

Eötvös Loránd Tudományegyetem
Informatikai Kar
Térképtudományi és Geoinformatikai Tanszék

Webes térképkatalógus

Molnár Nikolett
térképész szakos hallgató

Témavezető:

Dr. Gede Mátyás
adjunktus



Budapest, 2012

Tartalomjegyzék

1. Bevezetés	3
2. A katalógus célja, elvárások	4
3. Tervezés	5
3.1. Általános működési terv	5
3.2. Adatbázisterv	6
3.3. Felhasználói felület.....	7
3.3.1. Googel Earth API.....	8
3.3.2. Keresés	9
3.3.3. Bejelentkezés	10
3.4. Kódolás, megvalósítás	10
3.4.1. Dinamikus tartalomgenerálás.....	10
3.4.2. Szerver-kliens összekapcsolás.....	11
3.4.3. Javascript.....	12
4. Adatbázis.....	13
5. Részletes bemutatás	15
5.1. Weboldalak kapcsolata	15
5.2. Keresés.....	16
5.2.1. Adatok alapján	16
5.2.2. Pontlerakással.....	16
5.3. A keresés eredménye	17
5.4. Bejelentkezés	19
5.5. Új térkép feltöltése.....	19
5.5.1. Úrlap.....	20
5.5.2. Georeferálás	21
5.5.3. Poligonrajzolás.....	23
5.6. Módosítás.....	24
5.7. Felhasználói élmény növelése	24
6. Webes szabványok.....	26
7. Tesztelés.....	28
8. Bővítési lehetőségek	29
9. Összegzés	30
10. Felhasznált irodalom.....	31
11. Köszönetnyilvánítás.....	32

1. Bevezetés

Mindennapjaink részévé vált a digitális világ, egyre fontosabbá vált az internet, ma már felhasználók milliárdjait köti össze, a világ elektronikus postájává vált.

A kor szellemének nélkülözhetetlen eszköze a világháló. Elnevezésére utal, hogy fontos szerepet tölt be úgy globális szinten, mint perszonális területen. Használata elengedhetetlen a teljesség igénye nélkül az üzleti-, kereskedelmi élet, valamint intellektuális programok területén. A kulturális élet, a gazdaság, pénzügyi forgalmak sem nélkülözhetik. Mára a személyes kapcsolattartásnélkülözhetetlen eszköze.

Az információáramlás hirtelen növekedésével egyre nagyobb lett az igény az online szolgáltatásokra.

Egyetemi éveim alatt több kurzus keretében is lehetőségem nyílt megismerkedni weboldalak elkészítésével, szerkesztésével. Az idő előrehaladtával érdeklődésem is kezdett egyre inkább a webes fejlesztés tárgyköre felé terelődni.

Választásom olyan témára esett, amelyben térképészeti ismereteimet az informatika egy aktuális, dinamikusan fejlődő területével, a web alapú alkalmazásfejlesztési technológiákkal összekapcsolhatom.

Diplomamunkám során ingyenes szoftverek segítségével valósítok meg egy internetes térképnyilvántartó rendszert. Ennek az oldalnak a megtervezésével, gyakorlati tevékenykedéssel alkalmam adódott mindkét témakörben szerzett tudásom tapasztalati tesztelésére, annak tovább fejlesztésére, tökéletesítésére.

Célul tűztem ki, hogy e konkrét feladaton keresztül bemutassam a rendszer lehetőségeit, szolgáltatásait, ezeken túl, elkészítsek egy élő, működő adatbázis háttérrel rendelkező honlapot.

2. A katalógus célja, elvárások

Az első magyarországi tájfutó verseny 1925. január 11-én zajlott le a Budai-hegységben. A második világháború után kezdett a tájékozódási futás szélesebb körben elterjedni. Azóta nagyon nagy mennyiségű térképanyag gyűlt össze. A térképek számával együtt a térképek színvonala is folyamatosan emelkedett. A térképek szaporodásával pedig nőtt a térképgyűjtők száma, és erősödött az igény egy térképkatalógus összeállítására.

Zentai László 1995-ben létrehozta az internetes Magyar Tájfutó Oldalakat. Itt évenkénti bontásban táblázatos rendszerben megtalálhatók az 1993 óta kiadott tájfutó térképek.

Az azóta eltelt időben többféle katalógus is megjelent, amelyek hosszabb-rövidebb időszakokat átfogva igyekeztek a térképeket bizonyos szempontok szerint rendszerezni. Mára már számos internetes katalógus is található. De mégis úgy gondolom, hogy szükséges egy olyan tájfutó térképkatalógus, amely a lehető legátfogóbb képet nyújtja az eddig megjelent térképekről. Az eddigi katalógusokhoz képest a legnagyobb előny az lenne, hogy a térképek egy másik térképen, pontos földrajzi helyzetükkel megjelenítve legyenek láthatók, megkönnyítve ezzel a böngészést.

A rendszerrel kapcsolatos további követelmény, hogy több ezer térképpel kapcsolatos adatok webes lekérdezése webböngészőn keresztül teljesüljön. Minden térképhez tartozik egy KML állomány is, amely a térkép helyzetét határozza meg. Szintén kezelni és szolgáltatni kell a térkép beszkenelt képét, illetve esetleges letölthetőségét más formátumban is. Fontos követelmény, hogy új adatok laikusok számára is könnyen megadhatók legyenek, de természetesen a webes szolgáltatásba feltöltés csak megfelelő jogosultsággal legyen lehetséges.

3. Tervezés

Az első lépésem a katalógus gondolatban való megtervezése volt. Fontos, hogy ez legyen az első lépés, hiszen egy jól megtervezett munkavégzés nagymértékben felgyorsíthatja a kivitelezést. Ebben az esetben a részfeladatok funkciójuk szerint kerülnek megalkotásra. Így könnyen tudjuk ellenőrizni a működésüket.

3.1. Általános működési terv

Készítettem egy általános működési tervet. Ez az elképzelés, a rendszer általános működésének vázlata arra szolgál, hogy mielőtt nekilátunk a munkának, kialakuljon bennünk egy kép a végleges formáról. Ez magában foglalja a szerkezeti felépítést és a megjelenést is.

Át kell gondolnunk, milyen követelményeket kell teljesítenie a rendszernek. Két legfontosabb szempont az emberi kényelem és a rendszer teljesítményének optimalizálása.

Olyan térképnyilvántartó-rendszert akartam létrehozni, amely digitális térképeket tárol, a rájuk vonatkozó adatokkal, pontos földrajzi helyzetükkel kiegészítve egy adatbázis háttérrel biztosítva. Fontos volt, hogy ezekre az adatokra lehessen szűrni, legyen elérhető a térképek közötti böngészés és lehetőleg ez minél interaktívabb módon valósuljon meg. Elengedhetetlen szempont a tervezéskor, hogy olyan rendszert hozzak létre, amely felhasználói jogosultsággal rendelkezőknek későbbiekben bővíthető legyen. Az oldal használható legyen regisztrált felhasználók számára, ugyanakkor bizonyos funkciók a nem regisztrált látogatók részére is elérhetőek legyenek. Ilyen funkció például a keresés. Ahhoz, hogy keresni lehessen a térképek között nem lesz szükséges belépni az oldalra, csupán a kereséshez kívánt adatokat kell megadni. Regisztrált felhasználók számára azonban lehetőség nyílik további funkciók elérésére. Ilyen például a térképek feltöltése.

A katalógus elkészítése két fő részre osztható:

- Az adatbázis létrehozása és feltöltése.
- Az internetes oldal, az adatbázis megjelenítésére.

Az oldal funkciói:

- keresés
 - térképekre vonatkozó adatok alapján
 - földrajzi pontok alapján
- térképek különböző megjelenítése
 - poligonok
 - ground overlay
- letölthető térképek
- bejelentkezés
- új térképek feltöltése
- térképek szerkesztése, georeferálása

3.2. Adatbázisterv

Mivel az adatbázis az alapja az oldalnak, az összes adatot ebben tároljuk, ezért célszerű azt alaposan, teljes optimalizálással megtervezni a leghatékonyabb működés érdekében, ugyanis a későbbi változtatás nehezen eszközölhető.

