a `move_uploaded_file()` függvénnyel áthelyezhetjük a célmappába. Ekkor érdemes megvizsgálni, létezik-e már ilyen nevű fájl az adott mappában a `file_exists()` függvénnyel, mivel véletlenül felülírhatjuk. Ha létezik, akkor a fájlnevet egy számmal módosítom. Itt figyelni kell arra, hogy az adatbázisba beírt név egyezzen a feltöltött fájl nevével. Másik fontos dolog a célmappának is be kell állítani a jogosultságait. Ezt pl. WinScp-vel tehetjük meg. A jogosultságot „`rw-rw-rw-`”-re (0777) kell állítani, vagyis a mappa tulajdonosa, egy meghatározott csoport továbbá mindenki más is módosításokat hajthasson végre a mappában. Más esetben letiltódik a fájlmozgatás parancs és nem kerül át az ideiglenes mappából a célmappába. A `php.ini` fájlban

Új térkép hozzáadása

Cím:

Szerző:

Kiadó:

Kiadás helye:  (Város, Ország)

Kiadás ideje:

Méretaránya: (1 : x)  Csak a tört nevezőjét kell megadni!

Vetülete:

Leírás:

Jelzete:

Kulcsszavak:  Kulcsszavak szóközzel elválasztva.

Hasonló térképek:  Felsorolásakor az elválasztó karakter a ";".

Kmz fájl:  Fájlnev kiterjesztés nélkül!

Kép fájl:  Fájlnev kiterjesztéssel együtt!

Google Maps fájl:  Kiterjesztés nélküli fájlnev!

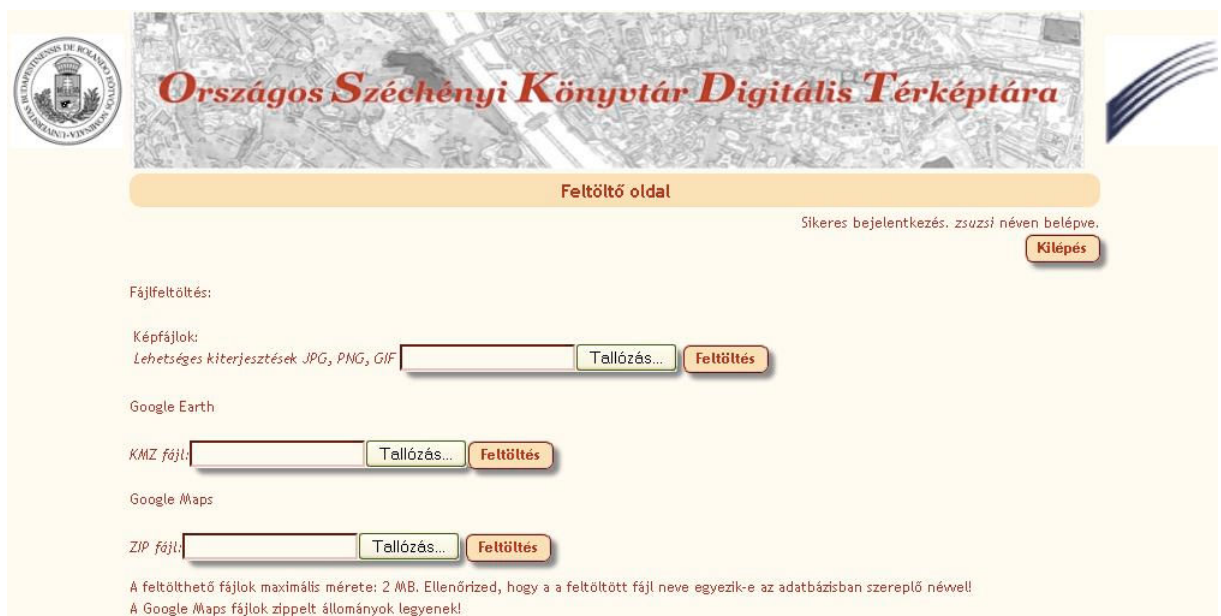
Polygon koordinátái:  Ellenőrizd, hogy kitöltötté-e minden mezőt!

5. ábra: Az adatlap feltöltése

definiáltak szerint, (ebben lehet megadni a PHP alapbeállításait) 2 MB-nál nagyobb méretű fájl erre szerverre nem tölthető fel.

A fájlfeltöltés előtt még egy dolgot érdemes megvizsgálni. Erre nem a PHP, hanem a JavaScript nyelvet használom. Mivel külön mappába kerülnek a képfájlok és a KMZ fájlok, ezért egy szkript függvénnyel beállítom a lehetséges kiterjesztéseket, amelyre engedélyezett lesz a feltöltés. A függvény egy button-onclick eseményére hívódik meg, egyezés esetén elindul a feltöltés. (Bár a következő függvény JavaScript-ben van megírva, de tartalmilag ehhez a fejezethez tartozik.)

```
Function vizsgalExt(f)
{
var ext=f.newfile.value.substr(f.newfile.value.length-4);
if (ext!=".jpg"&&ext!=".gif"&&ext!=".png")
{alert("Csak .GIF, .JPG, vagy .PNG fájlok jöhetnek!");}
else f.submit();
}
```



**6. ábra: A különféle fájlfeltöltési lehetőségek**

A PHP segítségével oldottam meg a Google Maps-ben megtekinthető fájlok feltöltését is. Mivel a *Global Mapper*-ben a Google Maps Tiles készítésekor a végtermék egy HTM oldal és az egyes tile-okat tartalmazó mappák. Ezeket egyenként hosszadalmas lenne feltölteni, ezért érdemes összetömöríteni a fájlokat egy ZIP állománnyá, melyet a PHP beépített függvényei segítségével ki tudok tömöríteni. A feltöltés után, (ebben az esetben a kiterjesztés csak „.zip” lehet) a kitömörítést végző

parancssorban először meg kell határozni a pontos elérési utat, hova szeretném feltölteni a fájlt. A mappa nevét a „zip” kiterjesztés levágásával nyerem. A mappa létrehozása előtt, meg kell vizsgálni, létezik-e már a mappa, ha igen, hibával visszatér, és újra kell kezdeni a feltöltést. A mappa jogosultságain a chmod() függvénnyel állíthatok.

```
$dirname=substr($filename,0,-4);
if(is_dir($dirname))
{echo "Már van ilyen nevű mappa!";}
else{mkdir($dirname);
chmod($dirname,0777);}
```

A következő lépés a kitömörítés. Első feladat megtudni azt, van-e olyan fájlom, amit kitömöríthetek, ha igen, a zip\_open()-nel megnyitom, majd a zip\_read()-el az archívumban található elemeket egyenként végigolvasom, és kimásolom az állományból. Végül bezárom az archívumot. Mivel az archívumra a továbbiakban már nincs szükségem, mert a létrejött mappa tartalmazza az egyes fájlokat, ezért az unlink('mappanéV') paranccsal eltávolítom a szerverről. A mappa védelme érdekében érdemes újra korlátozni a jogosultságokat.

```
if (file_exists("$file"))
{
$zip = zip_open($file);
if(is_resource($zip))
{while ($zip_entry = zip_read($zip))
{$fp=fopen("/data/home2/hal2007/ungvarizs/public_html/oszk/
gmaps/". $dirname."/".zip_entry_name($zip_entry), "w");
if (zip_entry_open($zip, $zip_entry, "r")) {
$buf =zip_entry_read($zip_entry,zip_entry_filesize($zip_entry));
fwrite($fp,"$buf");
zip_entry_close($zip_entry);
fclose($fp);}}}
zip_close($zip);
```

Mint mondtam, a honlapom két részből áll, az egyik a kereső, a másik a védett feltöltő oldal. A feltöltő oldalon szükség van a felhasználók bejelentkezésére, mivel csak az arra jogosultak férhetnek hozzá. A bejelentkezéshez a PHP Session kezelését alkalmazom.

**7. ábra: A bejelentkezés form-ja**

A bejelentkezéshez ki kell tölteni egy form-on a felhasználónév és a jelszó mezőket, majd elküldöm az adatokat feldolgozásra. Először lekérdezzük, hogy létezik-e ilyen felhasználó és helyes-e a jelszó. Ha igen, még az adatbázisban leírt jogosultságokat is meg kell vizsgálni, és ezek szerint továbbirányítani a megfelelő oldalra. Ezután a felhasználónév alapján elindítom a sessiont a `session_start()`

**8. ábra: A bejelentkezés utáni lehetőségek adminisztrátoroknak**

függvénnyel, majd regisztrálok a `session_register()`-rel, így létrejön egy globális változó. A `session_register()`-t használtam akkor is, ha egy változót át akartam adni két PHP oldal között. Ha törölni akarom a változót, a `session_unregister()`-t hívom meg. A kijelentkezéshez a változó törlése nem elég. A session-t a `session_destroy()`

függvénnyel lehet lezárni, amely minden adatot és változót töröl, amelyet az adott session hozott létre.

A jelszavakat az adatbázisban nem érdemes string-ekként tárolni, mert könnyen megfejthetőek lesznek, ezért a `Password()` függvénnyel lehet titkosítani, melynek eredménye egy 41 karakterből álló sorozat lesz, amely gyakorlatilag visszafejthetetlen.

