EÖTVÖS LORÁND TUDOMÁNYEGYETEM INFORMATIKAI KAR

A Tisza-tó idegenforgalmi célú térinformatikai adatbázisa

DIPLOMAMUNKA TÉRKÉPÉSZ MESTERSZAK

Készítette: Eigner Péter

Témavezető: dr. Elek István habilitált egyetemi docens ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2019

Témabejelentő helye

Tartalomjegyzék

1.	Bev	rezete	és	3
2.	A fe	ejlesz	ztői környezet kialakítása	.4
2	.1	Sze	rveroldali szoftverkomponensek	. 4
	2.1	.1	Adatbázis-kezelő: PostgreSQL és PostGIS	. 5
	2.1	.2	Webszerver: Tomcat	6
	2.1	.3	Téradatszerver: GeoServer	. 7
	2.1	.4	PHP: Quercus	. 8
2	.2	Klie	ensalkalmazások és segédprogramok	. 8
2	.3	A re	endszer elvi működése	. 8
3.	Aza	adatb	pázis kialakítása	10
3	.1	Aza	adatok forrása	11
3	.2	Aza	adatbázis felépítése	13
4.	Téra	adat-	szolgáltatás GeoServerrel	14
4	.1	ΑW	VMS szolgáltatás	14
	4.1	.1	Méretarányfüggő jelkulcs SLD-vel	14
4	.2	ΑW	VFS szolgáltatás	18
4	.3	A sz	zerverteljesítmény fokozása	19
5.	Aza	adato	k vizualizációja webtérképen	20
5	.1	Aw	eboldal felépítése	20
5	.2	Aw	ebtérkép	21
	5.2	.1	A webtérkép alapvető jellemzői	21
	5.2	.2	A háttértérképi rétegek	23
	5.2	.3	A tematikus rétegek	24
6.	Tov	ábbf	ejlesztési lehetőségek	29
7.	Öss	zegz	és	30
8.	Irod	lalon	njegyzék	31
9.	Ábr	ajeg	yzék	33
10.	Kös	zöne	tnyilvánítás	34
11.	Mel	lékle	etek	35
12.	Nyi	latko	zat	39

1. Bevezetés

Diplomamunkám témájához egy korábbi egyetemi projektmunkám szolgáltatta az alapötletet. Az akkori feladat (amelyen Huszánk Daniella hallgatótársammal közösen dolgoztunk) eredeti célkitűzése egy olyan adatbázis megalkotása volt, amely a Tisza-tó idegenforgalmát, vízügyi, horgászati és vízisport-infrastruktúráját dolgozza fel, továbbá a munka későbbi folytatásaként papírtérképen ábrázolja is az adatokat.

A kitűzött célok azonban az időszűke miatt csak részben valósultak meg. Az adatgyűjtési is digitalizálási munkákon túl az adatok adatbázisba rendezésére már nem volt módunk. Ekkor hívta fel figyelmemet a projektfeladat akkori irányítója, illetve jelenlegi témavezetőm Elek István a feladatban rejlő további lehetőségekre. Konzultációink során arra a megállapításra jutottunk, hogy bár a munka végső célja egy analóg papírtérkép előállítása lenne (azaz klasszikus kartográfiai feladat), érdemes lehet a témát tisztán térinformatikai szemszögből is vizsgálni. Így a projekt ezen a ponton két ágra szakadt. Jómagam a témában rejlő térinformatikai lehetőségek feltárását kívántam megvalósítani, míg a feladat eredeti céljainak kidolgozását Kulcsár Hajnalka folytatta, amelynek részleteit *A Tisza-tó idegenforgalmi térképe című* diplomamunkájában ismerteti.

Dolgozatom célja tehát egy olyan térinformatikai rendszer tervezése, amelynek szerves részét képezi az az adatbázis, amely a korábban gyűjtött adatokból, illetve azok további bővítéséből és pontosításából áll. Erre az adatbázisra egy olyan webalkalmazást kívánok (közvetett módon) ráépíteni, amely az adatok vizuális megjelenítését is megvalósítja interaktív lehetőségekkel. Vagyis a végtermék az eredeti analóg célokkal szemben egy digitális webtérkép.

Ennek a rendszernek a megépítése egy olyan fejlesztői környezet meglétét igényli, ami a fent megfogalmazott célok megvalósítását teljes mértékben lehetővé teszi. Így magam kívánom összeállítani azt a szoftveres infrastruktúrát is, ami a munka informatikai kereteit szolgáltatja. Munkám során különös hangsúlyt szeretnék fektetni a nyílt forráskódú (open source) eszközök használatára.