Adatbázis kezelő rendszerként a MySQL szervert választottam, több kurzus keretében is volt lehetőségem alaposabban megismerkedni vele. Ez az egyik legelterjedtebb adatbázis kezelő, egyszerű módszerekkel végezhetünk el dinamikus webhelyek kiszolgálását, stabil és megbízható. A MySQL egy többszálú, több felhasználós, SQL adatbázis szerver, nyílt forráskódú, így a fejlődés is jóval gyorsabb ütemben történik, mint a többi alkalmazás esetében. A fejlődés folyamatos, csaknem naponta jelenhetnek meg a frissebb, jobb verziók. A MySQL szervernek számos olyan része van, amit a felhasználókkal közösen fejlesztettek ki, ezzel is növelve a használhatóságot. A táblák, az adatok felvételét SQL nyelv használatával tudjuk megtenni. Az SQL (Structured Query Language – Strukturált lekérdező nyelv) egy, a relációs adatbázisok létrehozására és lekérdezésére szolgáló nyelv. Elsősorban atáblákban tárolt adatok visszanyerésére szolgál, de emellett támogatja az adatok beírását és csoportosítását is. Ez egy eljárásmentes nyelv, az adatokra vonatkozó tulajdonságokat kell megadni, nem az elérési módjukat.

A MySQL másik nagy előnye, hogy megvalósította a térbeli kiterjedésekre vonatkozó adatok kezelését is. Ez a Spatial Extensions kiegészítő az OGC (Open Geospatial Consortium) megoldásait követve kezeli a térbeli adatokat. A Spatial kiegészítőben 2 fő típus van a térbeli

adatok tárolására, egyik a geometry, másik a geography. A geometry tulajdonságú kifejezetten olyan adatok tárolására használatos, amelyet főként térképészek használnak a Föld térképezése szempontjából. A geometry típusú adat lehet egy meghatározott pont, vagy pontok többsége, vagy bármi a föld kerekén, aminek van pontos helyzete. Ez a későbbiekben nagyban megkönnyíti a munkát.

Az adatbázis kezelőfelületeként a phpMyAdmint választottam. A phpMyAdmin napjaink egyik legnépszerűbb ingyenesen használható webes alapú kezelőfelület a MySQL adatbáziskezelő-rendszerhez. A phpMyAdmin tulajdonképpen egy PHP függvénygyűjtemény. Célja a MySQL adatbázisok interneten keresztül történő, vizuális kezelése. A megfelelő jogosultságok mellett (lekérdezési, módosítási, törlési jogosultság), a grafikus felületnek köszönhetően sokkal könnyebben áttekinthető, mint a karakteres felületen üzemelő programok, egérmegnyitásokkal vezérelhető.

3.3. Felhasználói felület

A felhasználói felület kialakítása során több szempontot kellett figyelembe vennem. A használhatósági tervezés során elsősorban arra törekedtem, hogy a felhasználó minél kisebb erőfeszítéssel, minél egyszerűbben és könnyebben érje el szándékát, célzottan és eredményesen megtalálni a keresett tartalmakat. Az oldal kialakításában az információ elrendezése és a navigálás megkönnyítése azt kívánta, hogy 3 fő részre osszam az oldalt.

Oldalstruktúra:

- GoogleEarth API
- Keresés és vezérlők
- Találati lista

3.3.1. GoogleEarth API

A GoogleEarth egy ingyenes számítógépes program, egy virtuális földgömb. A Föld térbeli modelljére vannak vetítve térinformatikai adatok, műholdképek, légi felvételek. A GoogleEarth API az asztali program webes alkalmazásprogramozási felülete, weboldalakra beágyazva is tudjuk használni a program szolgáltatásait. A GoogleMaps-hez képest inkább a megjelenítésre helyezi a hangsúlyt, lehet háromdimenzióban nézelődni és rárepülni a helyekre.

Alapértelmezésben a térképek által ábrázolt területet fedő poligonok jelennek meg, így a néző szeme elé tárul az összes térképpel fedett terület. Ez azért jó ötlet, mivel a térképek között számos olyan található, amely ugyanarról a területről készült. Mivel a poligonok kapnak némi átlátszóságot, így könnyebben áttekinthetők azok a területek is, amelyekről több térkép is készült anélkül, hogy egymást kitakarnák. Ezek a poligonok KML fájlok.

AKML (Keyhole Markup Language) egy olyan jelölőnyelv, amely alkalmas a térben ábrázolt alakzatok. A KML fájl különféle tulajdonságokat határoz meg, ilyenek a hely, kép, poligon, 3D modell, textúra, leírás, stb. jelölése. Minden helynek van földrajzi hosszúsága és szélességi koordinátája. A további paraméterek a kamera-nézetet határozzák meg, ezek a dőlésszög, az irány, és a magasság.

Példa egy KML fájlra:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2">
  <GroundOverlay>
    <Icon>
      <href>http://mercator.elte.hu/~monoaat/php/images/1966-
      borzsony.jpg</href>
    </Icon>
    <gx:LatLonQuad>
      <coordinates>
        18.866486664673037,47.84503653315579
        18.931180379227122,47.844256075524974
        18.931224903086378,47.90542705861136
        18.866531188532292,47.906207516242176
      </coordinates>
    </gx:LatLonQuad>
  </GroundOverlay>
</kml>
```


Ennek az alkalmazásnak a beágyazása tette lehetővé a térképek térbeli elhelyezkedésének földrajzi megjelenítését, ezzelgrafikusan is megtekinthető teljes katalógus készletet. Az oldal címének beírás után, első betöltéskor a térképen megjelenik az adatbázisban tárolt összes térkép helyzetét jelölő poligon. Ezek száma és megjelenési módja majd a későbbiekben, a további szűrések alapján változik.

A Google Earth plugin lehetőséget biztosít raszteres képek domborzatra való ráfeszítésére is. Ezt kihasználva a térképek beszkenelt változatai is megtekinthetők domborzati viszonyokkal felruházva, a poligon által lefedett területről, mindössze egy plusz réteg bekapcsolásával. Ezek az úgynevezett ground overlay-ek is KML fájlok.