Ha bejelentkezett a felhasználó a szkriptfájl továbbirányítja a jogosultságának megfelelő oldalra. Ellenben ha olyan oldalt szeretnénk megtekinteni, amelyik nem felel meg a jogosultságnak, pl. ha legszigorúbb feltételek mellett begépelik a címsávba a jóváhagyó vagy a törlő oldal címét, az át fog lépni, mert csak a bejelentkezésnél választottam külön a felhasználókat. Ezért az olyan fájlknál, pl. jóváhagyás és törlés, aminél nem szeretném, hogy minden felhasználó lássa az oldalakat, bele kell tenni egy jogosultság ellenőrzést, amely a következőképpen néz ki: lekérdezem, az adatbázisból a felhasználó jogosultságait, és egy if elágazást használok az elkülönítésükre.

## 5. A JavaScript nyelv

„A JavaScript programozási nyelv egy objektumalapú szkriptnyelv, amelyet weblapokon elterjedten használnak. Eredetileg Brendan Eich, a Netscape Communications mérnöke fejlesztette ki, neve először Mocha, majd LiveScript volt, később „JavaScript” nevet kapott, és szintaxisa közelebb került a Sun Microsystems Java programozási nyelvéhez. A JavaScriptet először 1997–99 között szabványosította az ECMA „ECMAScript” néven (Wikipédia, 2012).”

A JavaScript kliens oldali szkriptnyelvet több függvény esetében is felhasználtam. A fő ok, amiért mindenképpen alkalmazni kellett, mert a Google Maps és Earth, valamint az Open Layers is szkriptként épül be a HTML oldalba.

Mivel kliensoldali szkriptnyelv, ezért a felhasználó is láthatja. Előbb már említettem, hogy a HTML oldalba is beilleszthetem script tag-ek közé, vagy akár egy külső fájlként is letölthető, amelyre a HTML dokumentumban hivatkozunk, pl. az OpenLayers vagy a Google szkriptfájljai. A következő alfejezetben a Google-t, mint beépülő modult és alkalmazását mutatom be.

### 5.1 A Google Maps

„A Google Maps a Google által fejlesztett ingyenes internetes térképszolgáltatás. Az elérhető térképek és műholdfelvételek az egész Földet lefedik. A többi Google internetes alkalmazáshoz hasonlóan nagy mennyiségű JavaScriptre épül. A térkép mozgatása során a kis térképrészletek folyamatosan töltődnek le a szerverről, illetve jelennek meg a böngészőablakban. A Google Maps-ben szabadon mozgatható és nagyítható térképeket találunk. A megjelenített részletek a nagyítástól függően folyamatosan változnak. A Google Maps nyílt API-jának következtében kiválóan alkalmas arra, hogy bárki bármilyen adatbázist illesszen fölé.” (Wikipédia, 2012).

A Google Maps beágyazásával az a célom, hogy az általam létrehozott adatbázisban található adatokat, koordinátákat kereshetővé tegyem, illetve egy földrajzi keretet teremtsék a keresőnek. A beágyazáshoz szükséges kód a Google Maps Code oldalán elérhető, ennek segítségével dolgoztam ki a funkciókat.

A beágyazáshoz az első lépés a Google API kulcs letöltése és a külső Javascript fájl lekérése a HTML header-jébe ágyazott szkriptként.

```

<script type="text/javascript"
src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sen
sor= FALSE">
</script>

```

A következő lépés az inicializációs függvény megalkotása. Ebben a függvényben írjuk le azt, milyen kezdeti beállításokkal töltődjön be az API. Itt kiválasztom a térkép nézetét, a meghívható rétegeket (utcatérkép, műholdkép, vagy hibrid).

A Google Maps egyik szolgáltatása többek között a Geokódoló, amellyel valós címekhez rendelhetünk földrajzi koordinátákat. Nálam ez a kereső miatt fontos. Az inicializálási függvény után ezt is definiálnom kell. A függvény bemeneti paramétere maga a cím és a geokódoló státusza, amely a működési hibák feltárásában segíthet. Általában az eredménytömb több értékkel tér vissza, ebből az elsőt választom ki és dolgozom fel. A tömb egyes elemei tartalmazzák az objektum típusát, a formázott címet és a geometriát. A geometriából több adatot is ki lehet nyerni, számomra a legfontosabb a földrajzi koordináták megszerzése. Miután ez megvan, a koordinátákat beleteszem az oldal címsorába, aztán a PHP szkript onnan tudja kiszedni az oldal újratöltése után.

```

function geocode(addr)
{
geocoder.geocode( { address: addr }, function(results, status) {
if (status == google.maps.GeocoderStatus.OK) {
map.setCenter(results[0].geometry.location);
marker = new google.maps.Marker({
map: map,
position: results[0].geometry.location,
title: addr
});
x=results[0].geometry.location.lat();
y=results[0].geometry.location.lng();
alert("A következő koordináták alapján keresünk az adatbázisban:
fi: "+x+", lambda: "+y);
window.location.href="index.php?f=" + x +"&l=" +y;
} else {
alert("A geokódolás nem sikerült. Írj be egy valós címet! A hiba
oka: " + status);
}});

```

A geokódoló függvény hiába van definiálva, anélkül, hogy hozzáadtuk volna a térképhez, nem működik.

```
geocoder = new google.maps.Geocoder();
```

Továbbá a térképet egy div-ben kell beágyazni a HTML dokumentumba, különben nem látszik.

```
<div id="map_canvas" style="width:700px; height:500px"></div>
```

A másik koordinátás keresési lehetőség az, hogy kattintunk a térképen. Ekkor ennek a pontnak a koordinátáit szeretnénk behelyezni a címsorba, majd újratöltődés után, ugyanazt a PHP kódot futtatom le, amelyet az előbb. A „klick” esemény hatására meghatározzuk a koordinátákat, melyek aztán a címmezőbe kerülnek.

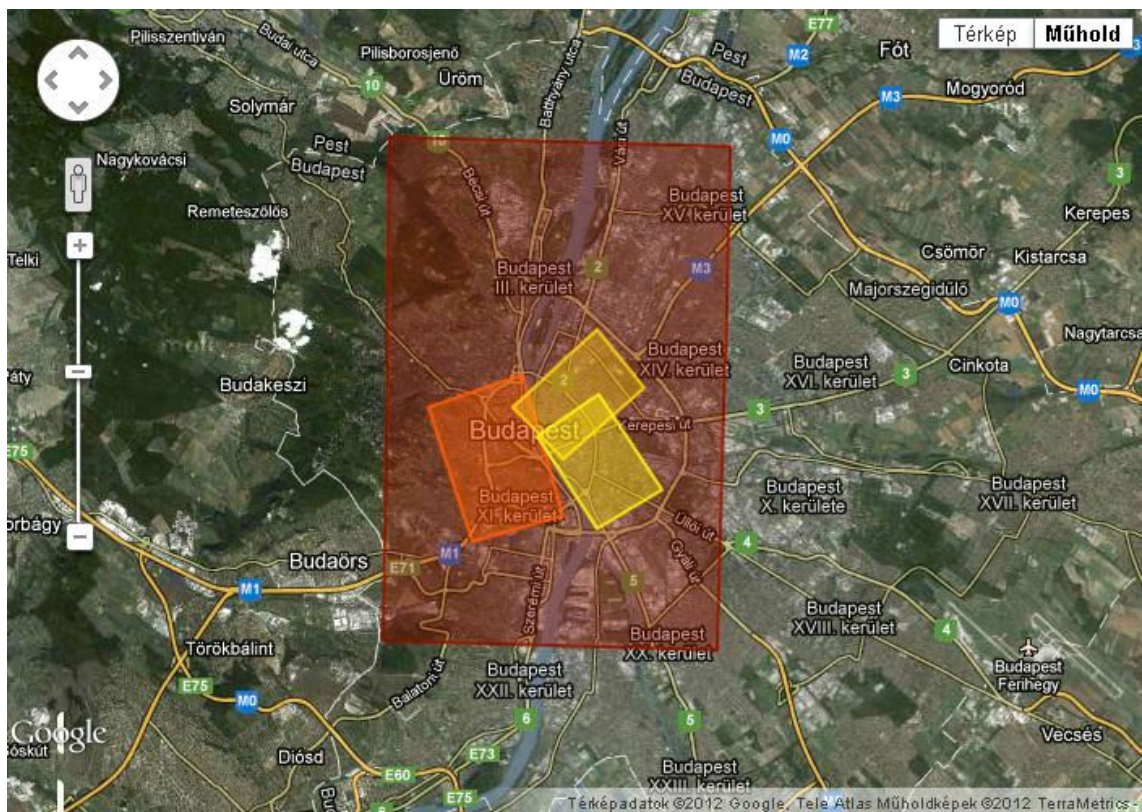
```
google.maps.event.addListener(map, 'click', function(event) {  
  with(event.latLng)  
  {  
    fi=lat();  
    la=lng();  
    alert("A következő koordináták alapján keresünk az adatbázisban:  
    fi="+fi+"° lambda="+la+"°");  
    window.location.href="index.php?f=" + fi +"&l=" +la;  
  } });  
}
```

A harmadik lehetőség beírni a koordinátákat egy input box-ba. Ezt a funkciót lehet használni a Google alaptérkép nélkül is, hiszen mi adjuk meg a koordinátákat. Itt egy olyan lehetőséget is bevezettem, hogy ne csak pont lehessen a tizedes elválasztó jele, hanem vessző is, mivel a magyarok inkább a vesszőt használják, a tizedespontot csak az informatikai rendszerekben terjedt el. Vagyis a koordináták megadása kétféleképpen is történhet: 47.5 és 47,5. A függvény úgy van megírva, ha az elválasztó karakter vessző, akkor kicseréli egy pontra, amely már megfelel az adatbázisban szereplő karakternek. Erre a JavaScriptes `split()` és `join()` függvényeket használtam fel.

```
function koord (keres)  
{  
  var koordinatax;  
  koordx=keres.koordx.value;  
  koordinatax=koordx.split(",");  
  koordx=koordinatax.join(".");  
  window.location.href="index.php?f=" + koordx +"&l=" +koordy;  
}
```