2. A fejlesztői környezet kialakítása

A bevezetésben felvázolt feladat hatékony megoldásához nélkülözhetetlen a megfelelően kialakított, jól összehangolt fejlesztői környezet. A webfejlesztői munka során már bevett rutinnak tekinthető, hogy a weboldal készítője rendszerint egy virtuális környezetben fejleszti és teszteli munkáját. Egy ilyen virtuális munkakörnyezetben lehetőség van arra, hogy leellenőrizzük a weblap struktúráját, a népszerű böngészőkben való megjelenését, vagy épp az adatbázis kimeneti eredményeinek helyességét még az éles beüzemelés előtt.

A fejlesztés során én is ezt a gyakorlatot kívántam követni, így saját munkakörnyezetemet is egy virtuális térben építettem ki. A legpraktikusabb és a hordozhatóság szempontjából is legideálisabb megoldás erre a célra egy virtuális gép létrehozása. Saját virtuális gépemet a **VirtualBox** szoftver segítségével hoztam létre, amely az Oracle ingyenes virtualizációs alkalmazása.

Az virtuális gépen az **Ubuntu 16.04** LTS Xenial Xerus operációs rendszer fut. Az Ubuntu a Linux egyik Debian alapokon nyugvó disztribúciója, vagyis a rendszer alapja egy Linux kernel (Linux 4.2). A disztribúció ingyenes és nyílt forráskódú, továbbá frissességéről és az új informatikai megoldásokra való gyors reagálásáról ismeretes.

A fejlesztői környezet összeállítása általában három részre bontható. A futtató környezet kialakítására (szerverek), a megfelelő kliensek telepítésére, valamint a szerkesztőprogramok és egyéb szoftverek telepítésére. Az így kialakult szoftvergyűjteményt szokás fejlesztői csomagnak (angolul development stack-nek) is hívni. Az ebben szereplő komponensek logikailag általában egymás alatt helyezkednek el, különböző logikai rétegeket alkotva. (Tarcsi-Horváth, 2012)

2.1 Szerveroldali szoftverkomponensek

A szerver kialakítása során minden komponens telepítését saját magam végeztem. Ha ilyen módon végezzük a szerver összeállítását, számolnunk kell azzal, hogy ez egy sokkal időigényesebb folyamat, mint az előre összeállított szoftvercsomagok használata, továbbá megfelelő szakértelmet is igényel. Ezzel a módszerrel azonban a legapróbb részletekre odafigyelve végezhető el a szoftverek testreszabása és összehangolása, így végül egy átlátható és teljes mértékben konfigurálható rendszert kapunk. A rendszer összetevői szerver oldalon az adatbázis-kezelő, a webkiszolgáló, a téradat-kiszolgáló, illetve a PHP.

2.1.1 Adatbázis-kezelő: PostgreSQL és PostGIS

A **PostgreSQL** (más néven Postgres) egy korszerű objektumrelációs adatbázis-kezelő rendszer, ahol biztonságosan tárolhatjuk adataidat komplex felhasználás mellett is. Fejlesztése a Berkeley Egyetem Számítástudományi Tanszékének vezetésével zajlik. (Nyíri, 2003) Nyílt forráskódjának köszönhetően több módon lehet kapcsolódni hozzá. Teljesítménye és rugalmassága a piacvezető kereskedelmi adatbázis-kezelőkkel (Oracle Database, IBM DB2) vetekszik.

A **PostGIS** a PostgreSQL adatbázis-kezelőrendszernek egy kiegészítő modulja, amely geometriai adatok és raszteres adatok tárolását támogatja. (Nguyen, 2012) A PostGIS térinformatikai képességekkel bővíti a Postgrest, ezzel lehetővé téve, hogy térinformatikai rendszerek és webtérkép-alkalmazások háttéradatbázisa legyen. A PostGIS egy stabil, gyors, szabvány követő szoftver, a térinformatikai függvények százaival, amely jelenleg a legszélesebb körben használt nyílt forráskódú téradatbázis. (OSGeoLive)

A Postgres telepítése PostGIS bővítménnyel:

Az adatbázis-kezelőt és a hozzákapcsolódó PostGIS modult az alábbi módon telepítettem az Ubuntu XTerm alkalmazásával (hagyományos parancssoros telepítés).