```
<kmlxmlns="http://www.opengis.net/kml/2.2"
xmlns:gx="http://www.google.com/kml/ext/2.2">
  <GroundOverlay>
    <Icon>
      <href>http://mercator.elte.hu/~monoaat/php/images/1966-
      borzsony.jpg</href>
    </Icon>
    <gx:LatLonQuad>
      <coordinates>
        18.866486664673037,47.84503653315579
        18.931180379227122,47.844256075524974
        18.931224903086378,47.90542705861136
        18.866531188532292,47.906207516242176
      </coordinates>
    </gx:LatLonQuad>
  </GroundOverlay>
</kml>
```

3.3.2. Keresés

Második nagyobb egység a keresés blokk. A célzott és hatékony böngészés érdekében kétféle keresési lehetőséget terveztem. Az egyik a térképre vonatkozó adatok szerinti szűrés, pl. a térkép kiadásának éve, méretaránya, vagy a rajzoló személye. Input mezőkbe beírva a keresési feltételt elindul a lekérdezés. Az üresen hagyott mezők a saját kategóriájukban minden egyes rekordot visszaadnak, ilyenkor nincs szűrés. Több mező kitöltése esetén a keresés csak olyan rekordokat ad vissza, amelyek minden feltételnek megfelelnek.

A lekérdezés elküldése után tehát frissül a Google Earth API tartalma is, így csak a keresett térképek poligonjait láthatjuk. Emellett a térkép további adataiegy táblázatban jelennek meg.

A másik keresés pontlerakással történik. Ha a Google Earth-ön megjelenő poligonok által ábrázolt térkép valamelyikéről több adatot kívánunk megtudni, akkor egy placemark lerakásával egy olyan lekérdezés fut le, amely a pontlerakás helyén lévő összes térképet visszaadja. Ez azért hasznos, mert ha egy területről több térkép is készült, a találati listában egymás alatt szerepel az összes kijelölt helyet magában foglaló térkép, a rá vonatkozó adatokkal, illetve kiegészítőkkal együtt.

A harmadik rész pedig a már fent említett táblázat, amelyben megjelennek a keresési találatok. Ez mindkét keresésre ad vissza eredményeket, szintén dinamikusan változik. Itt jelennek meg a találatokra vonatkozó egyéb az adatbázisban eltárolt adatok. Minden egyes térkép új sort foglal el, az oszlopok pedig a rájuk vonatkozó adatokat mutatják.

3.3.3. Bejelentkezés

Ezek mellett lehetőségük lesz a felhasználóknak regisztrálni az oldalra, amely felhasználói jogokat biztosít számukra. Bejelentkezés után lehetőségük nyílik további térképek feltöltésére. Az új térképek feltöltése a katalógusba 3 fő lépésből áll. Első a digitális kép feltöltése, és a rá vonatkozó adatok megadása. Második a térkép georeferálása. Harmadik pedig a térképet körbevevő keret megrajzolása.

3.4. Kódolás, megvalósítás

A kivitelezés megtervezések és az előzetes elképzelés alapján sokkal egyszerűbb. A fejlesztés előtt a megfelelő környezet kiválasztása volt az elsődleges feladat.

3.4.1. Dinamikus tartalomgenerálás

Ahhoz, hogy az oldalunk dinamikus legyen, a háttérben lévő adatbázisból adatokat kérhessünk le, tehát az oldalunk tartalma a lekérdezéstől függően változzon, szükséges egy olyan szerver oldali nyelv használata, amely biztosítja ezt. A PHP segítséget nyújt ebben.

A PHP egy elterjedt nyílt forráskódú szerveroldali programozási nyelv. A PHP egy betűszó, amely eredetileg a Personal Home Pages (személyes honlapok) rövidítése volt, de a jelentése időközben PHP: Hypertext Processor-ra (PHP: hiperszöveg-feldolgozó) változott. A PHP oldalak elkészítésénél az eredmények formázására a HTML-t használják. A PHP lehetővé teszi, hogy egy weboldal tartalmát a kiszolgálón módosítsuk, mielőtt az oldal a

böngészőprogramhoz kerülne. A kiszolgálón fut egy PHP-parancsfájl, amely igény szerint állítja elő vagy módosítja a HTML-kódot. A kiszolgáló először feldolgozza a PHP utasításokat, és csak a kész kimenetet küldi el a böngészőnek, így a programkód nem is látható kliens oldalról. A PHP nyelv segítségével olyan összetett alkalmazásokat is készíthetünk, mint pl. a bejelentkezés, az adatbáziskezelés, fájlkezelés, e-mail küldése, adatfeldolgozás, dinamikus listakészítés stb..

A PHP lehetővé teszi, hogy egy kiszolgálón futó parancsfájlban adjunk ki SQL-utasításokat, így a PHP-parancsfájlokat kell hozzákapcsolni a MySQL-adatbázishoz.

A PHP előnyei:

- Egyszerű, nagyon könnyen elsajátítható szintaxis
- Nagyon jó dokumentáció
- Ingyenes, nyílt forráskódú
- Gyorsaság
- Nagyobb teret biztosít a webfejlesztés során, azzal, hogy a kódja közvetlenül beágyazódik a HTML-be.

A PHP, mint HTML-be ágyazható nyelv rendelkezik olyan speciális tulajdonságokkal, amelyek elsőre ugyan nem tűnnek különlegesnek, mégis nagymértékben megkönnyítik a webalapú alkalmazások fejlesztését. A PHP kódban a kliensoldal felől érkező HTTP, POST vagy GET változók globális tömbként elérhetőek anélkül, hogy bármilyen külön műveletet, függvényhívást kellene ehhez végrehajtanunk.

3.4.2. Szerver-kliens összekapcsolás

A webböngészők olyan kliensek, amelyek weboldalak megtekintésére és azokkal való interaktív kapcsolattartásra alkalmasak, a webes kommunikáció ügyféloldalaként működnek.

A kliens-szerver modellek lehetővé teszik, hogy kapcsolatba kerüljön a kliens a webkiszolgálóval és az weboldalakat továbbítsa az ügyfélgépek felé. A kliens kéréseket, lekérdezéseket küld a szervernek, amiket az feldolgoz, majd visszaküldi a választ, hogy a kliens megjelenítse a kért weblapot.

Az Ajax (Asynchronous JavaScript and XML) interaktív webalkalmazások létrehozására szolgáló webfejlesztési technika. Képes az oldal újratöltése nélkül, a háttérben adatokat cserélni a kliens és a szerver között. Az Ajax küld egy kérést a szerverre, amelyen lefut a