Sokkal szemléletesebb, ha megjelenik a Google Maps-es térképen is a keresés eredménye, vagyis az egyes térképek által lefedett területek színes polygonok formájában. A MySQL táblám többek között ezért is tartalmazza a térképek sarokpontjainak koordinátáit. Ezeket kell átalakítanom a Google Maps API számára is olvasható formára. Ezeket string-eket a PHP-val gyártatom le és íratom be a JavaScript inicializációs függvényébe. A string-ek összeállításához az `explode` (szöveget szétbont), `implode` (szöveget hozzáragaszt) és `array_push` (az adott elemeket egy tömbbe rendezi) PHP függvényeket használtam. A polygon definíciója leírja a helyét és küllemét. Egyébként ahhoz, hogy minden polygon más színű legyen egy külső PHP fájlban létrehoztam egy tömböt, amely HTML színekódokat tartalmaz. A tömb legyártásakor, amely az összes, térképen megjelenítendő polygont tartalmazza, egy számláló, amely önmagát növeli minden ciklus lefutásban, válassza ki a színeket tartalmazó tömb soron következő elemét.



**9. ábra: A térképlapok által lefedett területek**

```
var Coords1=[new google.maps.LatLng(47.58155278,18.98367778)
... /polygon töréspontjainak koordinátái/];
Coords1 = new google.maps.Polygon({
paths: Coords1, strokeColor: "#8B0000", strokeOpacity:0.8,
strokeWeight: 2, fillColor: "#8B0000", fillOpacity: 0.35});
```

```
Coords1.setMap(map);
```

Ahhoz, hogy a térkép adatlapját a polygonra való kattintással is el lehessen érni, először meg kell nyitni egy információs buborékot, amely tartalmazza a térkép címét, amely a részletes adatlapra irányít tovább.

```
var hely1=new google.maps.LatLng(47.58155278,18.98367778);var
content1="A térkép címe: <a href=rezsletes.php?id=1
target=_blank> Térkép címe
</a>";google.maps.event.addListener(Coords1, "click",
function(e) { var infobox = new
google.maps.InfoWindow({position: hely1, map: map, content:
content1}); });
```

A koordináták megadása a feltöltés során többféleképpen is lehetséges. Az új térképi adatlap létrehozása során a koordinátákat csak kézzel lehet beírni, amelyet azonban könnyű elrontani, a MySQL nem tolerálja a hibákat, pontosan ezért hoztam létre egy segédletet. Ezt a funkciót viszont csak abban az esetben lehet meghívni, ha már létezik a térkép adatlapja, és feltöltötték a KMZ fájlt is. A Google Maps-ben betöltődik a KMZ fájl, a Drawing Tools segítségével körberajzolható a térkép. A Google szerveréről ez esetben a kód fejlécében a Drawing Library eszközeit is le kell kérni.

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&libraries=
drawing">
```

Ezek után már lehetséges meghívni a függvénykönyvtárat, amelyben a polygon rajzoló eszközt definiálom, az ikonok helyzetét, ebben az esetben felülre, középre kerül. Ezen kívül a polygon kinézetét is meg lehet adni, amelynek módját már korábban leírtam. Mint minden eszközt, ezt is hozzá kell csatolni a térképhez a setMap() segítségével.

```
var drawingManager = new google.maps.drawing.DrawingManager({
drawingControl: true,
drawingControlOptions: {
position: google.maps.ControlPosition.TOP_CENTER,
drawingModes: [
google.maps.drawing.OverlayType.POLYGON, ]},
polygonOptions: {...}
});
```

```
drawingManager.setMap(map);
```



**10. ábra: A polygon megrajzolása**

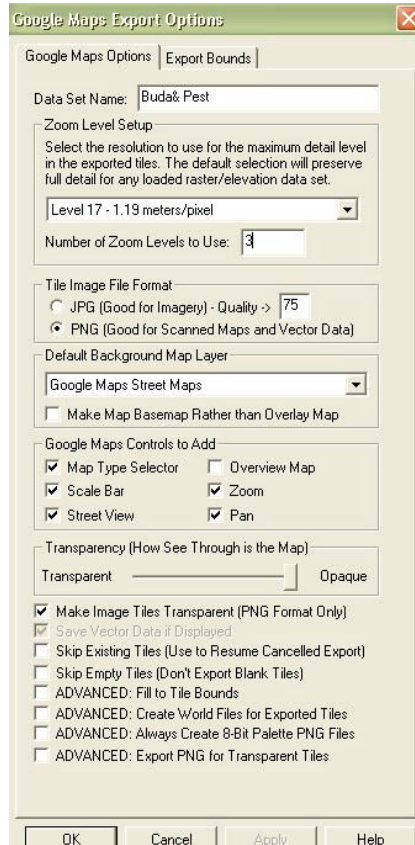
Ha befejeztük a polygon rajzolását, ennek sarokkoordinátáit fel kell dolgoznunk, és az adatbázisba írunk. Az alakzatok rajzolásának van egy „overlaycomplete” eseménye, amely akkor hívódik meg, amikor dupla kattintással befejeztünk a rajzolást.

Ekkor a koordinátákat kérdezem le a `getPath()` függvénnyel, amely magában egy tömb, ezért a `getArray`-jel olvasható szöveges formátumra tudjuk hozni.

```
google.maps.event.addListener(drawingManager, 'overlaycomplete',  
function(event) {  
if (event.type == google.maps.drawing.OverlayType.POLYGON) {  
sarkok = event.overlay.getPath().getArray(); }  
}
```

Az adatok feldolgozása és adatbázisban történő változtatása előtt, még egy megerősítést is kér a program.

A térképeket a Google Maps-ben is szeretném megjeleníteni. A térképeket külső fájlból töltöm be, és mint tematikus fedvény definiálom őket. Az adott HTML oldalt és a mozaikot tartalmazó könyvtárakat a *Global Mapper 13*-as verziójával legyártatom. A legyártott HTML oldal kódjából a fájl elérési útját relatív hivatkozásra változtatom, hogy a későbbiekben más szerveren való használat miatt ne okozzon ez problémát.



*11. ábra: A Google Maps Tiles készítése*

Mivel OpenLayers és OpenStreetMap használatával is megoldottam ugyanezt a problémát a fentebb említett okok miatt, ezért a következő fejezetben az eltéréseket bemutatására helyezem a hangsúlyt.

## 5.2 Az OpenLayers és az OpenStreetMap

„Az OpenLayers egy nyílt forráskódú kliens oldali JavaScript függvénykönyvtár, melynek segítségével különféle forrásokból származó térképi adatok jeleníthetők meg interaktív térképként weboldalakon.” (Gede, 2012) Forrástérképnek az OpenStreetMap szintén ingyenes térképét választottam. Az API elkészíthető úgy is, hogy a több térképet is definiálunk, és egy menü segítségével a felhasználó választhatja ki, melyik térképet használja. Beágyazható például még a Bing Maps, vagy a Google Maps is.

A három fajta földrajzi kereső közül, az előbb utoljára bemutatott opciót, vagyis amikor a felhasználó viszi be egy input box-on keresztül a koordinátákat, mivel az nem igényli a térképes hátteret, ezért erre nem térek ki újra. A másik kettő viszont a Google Maps-nél igen. Ellenben a címkereső az OpenStreetMap-ben meghívható a térképi

felület beágyazása nélkül is. Létrehozok két függvényt, legyen a mutat() függvény feladata, hogy beolvassa az input mező tartalmát, továbbá létrehozzon egy szkriptet, ami kapcsolódik az OpenStreetMap geokódoló oldalához, és lekéri az eredményt. A másik függvény feladata, hogy megmutassa a címhez tartozó koordinátákat. Ezek a koordináták címsorba való helyezésével el is indulhat az adatbázisban való keresés.

```
<script>
function megmutat(eredmeny)
{
if (eredmeny[0])
alert('fi='+eredmeny[0].lat+', la='+eredmeny[0].lon);
else
alert('Érvénytelen cím.');
```

```
}
function mutat()
{
cim=document.getElementById('cim').value;
var s=document.createElement('script');
s.src='http://nominatim.openstreetmap.org/search?q='+cim+'&format=json&json_callback=megmutat';
document.body.appendChild(s);
}
</script>
<input type=text id="cim">
<input type=button value="Holvan?" onclick="mutat()">
```

A harmadik keresési módhoz viszont szükségem lesz egy beépülő térképre, ugyanis azon kell bejelölni, és megszerezni a koordinátákat. Ehhez, mint említettem már az OpenStreetMap térképét használom. Maga a térkép több vektoros réteg összessége, itt nem beszélhetünk raszteres, műholdképes rétegről. A rendszer működése hasonlít a Google Maps-széhez. Ebben az esetben nem szükséges külön kulcsot generálni, amely azonosítja a honlapot, mivel teljes mértékben ingyenes, mindössze a függvénykönyvtárat kell letölteni a HTML header-jében.