Rendszerfrissítés:

sudo apt-get update

PostgreSQL telepítése:

sudo apt-get install postgresql postgresql-contrib

Felhasználói jelszó beállítása:

sudo passwd postgres

Jelszó megadása SQL parancson keresztül:

```
su - postgres
psql -d template1 -c "ALTER USER postgres WITH
PASSWORD 'új jelszó';"
```

PPA csomagtároló (Personal Package Archive) felvétele a PostGIS telepítéséhez:

```
sudo add-apt-repository ppa:ubuntugis/ppa
```

PostGIS telepítése:

sudo apt-get install postgis

PostGIS bővítmény hozzáadása a Postgres megfelelő adatbázisához:

```
postgres=# \c DATABASE_NAME
CREATE EXTENSION postgis;
CREATE EXTENSION postgis_topology;
```

2.1.2 Webszerver: Tomcat

Az Apache Tomcat (más néven Jakarta Tomcat) egy webes tároló, amely lehetővé teszi a Sun-féle Java Servletek és JSP (Java Server Pages) webalkalmazások kiszolgálását. (Ubuntu dokumentáció) Az *Apache Servert* és az *Apache Tomcat*-et gyakran keverik össze a nevükben lévő hasonlóság miatt. Fontos ezért megjegyezni, hogy míg az Apache Server egy tisztán HTTP webszerver, addig az Apache Tomcat egy olyan webkonténer, amely Java kódot futtat. További különbség, hogy az Apache Server statikus tartalmak kiszolgálására szolgál, míg a Tomcat elsősorban dinamikus tartalmakhoz használható. A Tomcat ugyanakkor rendelkezik egy beépített HTTP szerverrel is, tehát a Java szervletek kezelésén túl teljesértékű webszerverként is üzemeltethető. A motor teljes mértékben Java alapokon nyugszik, és a program fejlesztői nyílt forráskódú termékként kínálják.

A Tomcat telepítése:

A Tomcat webszerver telepítésének lépései terminálban:

Rendszerfrissítés:

```
sudo apt-get update
```

Tomcat telepítése:

sudo apt-get install tomcat8

További webalapú Tomcat eszközök telepítése:

```
sudo apt-get install tomcat8-docs tomcat8-examples
    tomcat8-admin
```

2.1.3 Téradatszerver: GeoServer

A GeoServer egy nyílt forráskódú, Java-ban írt szerverszoftver, amely lehetővé teszi a felhasználók számára, hogy megosszák és szerkeszthessék a térinformatikai adataikat. (GeoServer Felhasználói Kézikönyv) Segítségével lehetővé válik, hogy különböző formátumú térképeket és térbeli adatokat szolgáltassunk olyan szabványos klienseknek, mint a webböngészők és asztali GIS programok. Az adatokat az Open Geospatial Consortium (OGC) szabványokon alapuló interfészeken keresztül publikálja, mint például Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Web Processing Service (WPS) és továbbiak. Adatokat vektoros és raszteres adattárakból egyaránt képes szolgáltatni, úgy mint:

- vektoros: shapefile-ok, külső WFS, PostGIS, Oracle Spatial, MySQL, SQL Server
- raszter: GeoTiff, JPG, kép piramis, GDAL formátumok, kép mozaik, Oracle GeoRaster

A GeoServer telepítése:

A GeoServer telepítése előtt fontos meggyőződnünk róla, hogy számítógépünkön rendelkezésre áll a megfelelő Java Runtime környezet (JRE) vagy Java Developement Kit (JDK), mert enélkül a GeoServer nem képes működni.

Az **OpenJDK** parancssoros telepítését a következőképpen végeztem.

Rendszerfrissítés:

```
sudo apt update
```

OpenJDK telepítése:

sudo apt install default-jre

Java verzió ellenőrzése:

java -version

A GeoServer telepítése Linuxon egyszerűen elvégezhető. Az installáláshoz mindössze a megfelelő Web Archive (WAR) fájlt kell letöltenünk a GeoServer weboldaláról, és a webszerverünk megfelelő mappájába elhelyezni (esetünkben a Tomcat webapps mappájába).

2.1.4 PHP: Quercus

A PHP-t érdemes modulként telepíteni a webszerverre. Erre kínál megoldást a Caucho cég **Quercus** névre keresztelt terméke, amely egy Java Servlet alapú PHP implementáció. Működése közben a Quercus a .php állományokat automatikusan a QuercusServlet-nek továbbítja, amennyiben azokat a root könyvtárba helyeztük.