szerveroldali szkript, majd az visszaküldi az eredményt. Az Ajax-ot használó oldalak csak az oldal szükséges részét frissítik, így gyorsabban reagálnak a felhasználói interakciókra. Segítségével a jövőben a webes alkalmazások minden eddiginél interaktívabbá és így széles körben is sokkal népszerűbbé válhatnak.

Az Ajax a következő technikák kombinációja:

- XHTML (vagy HTML) és CSS a tartalom leírására és formázására.
- DOM kliens oldali script nyelvekkel kezelve a dinamikus megjelenítés és a már megjelenített információ együttműködésének kialakítására.
- XMLHttpRequest objektum az adatok aszinkron kezelésére a kliens és a webszerver között. XML formátumot használnak legtöbbször az adattovábbításra a kliens és a szerver között, bár más formátumok is megfelelnek a célnak, mint a formázott HTML vagy a sima szöveg.

3.4.3. Javascript

A JavaScript, mint neve is mutatja, egy szkript nyelv, a HTML lapokba lehet beágyazni, a lappal együtt töltődnek le, majd a böngésző értelmezi és futtatja azokat. A JavaScript-et azért fejlesztették ki, hogy az egyébként statikus lapokba dinamizmust vigyenek, látványosabbá tehetjük vele azokat, dinamikusan, generálhatunk HTML lapokat. Számos praktikus dologra használhatjuk még fel, pl. jelszót kérhetünk a lapunk megtekintéséhez, vagy ellenőrizhetjük az elküldendő adatok helyességét még a kliens. A JavaScript-re az eseményvezéreltség jellemző. Ez azt jelenti, hogy néhány programok a felhasználó bizonyos tevékenységeire válaszolva indulnak el. Esemény lehet pl. az, ha felhasználó megnyom egy gombot, vagy ráviszi az egér mutatóját egy linkre. Ezekkel a kliens oldali eseménykezelők egy részével igyekeztem az oldalt felhasználó közelebbé tenni.

4. Adatbázis

A rendelkezésemre álló alapanyag meghatározó volt az adatbázis szerkezetének kialakításában. A beszkenelt térképekre vonatkozó adatok egy táblázatban szerepeltek, nagyjából ezek szerint alakítottam az adatbázis mezőit is. Így a *terkep*ek nevű táblában a következő oszlopokat hoztam létre:

- Integer típusú:
 - *id*: minden táblában célszerű egy olyan mezőt létrehozni, amely tulajdonságainál be van állítva az *auto_increment* és a *primarykey*, így minden egyes sor beszúrásánál ez az *id*mező értéke 1-el növekszik, így a mezőnek a sorok értékei mindig nagyobbak, mint az előző beszúrásnál kapott érték. Ezáltal az összes térképnek lesz egy egyedi azonosítója.
 - *a térkép méretaránya*,
 - *kiadásának éve*.
- Szöveges mezők:
 - *a térkép címe*,
 - *a környező települések listája*,
 - *a megye*,
 - *a megjegyzések*,
 - *a térkép rajzolója*,
 - *az image*: Minden térképet egy mappában tároltam el, ezáltal azonos elérési útvonal lett, így ez az oszlop a digitális képek nevét tartalmazza.
 - *a GCP*: ebben az oszlopban tárolom a georeferálás során lerakott illesztőpontok koordinátáit, pontosvesszővel elválasztva.
 - *KML*: ebben a mezőben vannak a térképekhez tartozó azon KML fájlok, amelyek a digitális kép megjelenítésére alkalmasak.
 - *poligon_kml*: a térképet körülvevő poligonok koordinátái. Későbbiekben ezek felhasználásával lehet kiszámítani a Google Earth API-ban való megjelenítéshez szükséges adatokat.
- Geometry mező pedig a *poligon*: A poligonok tárolásához a geometry típuson belül a polygont választottam, hiszen a további keresésekhez, elemzésekhez ezt használva egyszerűsödnek leginkább a lekérdezések. Egyik nagy előnye ennek a típusnak, hogy

a poligon sarokpontjait ismerve bármely pontról meg tudja mondani, hogy a poligon tartalmazza-e vagy sem a megadott koordinátát.

A felhasználó táblában tárolom a regisztrált felhasználók adatait. Ide mindössze 3 oszlopot készítettem:

- *id*,
- *nev*,
- *pwd*: a felhasználók jelszavát titkosított formában tárolja az adatbázis. A *password()* függvény egy titkosítási algoritmust alkalmaz egy szöveges karakterláncra. Ennek célja a titkosított jelszavak biztonságának megőrzése. A bejelentkezés ellenőrzésekor nem kell ismerni az eredeti jelszót, hiszen ha a helyes szöveget írja be a felhasználó, akkor a függvény ugyanazt a kódot állítja elő, mint legelőször, elég tehát a titkosított jelszót összehasonlítani az adatbázisban szereplő értékkel.

Információk tárolásához vagy kereséséhez csatlakozni kell az adatbázishoz, egy érvényes lekérdezést kell küldeni, az eredményt ki kell tudni olvasni, majd le kell zárni a kapcsolatot.

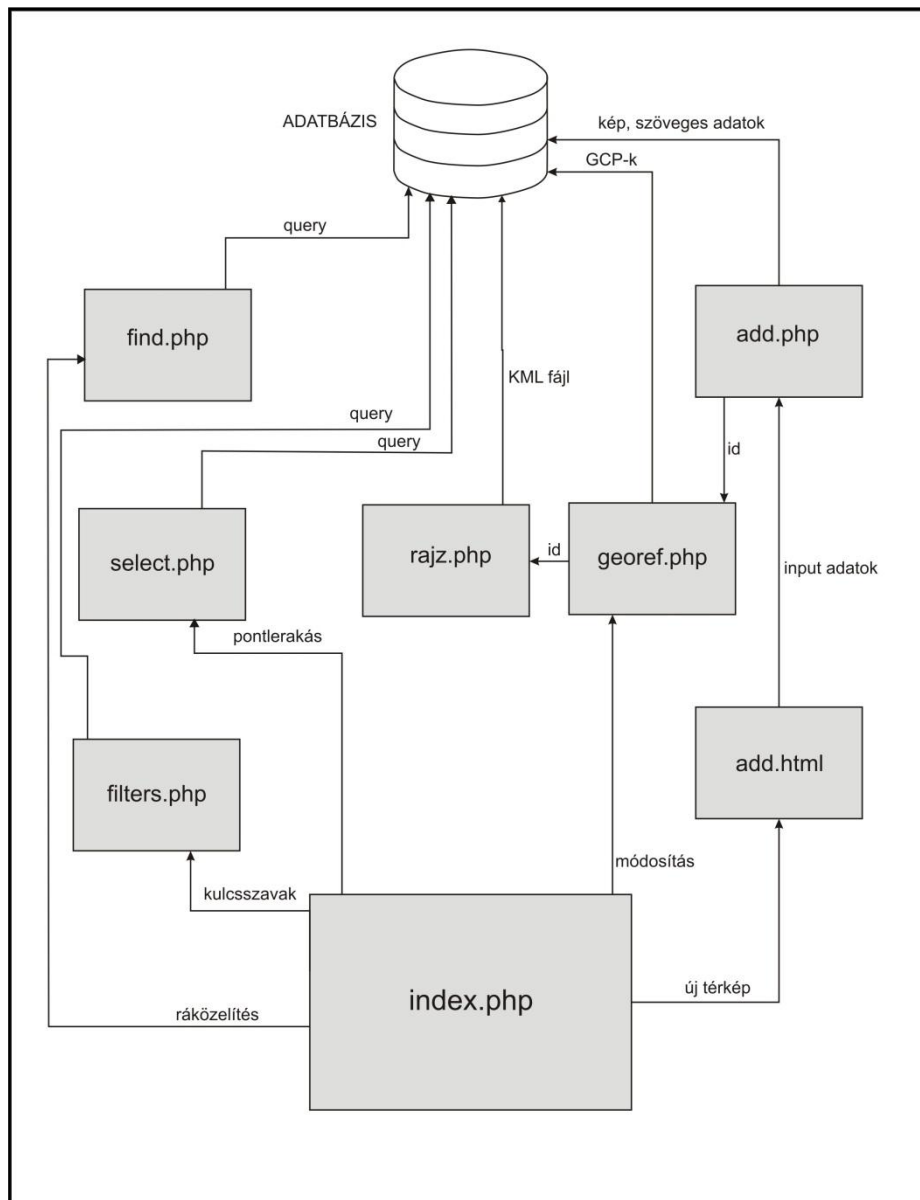
5. Részletes bemutatás

Az elkészített webes alkalmazás legfontosabb feladata a nyilvántartó-rendszer térképei közötti gyors böngészése, hatékonykeresése, azok listájának bővíthetősége.

A program gerincét az *index.php* nevezetű fájl adja, amiben a program működésétvezérlem. Azoldal megjelenését és stílusát HTML és CSS használatával készítettem el.

5.1. Weboldalak kapcsolata

A könnyebb érthetőség érdekében készítettem egy folyamatábrát, amely bemutatja, hogy az alkalmazás fájljai milyen kapcsolatban állnak egymással.



5.2. Keresés

Mivel a nyilvántartórendszerem egy katalógus, alappillére az adatbázisban tárolt adatok célzott és sikeres tartalmainak kiszolgálása.

5.2.1. Adatok alapján

Az alapvető elképzelés az volt, hogy szűréskor teljes szinkronban legyen a Google Earth API-n megjelenő poligoncsoport a táblázatban megjelenő eredményhalmazzal. Így a keresés után a poligonokat és a táblázatot is frissíteni kell. Ezért legegyszerűbb volt úgy megírni az SQL lekérdezést, hogy minden a két rész megjelenítéséhez szükséges adatot egyben kapjunk meg.