```
<script src="http://openlayers.org/api/OpenLayers.js"></script>
```

A szkript programkódja szintén a <script> tag-ek közé kerül. Az inicializációs függvényben definiálom a betöltendő térkép tulajdonságait, vagyis a vetületét és a kezdeti nézőpontját. Ezután következik a control-ok hozzáadása, ezek közé tartozik a „klick” esemény is. A kapott koordinátákat ismét a böngésző címsorába küldöm el, amit újratöltődés után a PHP szkript kinyer, és elindítja a keresést a MySQL adatbázisban.

A polygon megjelenítése a földrajzi keresés során itt is hasonlóan történt.

### 5.3 A Google Earth

„Az Earth Viewer 3D eredetileg egy fizetős program volt, amelyet megvásárolt a Google, ezután ingyenessé vált, és a neve is megváltozott. A program lényege, hogy egy háromdimenziós glóbuszon különféle földrajzi információkat jeleníthessünk meg. Ahhoz, hogy böngészőbe ágyazzuk be, le kell tölteni egy ingyenes Plug-In-t. A megjelenítés esetenként sokkal látványosabb lehet ebben a háromdimenziós környezetben, mint a sík térképen. A felhasznált műholdképek felbontása területtől függően változik, Magyarország esetén 2,5 méteres felbontású SPOT képeket használnak, Budapest és a néhány nagyobb város esetében nagyobb felbontás is elérhető, kb. 30 cm-es.” (Wikipédia, 2012) Ahogy a Google Maps-ben, itt is definiálhatók fedvények, akár saját külső fájlból is. Egyszerűen az inicializációs függvényben az alapbeállítások után meg kell adni a KML, vagy KMZ fájl forrását. A Google betölti a fedvényt, de ahhoz, hogy a gömb rá is álljon a betöltődés után, a nézőpontot is be kell állítani. A következő kóddal a KMZ fájlt nem kmlObject-ként írtam le, vagyis a későbbiekben nem manipulálhatom a fájlt, hanem hálózati adatforrásként, vagyis ebben az esetben csak olvasom a KMZ-t és nem lesz hozzáférésem továbbiakban a KML DOM objektumfához, de a mi esetünkben ez bőven elegendő. A nézőpontot a setFlyToView-hívja meg, ezzel a térkép kitölti a látóteret.

```
var networkLink = ge.createNetworkLink("");
networkLink.setDescription("NetworkLink open to fetched
content");
networkLink.setName("Open NetworkLink");
networkLink.setFlyToView(true);
var link = ge.createLink("");
link.setHref("ide jön az elérési út");
networkLink.setLink(link);
ge.getFeatures().appendChild(networkLink);
```

Az API-ra behívott térképek alapértelmezett esetben átlátszatlanok. Ahhoz, hogy a kitakart tartalommal, vagyis a műholdképpel össze tudjam hasonlítani a térképi tartalmat, egy átlátszóság állító opciót hoztam létre. Egy csúszka mozgatásával lehet változtatni az átlátszóságot: 100% a teljes láthatóságot jelenti, míg a 0%-nál a fedvény már nem látszik. A Google Code oldaláról letöltöttem azt a JavaScript függvénykönyvtárat

a header-ben, amely ennek a csúszkának a viselkedését leírja, valamint egy CSS fájlt, amely a csúszka megjelenését biztosítja.

```
<script src="simple-slider.js" type="text/javascript"></script>
<link href="simple-slider.css" type="text/css" rel="stylesheet">
```

A következő kódrészlet pedig az API-ban való működését és a HTML oldalhoz hozzáadását végzi.

```
var slider=new SimpleSlider(document.getElementById('slider-
container'), {
onSlide: function(pos) {
if (folder)
folder.setOpacity(pos);
},
max: 1.0,
startPosition: 1.0,
formatPosFn: function(pos) { return Math.floor(100 * pos) + '%';
}});
ge.getFeatures().appendChild(folder);
```



**12. ábra: A csúszka két állapota**

Végül meghatározom, hogy a betöltött objektumra vonatkozzanak az átlátszósági feltételek.

```
folder=kmlObject;
```

A KML és KMZ fájlok használatát „A térképek feldolgozása” c. fejezetben részletesen leírom.

A Google Earth-ös térképen a mai állapotokkal való összehasonlítás végett megjelenítem a határokat és az utakat egyaránt, amelyek láthatóságát gombokkal lehet ki- és bekapcsolni.

## 6. A honlap megjelenése

Manapság nem elég az, hogy egy honlap funkciói hibátlanul működnek, hanem valamilyen dizájnt is illik hozzárendelni az oldalhoz, amely tetszetősebb megjelenést kölcsönöz a hagyományos HTML dokumentumhoz képest. A megjelenést a Cascading Style Sheets stíluslapokkal definiálom egy külső fájlban. Másik lehetőség az lenne, hogy minden egyes oldal fejlécébe helyezem `<style>` tag-ek közé, de ennek a módosítása szintén nehézkes lenne, hasonlóan a jelszavakat tartalmazó PHP szkript fájlhoz. Így hát marad a külső fájl, amire az egyes oldalak fejlécében a következőképp hivatkozom:

```
<LINK href="style.css" rel="stylesheet" type="text/css">
```

A CSS jelenlegi szabványa a harmadik verzió. Ezt használom fel én is a honlap tervezésekor. Azonban azt tudni kell, hogy nem minden böngésző támogatja teljes körűen a legújabb stíuselemeket. A fejlesztés *Mozilla Firefox*-ban végeztem, így a stíuselemek is arra optimalizáltak. Az *Internet Explorer* 8-as kiadását teszteltem, néhány stíuselemet azonban nem tud megjeleníteni, pl. lekerekített sarkú input elemek, árnyékok, hover effektus input elemeken.

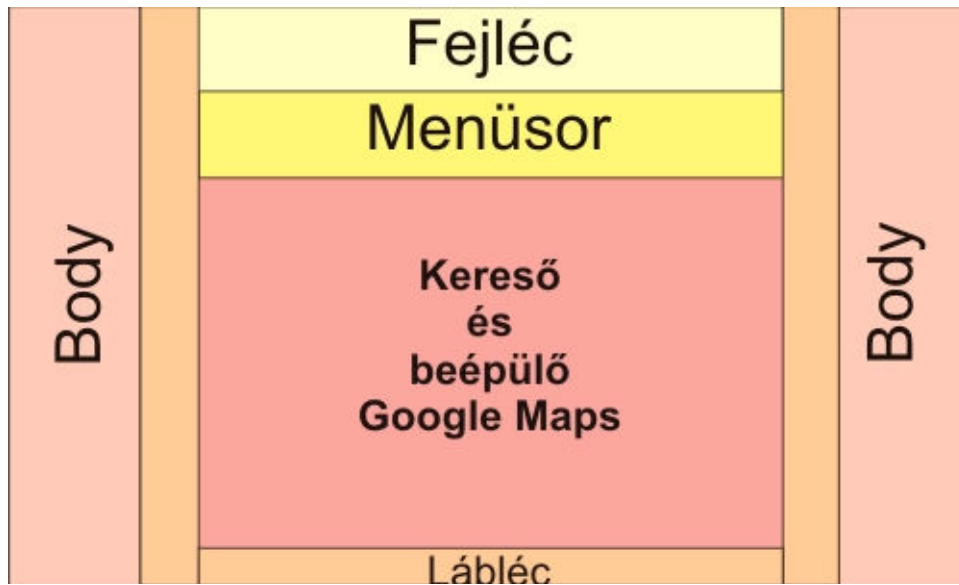
A CSS3 szintaktikája megegyezik a korábbi verzióval, de további stíuselemeket építettek be. Ilyenek például az árnyékhatások, vagy a lekerekített sarkú elemek. A nyelv alapja, hogy az egyes elemek tulajdonságát egy kapcsos zárójelben soroljuk fel, egymástól pontosvesszővel elválasztva. Az egyes szelektoroknak, mint pl. `body`, a `table` definiálhatunk megjelenést, de ennél jobban is differenciálhatunk az egyes elemek között. A honlapomon erre jó példa, hogy a menüben lévő linkeket és a táblázatban lévő linkeket különválasztottam betűméret alapján. Definiálni lehet az egyes elemeket osztály (`class="cim"`) alapján vagy ID (`id=menu`) alapján. Ennek megfelelően az előbbire a CSS fájlban „,cim”-mel hivatkozunk, míg az utóbbira „,#menu”-vel.

A következő példa a `body` elem stílusdefinícióját mutatja be.