A Quercus telepítése:

Telepítése Tomcat alatt a GeoServerrel megegyező módon történik. A szükséges .war fájl a Caucho holnapján szabadon hozzáférhető.

2.2 Kliensalkalmazások és segédprogramok

pgAdmin

A PostgreSQL adatbázis-kezelő adminisztrációjához egy jól használható asztali kliensalkalmazás a pgAdmin. A szoftver a Postgres grafikus felületű segédprogramjának tekinthető.

shp2pgsql-gui

A PostGIS tartalmaz egy shp2pgsql parancsot, amely lehetővé teszi, hogy shapefile-okat alakítsunk .sql állománnyá. Ezek az SQL állományok tartalmaznak minden szükséges információt, ami a forrásállománynak megfelelő PostgreSQL tábla létrehozásához és az adatok importálásához szükséges. A shp2pgsql alapvetően parancssorosan használható, ugyanakkor létezik egy shp2pgsql-gui program is, ami -mint a nevéből is következik-, a fent leírt feladatokhoz kínál grafikus felületet.

2.3 A rendszer elvi működése

A megjelenítendő térképi adatok forrása vektoros és raszteres fájlok együttese. A gyakorlatban ez shapefile-okat és GeoTIFF állományokat jelent, amelyek előállítása QGIS program segítségével történt. A vektoros adatok a shp2pgql-en keresztül töltődnek be a PostGIS bővítménnyel ellátott PostgeSQL-be és itt rendeződnek adatbázissá. Ezt az adatbázist használja forrásként a GeoServer. A raszteres állományok közvetlenül a GeoServerbe töltődnek.

A GeoServer -amely egy a Tomcat konténerbe elhelyezett webmodul (szervlet)-, ezután WMS és WFS szolgáltatásokon keresztül továbbítja a térbeli adatokat az OpenLayers-szel létrehozott webtérkép számára. A téradatszerver közbeiktatására azért van szükség, mert a PostGIS adatbázisok és az OpenLayers közötti közvetlen kommunikáció jelenleg nem lehetséges. Ezt a problémát hivatott áthidalni a GeoServer.

Végül az OpenLayers a szervertől kapott vektoros (WFS) és raszteres (WMS) adatokat grafikusan is megjeleníti.

A fentiekben vázolt folyamatot az *1. ábra* szemlélteti, amelynek részletes kifejtésére a dolgozat következő fejezeteiben kerül sor.



1. ábra: A rendszer architektúra diagramja

3. Az adatbázis kialakítása

A megvalósítandó térinformatikai rendszer alapját képezi az az adatbázis, amely a Tisza-tó idegenforgalmára, vízügyi, és horgászati infrastruktúrájára vonatkozóan tartalmaz adatokat. Az adatbázis a PostGIS kiegészítővel ellátott Postgres szerveren készült és a localhost 5432-es portján üzemel.

Az adatbázis kizárólag vektoros térinformatikai adatokat tartalmaz. Az adatok bemeneti formátuma a shapefile. A shapefile az egyik legnépszerűbb adatcsereformátum az egyes geoinformatikai alkalmazások között, ami egyrészt nyílt specifikációjának, másrészt rendkívüli elterjedtségének köszönhető. A forrásfájlok adatbázisba importálását a shp2pgsql segédprogrammal végeztem (2. *ábra*). A bemeneti shapefile az adatbázisban egy önálló táblaként jelenik meg.

View connect	ion deta	ils				
port Export						
mport List						
Shapefile	Schema	Table	Geo Column	SRID	Mode	Rm
/home/drupal/Letöltések/drive/retegek-3857/tematikus/horg	public	horgaszhelyek	geom	0	Create	
/home/drupal/Letöltések/drive/retegek-3857/szallashely.shp	public	szallashely	geom	0	Create	
/home/drupal/Letöltések/drive/retegek-3857/utak.shp	public	utak	geom	0	Create	
Add	rne					
Ontions		About			Cancel	
Options Import						
a Window						
g Window innectina: host=localhost port=5432 user=postgres password		*' dbname=post	ares client en	codine	=UTF8	
g Window nnecting: host=localhost port=5432 user=postgres password nnection succeeded.	.'******	*' dbname=post	gres client_en	coding)=UTF8	
g Window nnecting: host=localhost port=5432 user=postgres password nnection succeeded.	.'******	*' dbname=post	gres client_en	codin)=UTF8	