```
$query="SELECT id, cim, ev, mereta, telep, megye, rajz, megj FROM terkepek";
```

Továbbá, úgy terveztem a keresést, hogy az input mezők szűrőként viselkedjenek. Ha kitöltetlenül maradnak, akkor az azt jelenti, hogy az üres mező kategóriájára nincs szűrés, minden rekordot kilistáz. Ebből következik, hogy ha az összes mezőt üresen hagyjuk, az adatbázisban szereplő teljes térképlistát megkapjuk.

Amennyiben valamelyik mező ki van töltve, a lekérdezésnek ki kell egészülnie a beírt feltétellel. Például ha a cím mezőbe került szöveg, akkor a lekérdezéshez hozzá kell fűznünk a feltételt.

```
if(!empty($cim))  
$query .= "cim LIKE '%" . $cim . "%' AND "
```

Több mező kitöltése esetén a fentihez hasonló módon a php hozzáfűzi a lekérdezéshez a megfelelő feltételt.

5.2.2. Pontlerakással

A pontlerakással történő keresést külön checkboxban kell aktiválni. Ezután a térképre kattintva eltároljuk egy *mf* és *m* nevű változóban a pont földrajzi szélességét és hosszúságát.

Tájéfutó térképek katalógusa

Név: Jelszó:

Keresés

A térkép címe

A térkép méretaránya

Környező település

A térkép kiadásának éve

Keresés pontlerakással

Vezérlők

Átlátszóság:

Túlmagasítás mértéke:

1

1,5

2

Cím	Évszám	Méretarány	Környező települések	Megye	Rajzolta	Megjegyzés		
Döbrönte kupa 1968.	1968	25000	Bakonypölske, Döbrönte, G	Veszprém		fekete-fehér, helyi kiadás		
Döbrönte Kupa 1969.	1969	25000	Döbrönte	Veszprém		helyi kiadás		
Döbrönte Kupa 1971	1971	20000	Döbrönte, Magyarpolány	Veszprém		helyi kiadás		

Az Ajax hívásakor megadom, hogy milyen metódust használok, átadom a *find.php* fájlnek az url-ben a két változó értékét (a pont földrajzi helyzetét).

```
xmlhttp.open("GET", "find.php?mfi="+mfi+"&mla="+mla, true);
```

A lekérdezésben meg kell adnom minden szükséges további mezőt, hiszen az első eredményeket bemutató tábla ez alapján fog frissülni.

```
$query="SELECT id, cim, ev, mereta, telep, megye, rajz, megj FROM
terkepekwhereContains(poligon,GeomFromText('POINT('.$mfi.'
'.'.$mla.')'))=1;";
```

5.3. A keresés eredménye

Az adatbázisban kikeresett térképek adatait a PHP egy táblázatba rendszerezi. Az alábbi példához hasonlóan kapjuk meg a térképekre vonatkozó többi adatot.

```

<table>
<tr>
<td>
    A térkép címe
</td>
<td>
<selectid="cim">
<?php
require_once('dbvars.php');
$dbc=mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
$query="SELECT id, cim FROM terkepek";
$r=mysqli_query($dbc, $query);
$num_rows = mysqli_num_rows($r);
for($i = 0; $i<$num_rows; $i++)
{
$tmp = mysqli_fetch_row($r);
print("<optionvalue=".$tmp[0].">".$tmp[1]."</option>");
}
mysqli_close($dbc);
?>
</select>
</td>

```

Az adatbázisban tárolt adatokon kívül további oszlopokat adtam hozzá minden egyes térkép tulajdonságai mellé. Kiegészítettem egy jelölőnégyzettel, amit bepipálva a térkép Google Earth-re feszített változata is elénk tárul, domborzati viszonyokkal felruházva. Újabb oszlopban pedig a térképek letölthetőségét biztosítom.

Név: Jelszó:

Tájéfutó térképek katalógusa

Keresés


A térkép címe

A térkép méretaránya

Környező település

A térkép kiadásának éve 1966

Keresés pontterakással



Vezérlők

Áttátszóság:

Tülmagasítás mértéke:

1

1,5

2

Cím	Évszám	Méretarány	Környező települések	Megye	Rajzolta	Megjegyzés	
Börzsöny	1966	25000	Nagyirtáspuszta	Pest	Magyarszékly Béla	Kartográfiai Vállalat	
Balaton	1966	250000	Balatonszőlős	Veszprém	Kis Károly	foltos a térkép néhol	
Dobogókő	1966	25000	Dobogókő	Pest	x	x	
Bocskai kert	1966	25000	Bocskai kert	Hajdú-Bihar	x	x	
Egvetemi és főiskolai tájékozódási futó bajnokság	1966	25000	Ecseny, Lulla, Nyim, Ságvár	Somogy			
Balogváltás	1966	25000	Mátraszentimre	Heves, Nógrád			
Észak-Pilis	1966	25000	Dömös, Lepence, Visegrád	Komárom, Pest			
Földes Ferenc tájékozódási váltóverseny	1966	25000	Balatonendréd, Zamárdi	Somogy		sötétzöld színekkel	
III. Nemzetközi Vasutas Kupa és Csik Judit év.	1966	25000	Tarján, Tornyó	Fejér, Komárom			

5.4. Bejelentkezés

Regisztráció után minden felhasználóról három féle adatot tárol az adatbázis: a felhasználónevet, a jelszót, illetve egy egyedi azonosítót. A bejelentkezés mezőibe belépéshez be kell írni a felhasználónevet és a jelszót. A Belépés gomb megnyomásakor az alkalmazás összeveti a beírt adatokat az adatbázisban lévőkkel. Ha a felhasználónév és a jelszó érvényes, a felhasználó sikeresen beléptethető.

```
require_once('dbvars.php');

session_start();
//BEJELENTKEZÉS
if (!isset ($_SESSION['ID'])) {

if (isset($_POST['login'])) {

        //Csatlakozás az adatbázishoz
        $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
        $nev=mysqli_real_escape_string($dbc, trim($_POST['nev']));
        $pwd=mysqli_real_escape_string($dbc, trim($_POST['pwd']));

if (!empty($nev) && !empty($pwd)) {
        $query = "SELECT * FROM users WHERE nev = '$nev' AND pwd =
        password('$pwd')";
        $data=mysqli_query($dbc, $query);
```

```

if (mysqli_num_rows($data) == 1) {
$row = mysqli_fetch_array($data);
    $_SESSION['ID'] = $row['ID'];
    $_SESSION['nev'] = $row['nev'];
}
else{
$error_msg = "Baj van!";
echo$error_msg;
}
}
}
}
}

```

5.5. Új térkép feltöltése

Az index.php oldalon bejelentkezés után megjelenik egy link, amelyre kattintva a regisztrált felhasználók új térképeket adhatnak hozzá a katalógushoz.

```

<?php
if (isset($_SESSION['nev']))
echo "<input type=\"button\" value=\"Új térkép feltöltése\"
onClick=\"location.href='add.html';\">";
?>

```

5.5.1. Űrlap

Az új térkép feltöltésére mutató linkre kattintva egy űrlapot kell kitölteni a térképre vonatkozó adatokkal, valamint a besz kennelt állományt fel kell tölteni. Az űrlapon vannak olyan mezők, amiket kötelező kitölteni, anélkül a térkép nem kerülhet be az adatbázisba. Ilyen pl. a térkép címe vagy a feltöltött fájl.

```

if (isset($_POST['submit'])) {
//A kép adatainak kinyerése a POST tömbből
$name = $_POST['name'];
$ev = $_POST['ev'];
$mereta = $_POST['mereta'];
$telep = $_POST['telep'];
$megye = $_POST['megye'];
$megj = $_POST['megj'];
$rajz = $_POST['rajz'];
$image = $_FILES['image']['name'];
$image_type = $_FILES['image']['type'];
$image_size = $_FILES['image']['size'];
if (!empty($name) && !empty($image)) {
if (((($image_type == 'image/gif') || ($image_type == 'image/jpeg')
|| ($image_type == 'image/pjpeg') || ($image_type == 'image/png'))
&& ($image_size > 0) && ($image_size <= IMG_MAXFILESIZE)
) {
if ($_FILES['image']['error'] == 0) {
$target = IMG_UPLOADPATH . $image;
if (move_uploaded_file($_FILES['image']['tmp_name'],
$target)) {
$inserted_id = insert_terkep_adat($name, $ev, $mereta,

```