```
body
{
background-color:#EEE8AA;
font-family: Trebuchet MS;
color: #A52A2A;
}
```



A stílusdefiníció megadása azonban nem elég. Mint mondtam, az egyes elemek között lehet id-vel, vagy névvel differenciálni is. Az oldal elrendezését táblázattal oldom meg, az egyes elemeket sorokba, oszlopokba rendezem. A kezdőoldal felosztását a következő ábra demonstrálja.



*13. ábra: A kezdőlap felépítése*

## 7. A térképek

### 7.1 A térképek feldolgozása

A térképtár akkor lesz teljes, ha megfelelő tartalommal népesítjük be. Mint dolgozatom címe is mutatja, az Országos Széchényi Könyvtár Térképtárában található térképek digitális állományai számára készítettem ezt a keretrendszert. A célom az volt, hogy olyan régi térképeket mutathassunk be a modern technológia segítségével, amelyeket kényes dolog lenne minden látogató kezébe adni, a térképek koruk miatt sérülékenyebbek, a sok használat csak árthat állaguknak, viszont igény van a térképek megtekintésére. Fontos megemlíteni, hogy a térképtárba csak olyan térképek kerülhetnek fel, amelyekre már nem vonatkozik a szerzői jog védelme, a későbbi jogi viták elkerülése végett. Ez az (utolsó) szerző halálától számított 70 év, vagyis a legkésőbbi térképek körülbelül a 20's évekből kerülhetnek még fel. A könyvtárból kezdetnek egy olyan dvd-t kaptam, amely tartalmazza a feltenni kívánt térképek egy részének nagyfelbontású szkennelt, digitalizált képeit.

Mivel ezek szkennelt TIFF állományok először ezeket a térképeket georeferálnom kell. A földrajzi koordinátarendszerben való elhelyezést és az azt követő digitális vetületi transzformációkat a *Global Mapper* nevű térinformatikai programmal végzem el.

Minden térkép esetén meghatározandó a vetület, az alapfelület és a koordinátaháló metszéspontjaiba felvett illesztőpontok. Ezután a program kirajzolja a számított koordinátahálót. A következő lépés a vetületi transzformáció a számunkra megfelelő rendszerbe. Az esetünkben két rendszer jöhet szóba: a négyzetes hengervetület (angolul: Geographic /Latitude, Longitude/ vagy Plate) és a Mercator-féle hengervetület (angolul: Mercator projection). A négyzetes hengervetületet a Google Earth fájl esetén alkalmazom, míg a Google Maps a Mercator-féle hengervetülettel dolgozik. Az OpenLayers esetén bármilyen vetület beállítható a vetület EPSG számának megadásával. Az egyetlen jó megoldás, ha a háttértérkép vetületének, hasonlóan a Google Maps-hez, a Mercator-féle vetületet választom.

A KMZ fájlokat már a Global Mapper 8-as verziójában is lehet készíteni, de ezek csak vektoros elemek exportjára képesek. A 12-es verzió alkalmas a képpel kiegészített tömörített KMZ fájlokat generálni. A KMZ fájlok a KML fájlok és a kapcsolódó képek tömörítésével jönnek létre. A KML fájlok olyan egyfajta XML

szöveges állományok, melyek leírják a megadott rétegek elhelyezkedését és megjelenését. Ezek a KMZ fájlok kerülnek réteggként rá a Google Earth műholdképeire. A fájlok legyártásakor egyébként automatikusan beállítja a megfelelő vetületet a Global Mapper.

A Google Maps esetén piramis szerkezetű fájlok készítünk, így a különböző zoomfokozatokhoz, más-más felbontású képeket rendel, ezek letöltődése csak akkor indul el, ha az adott zoomfokozatot hívjuk meg. A Global Mapper 13-es verziója alkalmas ezeknek a képfájloknak és a leíró HTM oldalnak az elkészítésére. Azonban a Global Mapper 12-es verziója a Google Maps Api v2-t gyártja le, melyet az idei év folyamán érvénytelenítettek, vagyis ennek a verzióknak a fejlesztése és javítása is leállt, az ez alapján működő oldalak v3-ra ajánlott átírni. (Google Maps API Documentation, 2012) Emellett az is probléma, hogy a v2-ben szükséges volt még egy ún. API kulcs használata is, amellyel a szerverük azonosított minket. A v3-ban ezt már nem kéri, ezért ezen tile-ok legyártásához a Global Mapper 13 a legalkalmasabb. A HTML oldalon a tile-okat tartalmazó abszolút hivatkozásokat érdemes megszüntetni, és relatívrá váltani. Hivatkozásként csak a mappát kell megadni a térkép adatlapjánál, amely tartalmazza a különböző zoomfokozatokhoz generált képeket, vagyis tile-ok mappáit és a HTM fájlt.

## 7.2 Az első térképek

A honlapra felkerülő első térképeket a már említett szkennelt állományok képezik. Ezekből néhányat kiválasztottam és elkészítettem. Ezek a térképek igencsak régiak, a 18-19. században készültek, olykor bármiféle vetületi alap nélkül, ezért feldolgozásuk nem egyszerű. Első lépés ilyenkor a térkép készítőjének és a kiadás időpontjának meghatározása. Ez esetenként már utalhat a kezdőmeridiánra és a vetületre is. Több közülük még budai kezdőmeridiánnal készült. Ebben a fejezetben a kiválasztott térképekről írok.

Az első térképsorozat, melyet kiválasztottam az 1800-as évek elején készült megyetérképekből áll. Görög Demeter és Kerekes Sámuel Megyei Atlaszához az első szelvények még a 17. század végén készültek el, de némelyik csak 1806-ban. A kiadási dátum sejteti, hogy rézmetszetről nyomtatták, a térképeket utólag, kézzel színezték, kb. 34 x 48 cm-s papírra készültek, és Bécsben jelentek meg. A térkép vetülete

feltételezhetően Mercator-féle szögtartó hengervetület, az elvégzett vetületanalízis alapján következtettem erre. Az alapfelület kérdéses: a térképek megjelenési ideje alapján következtethetünk csak rá. A szóba kerülő ellipszoidok a Zách-Oriani, Laplace, Walbeck és De Lambert-féle. Több kísérlet is született a térkép georeferálására, a legpontosabb alapfelületnek ezek közül a Walbeck-féle ellipszoid bizonyult. Kezdőmeridiánnak a gellérthegyi csillagdán átmenő délkört választották (Irmédi-Molnár, 1967), melynek pontos értékét a szakirodalomban is nehéz fellelni. (Varga, 2005) A Global Mapper-es dátumok és ellipszoidok megadásánál segítségemre voltak Mészáros János által írt átszámító programok. (Mészáros, 2009)