2. ábra: Shapfile-ok importálása az adatbázisba a shp2pgsql segítségével

A shp2pqsql-el végzett betöltési folyamat egy összetett művelet, ugyanis a segédprogram nem pusztán az adatimportot hajtja végre, de egyúttal az adatok térbeli indexelését is elvégzi. A program a térbeli indexet az újonnan létrehozott tábla geom oszlopához rendeli hozzá, meggyorsítva ezzel az olyan jellegű lekérdezéseket, amikor az adatbázisból azokat az elemeket szeretnénk lekérdezni, melyek egy megadott területre esnek. Ezek a lekérdezések rendkívül gyakoriak webtérképes alkalmazásnál, valahányszor ki akarjuk rajzolni egy terület térképének egy részletét. Térbeli index nélkül egy ilyen lekérdezéshez át kellene vizsgálni a tábla összes rekordját, hogy valóban a kérdéses területre esik-e az objektum. Ez a megoldás nem túl hatékony, különösen nagyméretű táblák esetén. A térbeli index segítségével a program előszűrést tud végezni, amivel gyorsan kiszórhatja azokat az elemeket, melyek a vizsgált területnek még a közelében sincsenek. A PostGIS a térbeli indexelésre a GiST (Generalized Search Tree, magyarul: egyszerűsített keresőfa) algoritmust használja. (HupWiki, 2009)

3.1 Az adatok forrása

Az adatbázisban tárolt adatok több különböző forrásból származnak. Egyrészük meglévő raszteres állományok alapján történt digitalizáció, további részük önálló internetes adatgyűjtés, ezen felül pedig "kész" adatgyűjtemények, amelyek a Közép-Tisza-vidéki Vízügyi Igazgatóságtól (KÖTIVIZIG) származnak.

A háttértérképi adatok előállítása digitalizációs eljárás útján történt, amelynek alapja raszteres műholdfelvételek és légifotók. A digitalizáció munkafolyamatát Kulcsár Hajnalka és Varga Csaba Gergely hallgatótársaimmal közösen végeztem az alábbi munkamegosztásban:

- Eigner Péter: növényzeti fedettség és vízfelületek digitalizálása a Tiszató partvonalán belüli területekre a KÖTIVIZIG által biztosított nagy felbontású légifotók alapján. (3. ábra)
- Kulcsár Hajnalka: növényzeti fedettség digitalizálása a Tisza-tó partvonalán kívüli területekre, továbbá úthálózat
- Varga Csaba Gergely: épített fedettség digitalizálása



3. ábra: Ugyanazon földrészlet légifotón és digitalizálva saját jelkulccsal

Az idegenforgalmi tematikát szolgáló adatok önálló internetes adatgyűjtés eredményei. Az adatok forrása számos különböző weboldal, települések, vállalkozások honlapjai, szállásközvetítő oldalak stb. Ezzel a módszerrel készültek a következő tematikus rétegek:

- Bankok és ATM-ek: kereskedelmi bankfiókok és bankautomaták nyitva tartásukkal.
- Egészségügy: orvosi rendelők vagy praktizáló orvosok, illetve gyógyszertárak nyitvatartási vagy rendelési idővel.
- Éttermek: nagyobb kereskedelmi vendéglátóhelyek elérhetőségeikkel és nyitvatartási idejükkel.
- Látnivalók: múzeumok, tárlatok és egyéb turisztikai érdekességek.
- Strandok és kikötők: szabadvízi és épített fürdőhelyek, illetve hajókikötők.
- Szálláshelyek: különböző típusú szállás szolgáltatást nyújtó létesítmények a wellness szállodáktól kezdve a panziókon át, a vendégházakig.
- Templomok: vallási épületek és gyülekezeti helyek a mise / istentisztelet rendjével (amennyiben ismert).
- Tömegközlekedés: vasúti és autóbusz-megállók, illetve állomások / pályaudvarok
- Üzemanyagtöltő állomások: benzinkutak elérhetőségeikkel és nyitvatartási idejükkel.

A Közép-Tisza-vidéki Vízügyi Igazgatóságtól több alapanyag is származik, amelyek közül a raszteres állományok a következők:

- Infra ortofotó: Álló helyzetű 1 : 20 000-es méretarányú ortofotó a tóról és környékéről. Az infrafotó sokkal szemléletesebben mutatja be a növényzetborítottságot. A papír alapú térképet szkennelés után önállóan georeferáltam.
- Műszaki vízügyi térkép: Vízügyi szakmai adatokat tartalmaz feketefehér vázlatos formában.