```

$telep, $megye, $megj, $rajz, $image);
}else {
die('Nem sikerült mozgatni a fájlt!');
    }
}
}else {
echo '<p class="error">GIF, JPEG, vagy PNG, max' .
(IMG_MAXFILESIZE / 1024) . ' KB.</p>';
    }
    @unlink($_FILES['image']['tmp_name']);
    // visszamegyünk a formra
header('Location:georef.php?id=' . $inserted_id);
die();
}else {
echo '<p class="error">Kérlek töltsd ki az összes mezőt!</p>';
    }
}
}

```

1. Térkép feltöltése

A térkép címe:

Kiadásának éve:

Méretaránya:

Környező települések:

Megye:

Rajzolta:

Megjegyzés:

Feltöltendő térkép: Nincs fájl kiválasztva

Az oldalon a mentés gomb megnyomásakor egy újabb oldalra kerül a felhasználó a térkép georeferálásához. Átírányításkor az adatbázisba újonnan bekerült térkép id-jét át kell adni a következő oldalnak, hogy az be tudja tölteni a már feltöltött térképet.

5.5.2. Georeferálás

A térképek georeferálása során a beszkenelt raszteres képformátumú állomány képpontjait georeferenciával látjuk el. Kezdeti állapotban a raszteres kép képpontjainak csak a kép síkkoordináta-rendszerében érvényes pixel-koordinátái vannak. A georeferálás folyamán illesztőpontokat (GCP – Ground Control Point) választunk, amelyeknek megadjuk mind a pixelkoordinátáit, mind pedig a térképi koordinátáit.

A kép koordináta-rendszerének megjelenítéséhez az OpenLayers nevű alkalmazást választottam. Ez tulajdonképpen egy Javascript függvénykönyvtár, melynek segítségével interaktív térképként jeleníthető mega beszkenelt állomány. Ahhoz, hogy a kép pontjait térképi koordinátákkal felruházhassuk, a GoogleMaps segítségét választottam. A bal oldali OpenLayers-en látható feltöltött képen és a jobb oldali GoogleMaps-en is meg kell találnunk ugyanazt a pontot, majd egymásnak meg kell feleltetni azokat. Az új GPC gomb megnyomásakor mindkét térképre kapunk egy-egy illesztőpontot. Ezzel az illesztőponttal megadjuk mindkét térképen a pont valós helyzetét. Ezt a műveletet legalább háromszor el kell végezni, hogy létrejöhessen a georeferálás.

A georeferálás során kétféle illesztőponttal kell dolgozni. Az egyik a képhez tartozó x, y pont, a másik a földi koordinátapár: φ, λ . Ezeket a párokat kell egymásnak megfeleltetni.

$$(x_1, y_1) \longrightarrow (\varphi_1, \lambda_1)$$

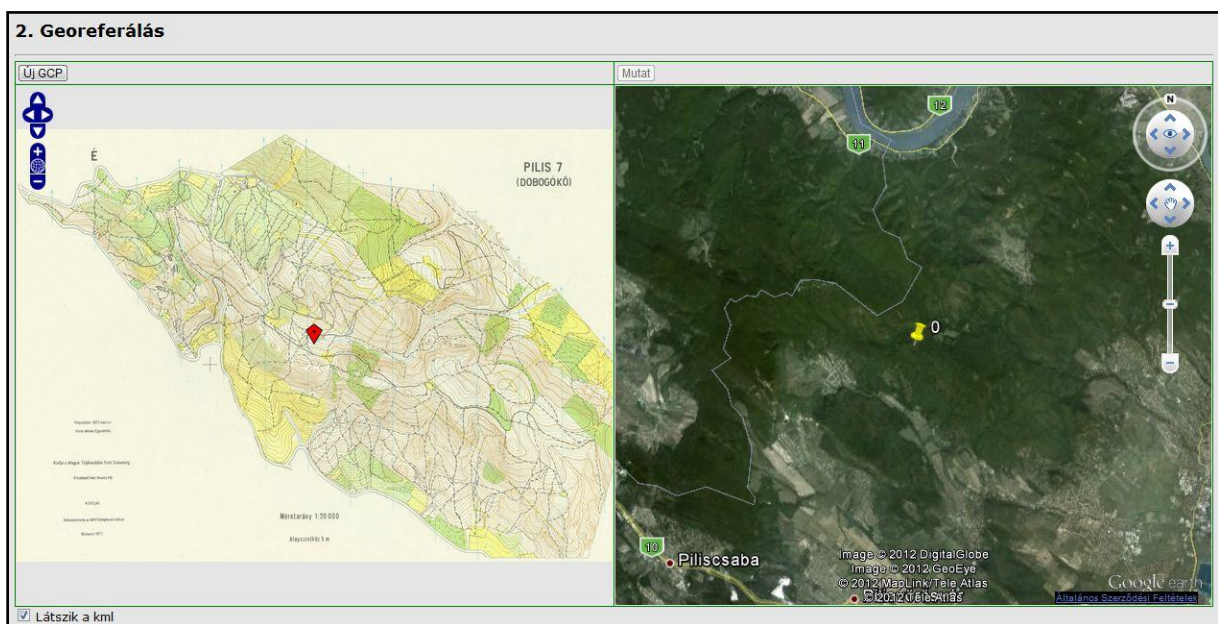
$$(x_n, y_n) \longrightarrow (\varphi_n, \lambda_n)$$

Gauss-elimináció segítségével megoldhatjuk azt a lineáris egyenletrendszer, ahol:

$$\varphi = a \cdot x + b \cdot y + c$$

$$\lambda = d \cdot x + e \cdot y + f$$

Keressük azt az a, b, c, d, f -et, hogy a két pont közötti eltérés legyen minimális.



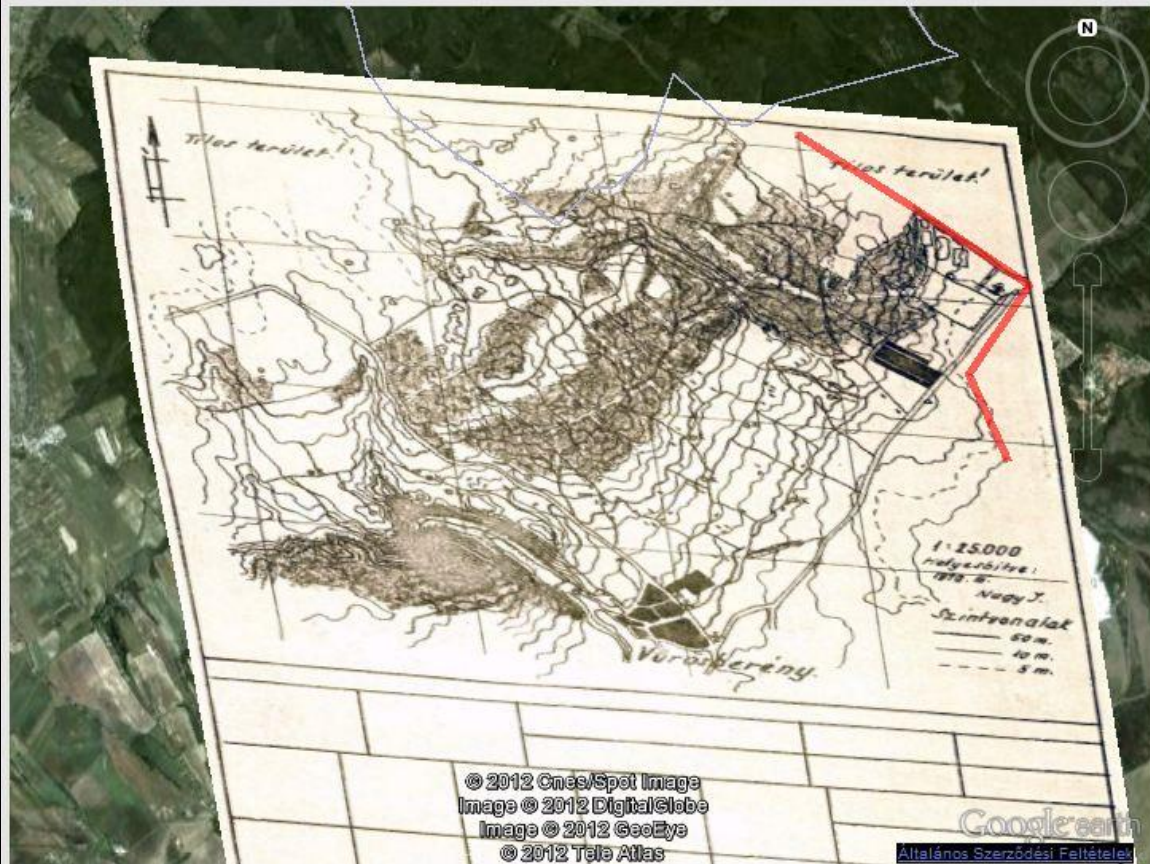
Ha megvan a 3-3 illesztőpontunk, akkor a Mutat gombra kattintva ellenőrizhetjük hogyan igazodik a GoogleMaps térképre a feltöltött térképünk. Ha módosítani szeretnénk, egy checkbox kipiálásával eltüntethetjük a már georeferált térképet és újabb illesztőpontokat helyezhetünk el a pontosabb elhelyezés érdekében. Addig ellenőrizhetünk, amíg nem véljük teljesen pontosnak a térkép helyzetét, majd a Küldés gombra kattintva tárolhatjuk azokat az adatbázisban. Ez után az alkalmazás a térkép id-jének elküldésével átirányít az utolsó feladatra, a poligonrajzolásra.

5.5.3. Poligonrajzolás

Miután georeferáltuk a feltöltött térképet utolsó lépésben meg kell rajzolnunk a kezdőoldalon megjelenő hozzátartozó poligont. A *rajz.php* fájl megkapja a térkép id-jét és a *rajz.js* fájl segítségével biztosítja a rajzolás lehetőségét.

A már georeferáláskor elmentett lookdata tömbből kinyerjük a térkép helyzetét, majd ráillesztjük a GoogleMaps-re, ezzel is könnyítve a feltöltő munkáját.

3. Poligon rajzolás



A rajzolás után a Küldés gomb megnyomásával elküldjük az adatokat, bekerülnek az adatbázisba.