A másik sorozat egy nagyméretarányú telekredezési célból készített kataszteri térkép, amely a korabeli Budát és Pestet ábrázolja 1837-ben. A kiadvány címe Buda és Pest szabad királyi városainak tájleírása. A térkép alapfelülete és vetületi rendszere alig meghatározható utalás híján, ezen kívül semmiféle hálózatot nem tartalmaz. Ezért a georeferálás során a második katonai felmérés szelvényeit használtam, mivel ezek nemcsak időben, hanem méretarányukban is közelítik egymást, bár a II. katonai felmérés térképei kicsit újabbak, de a változások még nem olyan mértékűek a városszerkezetben, hogy hibát hozzanak a georeferálásnál. A kiadvány négy térképet tartalmaz, ebből az áttekintő kb. 1 : 36 000 méretarányú, a részlettérképek 1 : 10 000-szek.

A térképek feltöltésének menete a következő: a felhasználó a bejelentkezés után az Új gombra kattintva kitöltheti az adatlapot, majd kiválaszthatja a feldolgozott képet és Google állományokat. Ezután az adminisztrátornak az lesz a feladata, hogy ellenőrzi ezeket az adatokat, helyesek-e és láthatóvá válhatnak-e a keresőbe, ez a jóváhagyás folyamata. A térkép feltöltésekor a felhasználó neve is bekerül az adatbázisba, így követhetővé válik, ki hozta létre, vagy módosította utoljára az adatlapot. Hibák esetén módosításra és törlésre is lehetőség nyílik. A koordináták felvételét megkönnyítettem a már említett, polygon rajzoló eszközzel.

## 8. Kitekintés

A dolgozat célja, egy PHP-MySQL alapokon működő térképtár létrehozása volt, amelyben nemcsak keresőelemek, hanem a különféle megjelenítő-egységek is fontos szerepet játszanak.

A honlap tervezésekor figyelembe vettem, milyen ingyenes eszközök állnak rendelkezésemre, ezek alapján választottam. A tanszéki Mercator szerveren MySQL adatbázis fut. Azonban, ha a honlap átkerül egy másik szerverre, azon nem biztos, hogy ugyanezek a szoftverek lesznek telepítve. Ez ugyan további átalakítást eredményezne a programkódban, tehát kerülendő. Minimális átalakítással a MySQL-t tökéletesen helyettesítheti egy PostgreSQL. Az adatbázis kiválasztásánál figyelembe kell venni, hogy a PHP melyiket tudja kezelni. Mivel PHP-ban programoztam le a honlap működését, ezért egy más nyelvre való átírás már egy teljesen új honlapot eredményezne.

Az API-k tekintetében a Google Maps helyett már ki is dolgoztam egy lehetséges alternatívát, ez pedig az OpenLayers. Ennek oka az, többek között az is, hogy az OpenStreetMap térképei helyenként pontosabbak, mint a Google Maps, ezért egyre népszerűbbé vált a használata is. Így az weboldalam felhasználója szabadon választhat a két térképszolgáltató közül.

A honlap jövőbeli fejlesztése szükségszerűvé válhat, az informatika gyors fejlődése miatt. Először is a már említett okok miatt, érdemes lesz rátérni a MySQL 5.6 használatára, amelyben már nem a befoglaló téglalapokban keresünk, hanem magában a polygonban. Érdekes lehet egy Live Search kereső kialakítása is AJAX segítségével, melynél a begépelés pillanatában már láthatóak a találatok. Emellett összetett kereséssel is szűkíthető a találatok száma.

A térképtár kereső részét érdemes lehet több nyelven is megírni, hogy a magyarul nem beszélő személyek könnyebben használhassák a keresőfunkciókat.

## 9. Összefoglalás

Dolgozatomban létrehoztam egy olyan honlapot, amely képes lehet kiszolgálni az Országos Széchényi Könyvtár Térképtárának igényeit, és működhet, mint digitális térképtár. Munkámban részletesen bemutattam a program elkészítésének alapkérdéseit, a felmerülő problémákat, és megoldást is találtam rájuk.

Az első fejezetben olyan elvárásokat támasztottam, amelyek alapján elkészíthető egy komplex kereső, földrajzi kereső és megjelenítő, valamint különböző jogosultságokkal ellátott feltöltő rendszer.

A további fejezetekben az adatbázis felépítésére, és a lekérdezések futtatására tértem ki. Az adatbázisban szereplő adatokat a PHP szkriptnyelv segítségével jelenítettem meg egy HTML oldal keretein belül. A térbeli megjelenítésekhez a Google Maps és Earth, valamint alternatívaként az OpenLayers API-t hívtam segítségül.

A honlap küllemének megalkotásához a CSS stíluslapokat építettem be.

Végül kiválasztottam néhány régi térképet az 1800-as évekből, ezeknek a példáján demonstráltam, teszteltem a honlap működését.

A munkám célja, egy könnyen kezelhető és felhasználóbarát térképtár létrehozása volt, amelyben a kereső funkciók mellett a feltöltésre is lehetőség van, továbbá a térképek megjelenítésére több különféle mód is létezik.

## **10. Köszönetnyilvánítás**

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Gede Mátyásnak, aki a téma kiválasztásában és a dolgozat elkészítésében rengeteg segítséget nyújtott értékes tanácsaival, ötleteivel.

Továbbá köszönettel tartozom Mészáros Jánosnak a térképek vetületének meghatározásában, valamint a georeferálásukban nyújtott segítségéért.