Légifelvételek: A tó jelenlegi állapotát leghitelesebben bemutató adatforrás. A tavat alacsony vízálláskor mutatja be. Nagy felbontása pontos növényzetfedettség-ábrázolást tesz lehetővé. Hátránya, hogy a felvételek kiterjedése kizárólag a tó partvonalán belüli részekre korlátozódik. (Részlet az 3. ábrán)

A KÖTIVIZIG a raszteres források mellett vektoros adatgyűjteményt is biztosított, amely a vízilétesítmények, árvízvédelmi műtárgyak helyzetére és tulajdonságaira vonatkozó adatokat, valamint a tavon belüli hajózás szabályozásához kapcsolódó információkat tartalmazta, úgy mint:

- közlekedési tiltások
- hajóutak
- viharjelzők
- gátőrházak
- sorompók
- emlékhelyek

3.2 Az adatbázis felépítése

A 3.1-es alfejezetben ismertetett források a következőképpen rendeződtek adatbázisba. A vektoros adatokat tartalmazó PostGIS adatbázis az alábbi táblákból épül fel:

- alap kozl_tilt to₂ bankok
- egeszsegugy
- epitett_fedettseg
- etterem
- hajoutak
- horgaszhelyek
- kitakaro
- kitakaro_nagy

- megallok
- muzeumok
- nevrajz
- novenyzet
- strand_kikoto
- szallashely
- telepulesek
- templomok

Ezen felül közvetlenül a GeoServerhez csatlakoznak a következő raszteres állományok:

Infra ortofotó Műszaki vízügyi térkép

- toltoallomasok
- uj hatar
- . utak
- vasut
- vizfeluletek
- vizugy

4. Téradat-szolgáltatás GeoServerrel

A Postgresben létrehozott adatbázis a GeoServeren, mint téradatszerveren keresztül kerül publikálásra. A GeoServer közvetlenül csatlakozik a PostGIS adatbázishoz, és azt az adatok forrásaként kezeli. Ezután az adatokat WMS és WFS protokollok használatával szolgáltatja a kliens, vagyis a webtérkép felé.

4.1 A WMS szolgáltatás

A webtérkép azon rétegei, amelyekhez interaktív funkció nem kapcsolódik, WMS szolgáltatáson keresztül továbbítódnak az OpenLayers felé. Ezek a rétegek döntően a webtérkép háttértérképi rétegei (pl.: növényzeti fedettség, úthálózat, vízrajz stb.).

Az OGC **Web Map Service** (WMS) egy szabványos protokoll, amely segítségével a térképszerver térinformatikai adatbázison alapuló adatokat szolgáltat az interneten. (Pődör, 2010) Egy WMS szerver az elérhető téradatbázis rétegeit oly módon publikálja, hogy a kívánt térképi részletből a szerver oldalon egy raszteres képet állít elő és azt küldi át a kliensnek. Ezzel a megoldással viszonylag kis adatforgalom mellett lehet nagy térinformatikai adatbázisokat az interneten keresztül nyilvánossá tenni. (Siki, 2010)

4.1.1 Méretarányfüggő jelkulcs SLD-vel

A WMS szolgáltatás használatakor a kliens a GetMap műveleten keresztül kérheti a kiválasztott területre vonatkozó térkép visszaküldését. Tekintetbe véve, hogy a szerver válasza a GetMap kérésre minden esetben egy raszteres kép (kvázi egy "kész térkép"), szükségszerű, hogy előre meghatározzuk a téradatok képi megjelenítésére vonatkozó stílusszabályokat.

A GeoServer a stílusok definiálására a **Styled Layer Descriptor** (SLD) szabványt használja. Az SLD egy olyan XML sémanyelv, amellyel a térképi rétegek kliens oldali megjelenítése írható le.

A webtérképek legalapvetőbb sajátossága, hogy méretarányuk a felhasználó igényei szerint változtatható (zoomolható). Fontos emiatt, hogy a megjelenített térkép jelkulcsa mindig az aktuális nagyítási szinthez igazodjon. Az SLD-vel leírt stílusok esetén ez úgy lehetséges, hogy minden előre meghatározott méretarány-tartományhoz önálló jelkulcsot definiálunk.