```
$query="UPDATE terkep SET poligon =  
GeomFromText('POLYGON(('$poligon.))'), poligon_kml =' '$poligon2. ''  
whereid=" .intval($id);
```

Ha ez megtörtént, visszakerülünk a kezdőoldalra, már a legutóbb feltöltött térképpel kiegészült listával.

5.6. Módosítás

Bejelentkezés után a kilistázott térképek helyzetére vonatkozó módosításokat végezhet el a felhasználó. Az eredménytábla utolsó oszlopában egy link mutat erre a lehetőségre. A kijelölt térkép módosításához az egyetlen szükséges adat a térkép id-je, amit az *index.php* átad a *modif.php*-nak. A *modif.php* betölti az új térkép georeferálásához hasonló felületet, annyi

különbséggel, hogy az adatbázisból behívja a módosítani kívánt térkép illesztőpontjait is. Ezzel láthatóvá válik az esetleges eltérés a térkép valós helyzete és az adatbázisban rosszul megadott, pontatlanabb helyzet. Az illesztőpontok helyzetét szabadon módosíthatjuk, újabb pontokat adhatunk hozzá, majd eltávolíthatjuk azokat. Ha módosítottuk a georeferálást, akkor szükséges a térkép poligonjának újrarajzolása is, ezért a pontok mentése után a felhasználó átirányul a poligonrajzolásra. Ennek menete az új térkép körberajzolásával megegyezik.

5.7. Felhasználói élmény növelése

Az oldal látogatói számárakényelmi és használhatósági szempontból könnyíteni akartam az információ megértését. Ehhez olyan interakciódizájnt alkalmaztam, amely elsősorban a műveletekre jellemző. Dinamizmussal igyekeztem átláthatóbbá tenni az információt, növelni annak tartalmát.

Térképek címe

A térkép címére kattintva a GoogleEarth API a térkép poligonját megkeresve ráközelít arra, hogy könnyebben azonosítható legyen. Az adatbázis *terkepek* táblájából, a *poligon* nevű mezőből kinyerjük a poligon adatait, majd beillesztjük azokat a megfelelő helyre.

```
var s = xmlhttp.responseText;
varlookarray = s.split(" ");
varlookAt = e.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
lookAt.setLatitude(parseFloat(lookarray[0]));
lookAt.setLongitude(parseFloat(lookarray[1]));
lookAt.setRange(12000);
ge.getView().setAbstractView(lookAt);
```

Teljes nézet

A teljes nézet behívása során az elő betöltéskor megjelenő alpnézet visszahozása volt a cél. Ekkor a GoogleEarth-ön megjelenik a katalógus összes poligonja, és a könnyebb átláthatóság érdekében térképek nincsenek kilistázva az eredménytáblában. A gomb megnyomásakor Ajax

segítségével a poligonok KML fájljai megjelennek a térképen, ezzel a pontlerakásos szűrést felfüggeszti.