## 11. Irodalomjegyzék

**DeBolt, Virginia, 2005:** HTML és CSS Webszerkesztés stílusosan. *Kiskapu Kft., Budapest, 2005.*

**Elek István, 2006:** Bevezetés a geoinformatikába. *ELTE Eötvös Kiadó, Budapest, 2006.*

**Elek István, Gercsák Gábor, 2012:** Developing Map Databases: Problems and Solutions. *In: Maps for the Future, Lecture Notes in Geoinformation and Cartography, Editors: Zentai László, Jesús Reyes Nunez, Springer Verlag, 2012. pp. 139-149.*

**Gál Tibor, 2006:** A webprogramozás. *Műegyetemi Kiadó, Budapest, 2006.*

**Grimmer, Lenz, 2003:** Navigating the Spatial Data Support in MySQL 4.1. *International PHP Conference, Frankfurt, Németország, 2003.*

**Irmédi-Molnár László, 1967:** Térképalkotás. *Tankönyvkiadó, Budapest, 1967. pp.100-102.*

**Mészáros János, 2009:** Lányi Sámuel Tisza-térképének georeferálása. *Diplomamunka, ELTE Térképtudományi és Geoinformai Tanszék, Budapest, 2009.*

**Moulding, Peter, 2002:** PHP Haladóknak, Fekete Könyv. *Perfact-Pro Kft., Budapest, 2002.*

**Reese, George; Yarger, Jay Randy; King, Tim, 2003:** A MySQL kezelése és használata. *Kossuth Kiadó, Budapest, 2003.*

**Schlossnagle, George, 2004:** PHP fejlesztés felsőfokon. *Kiskapu Kiadó, Budapest, 2004.*

### ***Internetes források, irodalmak:***

**Gede Mátyás, 2012** honlapja-oktató anyagok, *ELTE Térképtudományi és Geoinformatikai Tanszék*

<http://mercator.elte.hu/~saman/>

### **Google Earth Reference**

<https://developers.google.com/earth/>

### **Google Maps Reference**

<https://developers.google.com/maps/documentation/javascript/>

### **Google KML Reference**

<http://code.google.com/intl/hu-HU/apis/kml/documentation/>



**Györffy János, 2011:** Vetületanalízis. *Jegyzet. ELTE Térképtudományi és Geoinformatikai Tanszék, 2011.*

[http://mercator.elte.hu/~gyorffy/jegyzete/MScVettan2/vet\\_anal/JEGYZE50.htm](http://mercator.elte.hu/~gyorffy/jegyzete/MScVettan2/vet_anal/JEGYZE50.htm)

**MySQL 5.0 Reference Manual-** dokumentáció, és Spatial Extensions

<http://dev.mysql.com/doc/refman/5.0/en/>

<http://dev.mysql.com/doc/refman/5.0/en/spatial-extensions.html>

**OpenLayers**

<http://openlayers.org/>

**OpenStreetMap**

[http://wiki.openstreetmap.org/wiki/Main\\_Page](http://wiki.openstreetmap.org/wiki/Main_Page)

**PHP Manual-**Internetes kézikönyv, leírás

<http://php.net/manual/en/index.php>

**Dr. Varga József, 2005:** Vetületnélküli rendszerektől az UTM-ig. Budapest, 2005.

[http://www.agt.bme.hu/staff\\_h/varga/Osszes/Dok3uj.htm#vetuletnelkuli](http://www.agt.bme.hu/staff_h/varga/Osszes/Dok3uj.htm#vetuletnelkuli)

**Wikipédia**

MySQL: <http://hu.wikipedia.org/wiki/MySQL>

Apache: [http://hu.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://hu.wikipedia.org/wiki/Apache_HTTP_Server)

Google Maps: [http://hu.wikipedia.org/wiki/Google\\_Terkep](http://hu.wikipedia.org/wiki/Google_Terkep)

JavaScript: <http://hu.wikipedia.org/wiki/JavaScript>

PHP: <http://hu.wikipedia.org/wiki/PHP>

**W3C honlapja** (World Wide Web Consortium)

<http://www.w3c.org>

**W3Schools** honlapja-internetes oktatóanyag

<http://w3schools.com/>

*A honlapok utolsó látogatása: 2012.05.31.*

***A felhasznált térképek jegyzéke:***

Országos Széchényi Könyvtár Térképtára által szkennelt nagyfelbontású állományok:

**Gróf Vásquez Károly:** Buda és Pest szabad királyi városainak tájleírása. *Pest, 1837.*

Jelzete: tr\_2520. Négy szelvény.

**Görög Demeter, Kerekes Sámuel:** Magyar átlás azaz Magyar, Horvát és Tót országok vármegyéi. Bécs, 1802-11. Jelzete: ta\_4678. Tolna, Baranya, Somogy és Szepes vármegye térképei.

## 12. Melléklet

A térképtár jelenleg a <http://mercator.elte.hu/~ungvarizs/oszk> címen üzemel.

A térképtár feltöltő oldalán való belépéshez jelszó szükséges. A próba user-ekkel a hozzáférés ... néven és ... jelszóval:

test1: almafa → Első szint, új térkép létrehozása

test2: almafa → Második szint, más által létrehozott record módosítása

test3: almafa → Admin jogok, jóváhagyás és törlés, új felhasználó hozzáadása

### I.számú melléklet:

A relációs adatbázist alkotó táblák:

#### 1. oszk\_maps:

Mező neve	Adattípus fajtája	Magyarázat
ID	Int(10)	Azonosító
Title	Text	A térkép címe
Author	Tinyint(5)	Szerzője
Publisher	Tinyint(5)	Kiadója
Pubplace	Text	Kiadás helye
Pubdate	Varchar(30)	Kiadás ideje
Scale	Bigint(10)	Térkép méretaránya
Description	Text	Leírása
Shelf_number	Varchar(20)	Jelzete
Projection	Tinyint(5)	Vetülete
Keywords	Text	Kereséshez kapcsolódó kulcsszavak
Relmaps	Text	Hasonló térképek
Kmzfile	Varchar(99)	Google Earth-ös KMZ fájl neve
Imagefile	Varchar(99)	Képfájl neve
Googlemapsfiles	Varchar(99)	Google Maps fájl neve
Poly	Geometry	A térkép által lefedett terület sarokpontjai
Map_complete	Tinyint(1)	Jóváhagyták?

Uploaded_by	Text	Feltöltő felhasználóneve
-------------	------	--------------------------

## 2. oszk\_authors

Authorid	Tinyint(5)	Elsődleges kulcs
Au_author	Text	Szerző neve
Au_description	Text	Leírás a szerzőről
Au_complete	Tinyint(1)	Jóváhagyták?

## 3. oszk\_publishers

Publisherid	Tinyint(5)	Elsődleges kulcs
Publisherdescription	Text	Kiadó neve
Publisherplace	Text	Kiadó székhelye
Publisher_complete	Tinyint(1)	Jóváhagyták?

## 4. oszk\_projection

Projectionid	Tinyint(5)	Elsődleges kulcs
Projectionepsg	Int(11)	EPSG szám
Projectiondescription	Text	A vetület neve
Projection_complete	Tinyint(1)	Jóváhagyták?

## 5. oszk\_members

Id	Int(11)	Elsődleges kulcs
Username	Varchar(30)	Felhasználónév
Name	Varchar(40)	Teljes név
Pword	Varchar(50)	Jelszó
E_mail	Varchar(50)	E-mail cím
New	Tinyint(1)	Újat hozhat létre
Modify_1	Tinyint(1)	Saját rekordot módosíthat
Modify_2	Tinyint(1)	Más rekordot is módosíthat
Clear	Tinyint(1)	Törölhet
Accept	Tinyint(1)	Jóváhagyhat

## II. számú melléklet

A PHP fájlok listája és működésük:

Neve	Funckciója
Index.php	Kezdőoldal, keresés (hagyományos és földrajzi Google Maps-szel)
Osm.php	Kezdőoldal, keresés (hagyományos és földrajzi OpenStreetMap-pel)
Reszletes.php	A térkép adatlapja, megtekintése
Checklogin.php	A bejelentkező adatainak ellenőrzése
Clear.php	Térkép törlése az adatbázisból
Config.php	Változók
Create_newauthor.php	Új szerző bevitele az adatbázisba
Create_newprojection.php	Új vetület bevitele az adatbázisba
Create_newpublisher.php	Új kiadó bevitele az adatbázisba
Create_newuser.php	Új felhasználó bevitele az adatbázisba