Webtérképemen a következő méretarány-tartományokat szabtam meg a jelkulcs kialakításakor:

- 1:5000-1:10000
- 1:10 000 1:20 000
- 1:20000-1:40000
- 1:40 000 1:80 000
- 1:80 000 1:160 000
- 1:160 000 1:360 000

Minden egyes határ a tartományok között egy generalizálási küszöbnek tekinthető. Ennek a jelkulcs meghatározási módszernek több háttértérképi rétegnél, de legfőképp az úthálózat és a növényzeti fedettség megjelenítésénél van különös jelentősége. Az utak esetén a vonalak száma, a vonalas jel szélessége és a vonal jellege, míg a növényzet esetén a felületi színt kiegészítő felületjel mérete és sűrűsége változik a méretarány függvényében.

Összetett SLD stílus felület típusú elemre:

A Tisza-tó környékének növényzeti fedettségét az adatbázis *novenyzet* táblája írja le. Az egyes növényzeti kategóriák között a *tipus* attribútum tesz különbséget. A növényzet stílusának összetettsége abban rejlik, hogy az SLD fájl készítésekor nem pusztán az egyes fogalmi kategóriák rajzi elkülönítését kell megvalósítanunk, de a kategórián belül több stílust is definiálnunk kell a méretarányoknak megfelelően.

A növényzet ábrázolása több ábrázolási módszer (felületi szín és felületjel) együttes alkalmazásával valósul meg, vagyis egy jelkulcsi jel több rétegből áll össze. Minden egyes stílusrétegréteget külön-külön szabályok írnak le SLD-ben.

A következő SLD részlet a felületi szín leírását tartalmazza.

```
...
<se:FeatureTypeStyle>
<se:Rule>
<se:Name>nadas0</se:Name>
...
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
<ogc:PropertyIsEqualTo>
```

```
<ogc:PropertyName>tipus</ogc:PropertyName>
<ogc:Literal>nadas</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
<se:PolygonSymbolizer>
<se:Fill>
<se:SvgParameter name="fill">#b5e1e0</se:SvgParameter>
</se:Fill>
</se:PolygonSymbolizer>
</se:Rule>
</se:FeatureTypeStyle>
...
```

A fenti részleten látható, hogy a szabály <ogc: Filter> paramétere gondoskodik arról, hogy ez a stílus csak a *nadas* típusú elemekre vonatkozzon. A szabály a méretarány-függő megjelenítésre vonatkozóan nem tartalmaz kikötéseket, hiszen a felületi szín a nagyítási szinttől függetlenül változatlan.

A felületi színt kiegészítő felületjel ugyanakkor a méretaránnyal együtt változik. A felületjeleket CorelDraw programmal rajzoltam meg, és belőlük raszteres (png) kitöltő mintákat hoztam létre, minden méretaránytartományhoz különböző jelmérettel és jelsűrűséggel (*4. ábra*). Ez a minta szabályosan ismétlődik a felületi szín rétege fölött, háttere átlátszó. A jelek elforgatott állása azért indokolt, mert a munkafolyamat végén a kész



4. ábra: Nádas kitöltőminta a 1:0000-nél nagyobb méretarányokban

webtérkép, 52°-os elforgatással (tehát nem északra tájolva) fog megjelenni. Mivel a raszteres WMS rétegek a térképpel együtt elforognak, gondoskodni kell róla, hogy a jelek a végterméken egyenes állásban jelenjenek meg. Hogy ezt biztosítsam, az eredeti, északra tájolt WMS rétegen a jelek -52°-os elforgatással láthatók. A felületjelek rétegét a következő SLD-szabály írja le:

```
<...
<se:FeatureTypeStyle>
<se:Rule>
<se:Name>nadas4</se:Name>
...
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
</ogc:Filter xmlns:ogc="http://www.opengis.net/ogc"/model"/www.opengis.net/ogc"//www.opengis.net/ogc"//www.opengis.net/ogc"//www.opengis.net/ogc"//w
```

Látható, szabályt túl hogy ezt az <ogc:Filter>-en a а <se:MinScaleDenominator> és <se:MaxScaleDenominator> paraméterek is kiegészítik. Ezek határozzák meg a réteg megjelenítésének méretaránykorlátait. A <se:ExternalGraphic> a külső forrásból származó kitöltő minta linkjét tartalmazza. Ez a szabály a nádas és más növényzeti kategóriák esetén is többszörösen ismétlődik különböző méretaránykorláttal és kitöltő mintával. Mindezen szabályok együttese az 5. ábrán látható jelkulcsot eredményezi a nádas kategóriára vonatkozóan. A teljes növényzeti jelkulcsot a 1. sz. melléklet mutatja be.