```
functionviewReset() {
    // Getthecurrentview
    varlookAt =
    ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);
    // Setnewlatitude and longitudevalues
    lookAt.setLatitude(47.180086);
    lookAt.setLongitude(19.503736);
    lookAt.setRange(700000.0); //default is 0.0
    // Update theviewinGoogleEarth
    ge.getView().setAbstractView(lookAt);
}
functionremovePoligon() {
    varfeatures = ge.getFeatures();
    while (features.getFirstChild())
    features.removeChild(features.getFirstChild());
}
```

Átlátszóság

A GoogleEarth-re feszített térképek megjelenítésénél a plugin biztosít egy olyan lehetőséget, hogy a KML fájl átlátszóságát szabályozhassuk. Ehhez elhelyeztem egy csúszkát a Vezérlők blokkban, ahol százalékosan állíthatjuk a képek átlátszóságát.

Túlmagasítás

A domborzatmodell által nyújtott hatás fokozása érdekében a térbeli viszonyok torzítását is látványosnak találtam. Ezzel ugyan plusz információt nem kap a felhasználó, de esztétikai szempontból plusz élménnyel gazdagodhatunk.

6. Webes szabványok

A világháló alapvető építőeleme a három fő webes szabvány, a HTML, a CSS és a JavaScript. Mára már beigazolódott ez a vezérelv, amelyet professzionális webfejlesztők és webdesignerek is vallanak, hogy a tartalom, a stílus és a működés szétválasztása a legjobb módja egy alkalmazás fejlesztésének. A régi módszerekkel szembenén is tapasztaltam ennek az előnyeit, a fejlesztés során igyekeztem a lehető legjobban szétválasztani az alkalmazás különböző részeit funkcionálisuknak megfelelően.

A HTML, mint „hiperszöveg jelölő nyelv” már a nevében érzékelteti a saját rendeltetését. A HTML-t régebben arra használták, hogy dokumentumokat tegyenek közzé, megadják ezek tartalmát és a felépítését, és definiálják a dokumentum különböző részeit.

A felhasználói felület megtervezésekor, a legegyszerűbb és a legátláthatóbb szerkezetet igyekeztem létrehozni, hogy megkönnyítsem ezzel az információ értelmezését. A grafikai megjelenítéshez a CSS által nyújtotta lehetőségekkel éltem. A CSS (Cascading Style Sheets) egy olyan leírónyelv, melynek segítségével különböző stíluslapokat hozhatunk létre és ágyazhatunk be HTML honlapjainkba. Ezek a stíluslapok befolyásolják az oldal megjelenését, lehetőséget ad a weboldal formázására és elrendezésének kialakítására. A CSS használható XML fájlok megjelenítésére is, így a strukturált dokumentumokhoz teljes körű stílust lehet megadni, befolyásolva az elrendezését, színét, betűtípusait. Beállíthatóka színek, a háttér, a betűk mérete és típusa, és megadhatóak a különböző elemek helyei a weblapon. Rengeteg előnye van annak, ha szétválasztjuk a tartalmat a formázástól.

Végül, a JavaScript dinamikus funkciókat biztosíthat a weblaphoz. JavaScript segítségével olyan kis programkódokat írhatunk a weblaphoz, amelyek a kliens számítógépén futnak lefutni, és nincs szükség semmilyen speciális szoftver telepítésére a kiszolgálón. A JavaScript lehetőséget ad arra, hogy néhány egyszerű funkcionálisit megvalósíthassunk a weblapon, és interaktívvá tehessek. A modern JavaScriptek már úgy működnek, hogy megkeresik a célzott HTML elemet, és ezzel csinálnak valamit, hasonlóan a CSS-hez, de a szintaxis és maga a művelet teljesen más ebben az esetben.

Tapasztalataim ezzel kapcsolatban:

- A CSS fájlok segítségével elég egyszer megadni a stílust és az elrendezést, ha módosításra van szükség, egy helyen, egy lépésben elvégezhetjük azt, egyszerűsödik a karbantartása.

- Hatékonyabbá tehetjük a kódot, ha a HTML-t a lehető legegyszerűbbre rövidítjük, a hozzá tartozó stílust és elrendezést pedig külön CSS fájlban adjuk meg. Ugyanis minél nagyobbak a fájlok, annál tovább tart a letöltésük. Ezzel gyorsabbá tesszük a letöltést, mert bár le kell töltenie a látogatónak néhány külön fájlt a dokumentum mellé, de ezt csak egyszer kell megtennie, mivel ezek a különböző lapokon újrafelhasználhatók, ezáltal összességében kevesebbet kell letölteni. Ráadásul a CSS és a JavaScript fájlok többnyire bekerülnek a gyorsítótárba, így amikor legközelebb látogatják meg az oldalt, a fájlok már ott lesznek a gépen, és nem kell újra letölteni őket.
- Ha egy alternatív stílust adunk meg, az elemek egyszerűen átformázhatók válnak a különböző tulajdonságokkal rendelkező eszközökön, például kisebb képernyőn. A CSS konzekvensen támogatja a különböző stílusok használatát a különböző megjelenítési módszerek és eszközök számára.
- Azért is hasznos megtisztítani a HTML-t amennyire csak lehet, mert a keresők, pl Google, keresőrobotok segítségével nézik át az oldalakat. Azonban ha a hasznos tartalmat nem lehet könnyen megtalálni a weblapon, az oldal hátrébb fog kerülni a találati listákon.

7. Tesztelés

Az oldal fejlesztése alatt folyamatosan teszteltem a helyes működést. Törekedtem arra, hogy bármely felmerülő hiba esetén idejében kiderüljön, hol van annak a forrása.

Az egyes problémák meghatározására és kiírására nagy segítséget jelentett a PHP néhány beépített függvénye. A „die” és a „mysql_error”, amelyek az adatbázishoz kapcsolódás során keletkező hibát a megadott sztring segítségével közlik.

Az egyik legfontosabb adatot, a jogosultságot a SESSION-be tároltam el, és ennekmeglétének ellenőrzésére felhasználtam a „print session_encode(;)” függvényt, mely kiírja az abban a pillanatban lévő SESSION tartalmát. Sajnos sokszor előfordult, hogy az egyes azonosítókat nem mindig sikerült eltárolni a SESSION-ben, vagy nem történt meg az átvitel egyik oldalról a másikra. A a GET és a POST metódusok segítségével lekérhettem az adatokat. Annak a lapnak az elejére, ahová a beviteli mezők értékei érkeznek, kiírtam ezeket a változókat. Így láttam, hogy megérkeznek-e az adatok. Ha ezek elküldése sikeres volt, akkor kezdtem csak hozzá az adattáblába való beillesztés leprogramozásához.

A problémák feltárásának egyik legjobb módszere, ha a fontosabb műveleteket if-else vizsgálatnak rendeljük alá. Egyszerű és bármely hiba esetén, pontosan meg tudjuk határozni a probléma forrását.

Összességében megkönnyítette a munkát az, hogy már a fejlesztés alatt az oldalt folyamatosan teszteltem és optimalizáltam. Így még időben kaptam hiteles képet arról, hogy az alkalmazás mennyire állja meg a helyét a gyakorlatban.

8. Bővítési lehetőségek

Célom volt diplomamunkám keretében egy olyan webes térképnyilvántartó-rendszert létrehozni, amely a tájékozódási futás versenyzőinek kiváló térképgyűjteménye lehet. Ez egy specifikus jegyzék, a rendszer fejlesztése közben azonban felvetődött a gondolat, hogy működése alapot adhatna egy komplex általános térképészeti adatbázishoz is. A bővítés mindenképpen megfontolandó, hiszen kevés változtatással nagyszerű katalógussá nőheti ki magát az alkalmazás.

Napjainkban a GPS, az internet világában sajnos egyre kisebb az igény a papír alapú térképekre. De vannak még olyan területek, ahol a digitalizálódás nem váltotta fel ezeket, a tájfutásnak, a túrázásnak még ma is elengedhetetlen eszközei. Ezért mindenképpen jó ötletnek találok azok gyűjteményét rendszerezni, népszerűsíteni.

Egy ilyen jellegű térképkatalógus sosem lehet teljes, így új és régi térképekkel folyamatos frissítés, rendszerkarbantartás szükségeltetik színvonalának emeléséhez.

9. Összegzés

A diplomamunka témaválasztásakor fontos szempont volt, hogy a meglévő térképészeti ismereteimet felhasználva egy olyan témával foglalkozhassak, amelynek segítségével sok új tapasztalatot gyűjthetek másik fontos érdeklődési körömmel, a web alapú alkalmazásfejlesztéssel kapcsolatban.

Lehetőségem nyílt a térképek új népszerűsítő eszközeül használnom a kor szellemének meghatározó, nagy befolyással bíró eszközét, az internetet.

A diplomamunkám témájához kapcsolódó webes alkalmazás fejlesztése során igenkülönböző technológiák illetve technikák együttes alkalmazására volt szükségem. Az elsőbbegyszerűnek tűnő felhasználói felületet és könnyű használhatóságot igénylő feladat egykihívásokkal teli munkává nőtte ki magát, ezek megoldására a következetes és átgondolt módszerek kitalálása magával ragadott. A feladat végére sikerült az előzetesen vártnál jobban elmélyülni a technológiákban. A munka egészét folyamatos tanulás és fejlődés jellemezte.

Remélem hasznos információforrás lesz a folyamatosan bővülő katalógus tájfutók, térképészek és laikusok számára egyaránt.

10. Felhasznált irodalom

LynnBeighley, Michael Morrison: Agyhullám: PHP &MySQL

Kiskapu, Budapest, 2009

ISBN: 978-963-9637-63-4

Nagy Gusztáv: Web programozás alapismeretek

Ad Librum Kiadó, Budapest, 2011

ISBN: 978-615-5110-26-9

EthanMarcotte, JeffreyZeldman: Szabványkövető webtervezés

Kiskapu, Budapest, 2011

ISBN: 978-963-9637-76-4

http://gepelmeny.blog.hu/2010/01/26/fogalmak_6

http://nagygusztav.hu/sites/default/files/csatol/webes_szabvanyok_jegyzet_0.1.pdf

<http://www.devcorner.hu/hogyan-szuletnek-a-designdontesek.html>

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

[http://hu.wikipedia.org/wiki/Ajax_\(programoz%C3%A1s\)](http://hu.wikipedia.org/wiki/Ajax_(programoz%C3%A1s))

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

Linkek utoljára ellenőrizve: 2012. június 7.

11. Köszönetnyilvánítás

Külön köszönetemet fejezem ki elsősorban Gede Mátyásnak, aki elvállalta a témavezető szerepét, felügyelte és javította munkámat, hasznos tanácsokkal látott el a dolgozat készítésével kapcsolatban, kritikai észrevételeivel, iránymutatásaival hozzájárult diplomamunkám eredményességéhez.

Köszönöm továbbá Zentai László tanár úrnak, hogy a téma iránt felkeltette érdeklődésemet.

Hálával tartozom ezen felül az ELTE Térképtudományi és Geoinformatikai Tanszék minden dolgozójának, hogy figyelmes munkájukkal szakmai ismereteim megszerzésében nagy segítséget nyújtottak.

Nyilatkozat

Alulírott, 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 benyújtásra elfogadott diplomamunka PDF formátumban való elektronikus publikálásához a tanszéki honlapon

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2012. június 8.

.....
a hallgató aláírása