Fajlok.php	Fájlok kiválasztása feltöltéshez, formátum ellenőrzéshez
Feltolt.php	Térkép feltöltése adatbázisba, utána lehetséges a fájlok kiválasztása feltöltéshez, formátum ellenőrzéshez
Jovahagy_success.php	Térkép jóváhagyása
Jovahagy_succes_kiado.php	Kiadó jóváhagyása
Jovahagy_success_szerzo.php	Szerző jóváhagyása
Jovahagy_success_vetulet.php	Vetület jóváhagyása
Jovahagyadatlap.php	A térkép adatlapjának ellenőrzése jóváhagyás során
Jovahagykiado.php	A kiadó adatainak ellenőrzése jóváhagyáshoz
Jovahagyrecord.php	A még ellenőrizetlen egységek listája
Jovahagyszerzo.php	A szerző adatainak ellenőrzése jóváhagyáshoz
Jovahagyvetulet.php	A vetület adatainak ellenőrzése jóváhagyáshoz

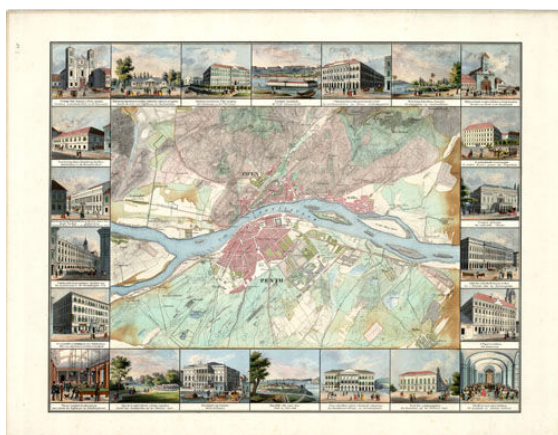
Login_succes_1.php	Sikeres bejelentkezés utáni kezdőoldal kezdőknek
Login_succes_2.php	Sikeres bejelentkezés utáni kezdőoldal, azoknak, akik módosíthatják mások rekordjait.
Login_succes_3.php	Sikeres bejelentkezés utáni kezdőoldal adminoknak
Logout.php	Kijelentkezést kezelő php oldal visszairányít az index.php-re siker esetén
Main_login.php	Bejelentkezési adatlap kitöltése
Mod.php	Módosítás adatbázisba írása
Modosit.php	Módosító adatlap, adatok bevitele
Modosit_1.php	Módosító adatlap, adatok bevitele, csak saját record módosítása
Modositrecord.php	Aki rendelkezik a módosító joggal, annak a módosítani kívánt egység kiválasztása
Modositrecord_1.php	Még nem jóváhagyott egység módosítása
Newauthor.php	Adatlap kitöltése új szerzőről
Newprojection.php	Adatlap kitöltése új vetületről
Newpublisher.php	Adatlap kitöltése új kiadóról
Newuser.php	Adatlap kitöltése új felhasználóról
Szintomb.php	A polygonok színeit tartalmazó fájl
Polyt_frissit.php	Feltölti a térkép sarkainak koordinátáit
Pontfelszed.php	A poligon sarokpontjainak bejelölés a drawing tools segítségével
Toroladatlap.php	A törölendő térkép adatlapja
Torolrecord.php	Kiválasztandó törölendő elem
Ujrecord.php	Új térkép hozzáadása
Upload.php	Képfájlok szerverre töltése
Upload_gmaps.php	Google Maps fájlok szerverre töltése zipként
Upload_kmz.php	Kmz fájl feltöltése szerverre
Zip.php	Kitömörít

Simple-slider.js	A csúszka működését leíró JS fájl. (Forrása: Google Code)
Simple-slider.css	A csúszka viselkedését írja le.
Style.css	A honlap stílusának leírófájla

### III. számú melléklet:

A felhasznált térképek.

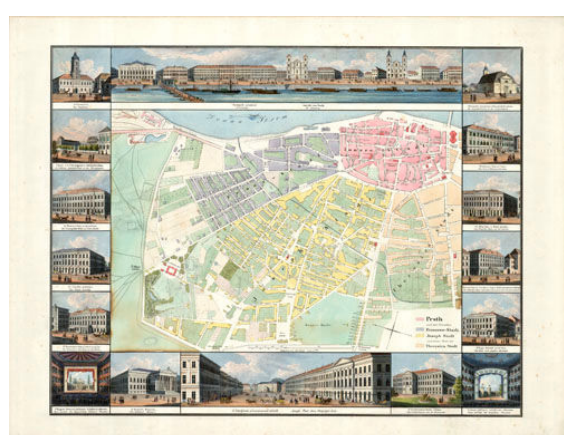
Tr\_2520 Áttekintő szelvény



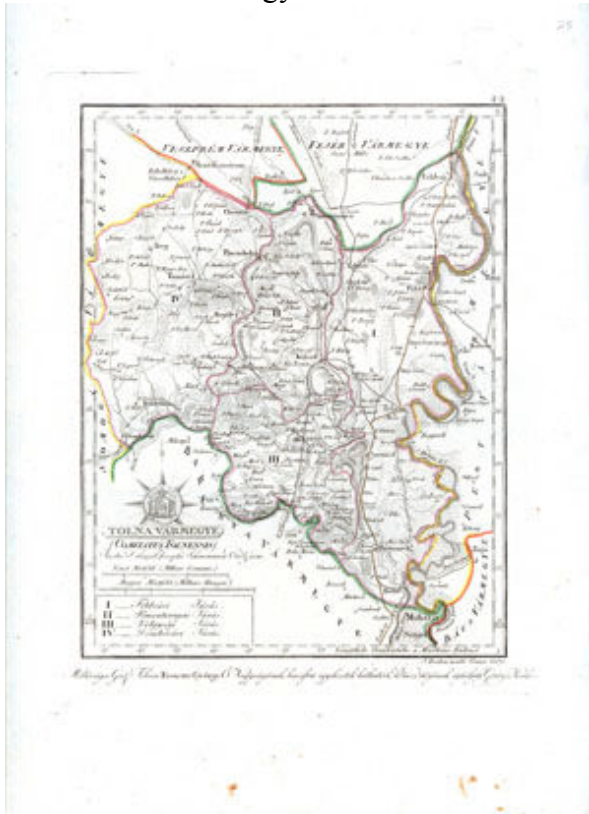
Budai Várat ábrázoló szelvény



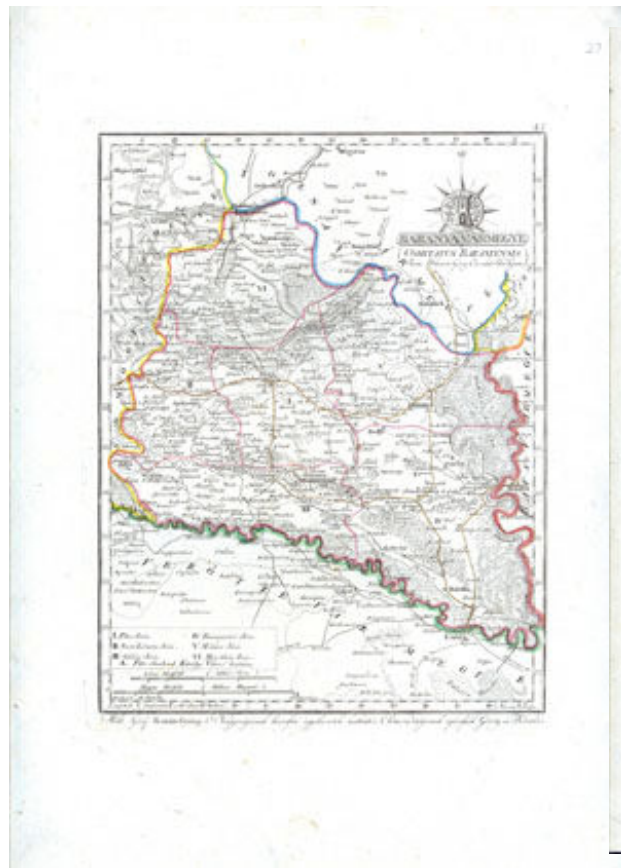
pesti belvárost ábrázoló szelvények



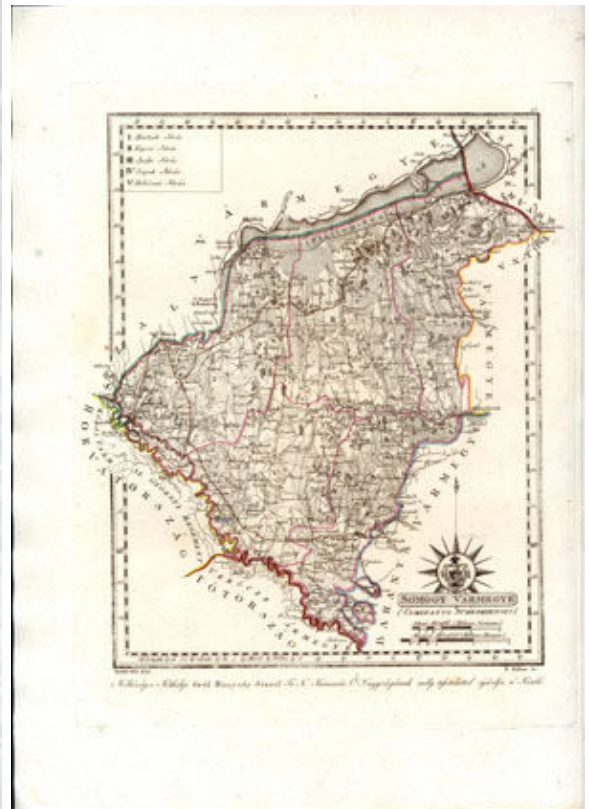
ta\_4678 Görög D. megyetérkép sorozata  
Tolna vármegye



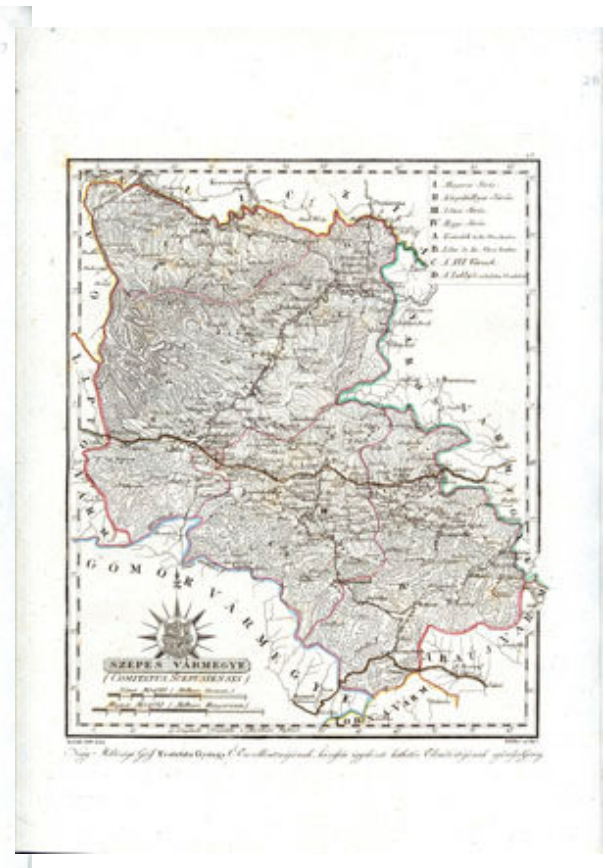
Baranya vármegye



Somogy vármegye

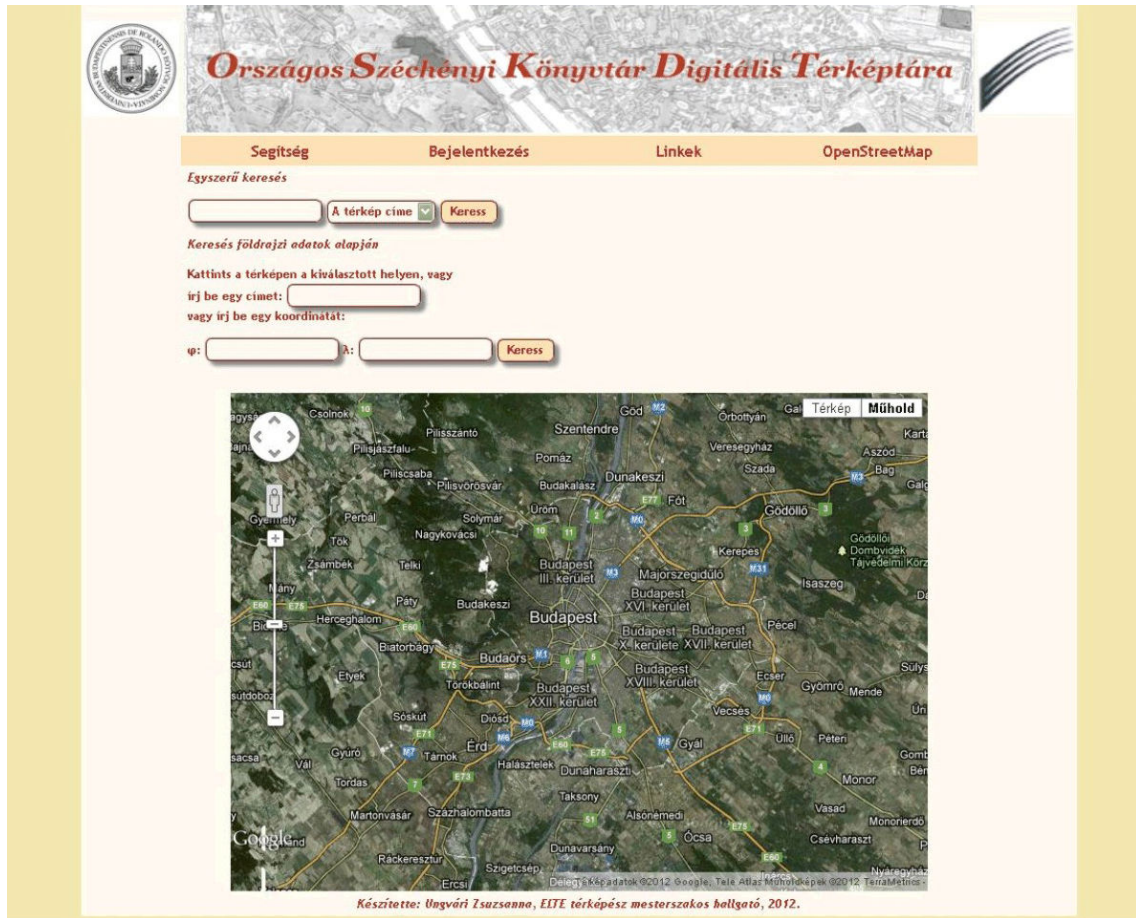


Szepes vármegye





**IV. számú melléklet**  
Képek a honlap működéséről.  
Kezdőlap



**Országos Széchényi Könyvtár Digitális Térképtára**

Segítség Bejelentkezés Linkek OpenStreetMap

Egyszerű keresés

Keresés földrajzi adatok alapján

Kattints a térképen a kiválasztott helyen, vagy  
írj be egy címet:   
vagy írj be egy koordinátát:

φ:  λ:

Készítette: Ungvári Zsuzsanna, EITÉ térképész mesterszakos hallgató, 2012.

A módosító, jóváhagyó és törölő oldalak keresője



**Országos Széchényi Könyvtár Digitális Térképtára**

Feltöltő oldal

Sikeres bejelentkezés. zsuzsi néven belépve.

Keresett szöveg: ta

Térkép címe	Szerzője	Kiadója	Kiadás helye	Kiadás ideje	Jelzete	Feltöltötte
Megyei Atlas	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_tolna	zsuzsi
Megyei Atlas	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1810	ta_4678_szepes	zsuzsi
Megyei Atlas	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_baranya	zsuzsi
Megyei Atlas	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_somogy	zsuzsi

## A jóváhagyás



### Országos Széchényi Könyvtár Digitális Térképtára



Feltöltő oldal - Jóváhagyás

Sikeres bejelentkezés. zsuzsi néven belépve.

[Kilépés](#)

**Térképek jóváhagyása**

Térkép címe	Szerzője	Kiadója	Kiadás helye	Kiadás ideje	Jelzete	Jóváhagyás?	Feltöltötte
Megyei Atlas Somogy vármegye	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_somogy	0	zsuzsi
Megyei Atlas-Baranya vármegye	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_baranya	0	zsuzsi
Megyei Atlas-Szepes vármegye	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1810	ta_4678_szepes	0	zsuzsi
Megyei Atlas-Tolna vármegye	Görög Demeter-Kerekes Sámuel	Ismeretlen	Bécs, Ausztria	1800	ta_4678_tolna	0	zsuzsi

**Szerző jóváhagyása**

Szerző neve	Jóváhagyás?
Görög Demeter-Kerekes Sámuel	0

**Szerző jóváhagyása**

Szerző neve	Jóváhagyás?
Görög Demeter-Kerekes Sámuel	0

**Kiadó jóváhagyása**

Nincs ellenőrizetlen tábla.

**Vetület jóváhagyása**

Nincs ellenőrizetlen tábla.

## Új felhasználó hozzáadása:



### Országos Széchényi Könyvtár Digitális Térképtára



Feltöltő oldal

Sikeres bejelentkezés. zsuzsi néven belépve.

[Kilépés](#)

**Új felhasználó hozzáadása**

Név

Felhasználónév

Jelző

Jelző újra

E-mail cím

Jogosultságok

Új-Bárki

Módosíthat  Nem

Jóváhagyhat  Nem

[Létrehoz](#)

# Nyilatkozat

ELTE Informatikai Kar, térképész mesterszak

ETR azonosító: UNZPABT.ELTE

Alulírott, Ungvári Zsuzsanna nyilatkozom, hogy jelen dolgozatom teljes egészében saját, önálló szellemi termékem. A dolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be. A diplomamunkámban felhasznált, szerzői joggal védett anyagokra vonatkozó engedély a mellékletben megtalálható.

A témavezető által elfogadott és elbírált diplomamunka elektronikus közzétételéhez (PDF formátumban a tanszéki honlapon)

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2012.05.31.

Ungvári Zsuzsanna