5. ábra: A nádas növényzeti kategória ábrázolása különböző méretarányokban

Összetett SLD stílus vonalas elemre:

Az adatbázis *utak* táblája tartalmazza az úthálózatra vonatkozó információkat. Az utak réteg egyes úttípusainak jele is több stílusréteggel kerül ábrázolásra. Ez azért szükséges, mert az SLD szabvány nem ismeri a vonalkitöltés és vonalkontúr fogalmát. Így azokat az útkategóriákat, amelyeknek jele ebből a két rajzi alegységből tevődik össze, SLD-vel két különálló szabályként kell leírnunk. A két szabály közül az "alsó" vastagabb vonal képezi a kontúrt, a rajta elhelyezkedő vékonyabb pedig a kitöltést. Így a vonalkontúr szerepét betöltő vonal stílusszabálya a következőképpen alakul:

```
. . .
<se:FeatureTypeStyle>
 <se:Rule>
   <se:Name>kontur_fout5</se:Name>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:PropertyIsEqualTo>
       <ogc:PropertyName>tipus</ogc:PropertyName>
       <ogc:Literal>fout</ogc:Literal>
     </ogc:PropertyIsEqualTo>
   </ogc:Filter>
   <se:MinScaleDenominator>10000</se:MinScaleDenominator>
   <se:MaxScaleDenominator>20000</se:MaxScaleDenominator>
   <se:LineSymbolizer>
     <se:Stroke>
       <se:SvgParameter name="stroke">#e31a1c</se:SvgParameter>
       <se:SvgParameter name="stroke-width">6.4251968504</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
       <se:SvgParameter name="stroke-linecap">round</se:SvgParameter>
     </se:Stroke>
   </se:LineSymbolizer>
 </se:Rule>
</se:FeatureTypeStyle>
```

Míg a kitöltést biztosító vonal szabálya a fentihez hasonló, de más vonalszín és vastagság paraméterekkel. Ezek a kontúr-kitöltés szabálypárok minden méretaránytartományhoz külön megfogalmazásra kerülnek. Így például a főutak útkategória esetén az 6. *ábrán* látható jelkulcsi jeleket kapjuk. A teljes úthálózati jelkulcsot a 2. *sz. melléklet* mutatja be.



6. ábra: A főutak útkategória ábrázolása különböző méretarányokban

4.2 A WFS szolgáltatás

A webtérkép interaktív funkcióval felruházott rétegeinek adatait Web Feature Service (WFS) protokoll szolgáltatja az OpenLayers-nek. A WFS egy olyan interfészt kínál, amely az adatbázisban foglalt adatok vektoros állományként történő elérését teszi lehetővé. Hasonlóan a WMS-hez, a WFS használatakor is interneten (HTTP interfészen) keresztül érhető el a GeoServer által közvetített téradat. Azonban míg a WMS az adatforrás adatmodelljétől (vektoros vagy raszteres) függetlenül mindenképpen raszteres formában teszi elérhetővé az adatot, addig a WFS szolgáltatáson keresztül vektoros formában férhetünk hozzá a téradathoz, aminek ebből fakadóan csak vektoros adatmodellel tárolt adat lehet a forrása. (Ónodi, 2017) A WFS szolgáltatáson keresztül elért rétegek a geometriai adatok mellett attribútum adatokat is tartalmaznak, de a megjelenítésre vonatkozó stílusszabályokat nem. Jelkulcsi beállításaik kliens oldalon történnek.

4.3 A szerverteljesítmény fokozása

A GeoServer által nyújtott WMS és WFS szolgáltatások megfelelő sebessége és hatékonysága kulcsfontosságú a későbbi felhasználásuk szempontjából. A nem megfelelően működő, lassú WMS alkalmazása nem csak a felhasználó számára kényelmetlen, de sokszor teljesen ellehetetleníti a vele való munkát. A WMS szolgálatások (és a teljes GeoServer) hatékonyabb működésére több megoldás is létezik, melyek közül munkámban a következőket alkalmaztam.

1. Natív JAI telepítése

A GeoServer a Java Advanced Imaging (JAI) keretrendszert használja a raszteres kép kirajzolására (renderelésre). Helyes telepítés esetén, a JAI keretrendszer felgyorsíthatja a WMS szolgáltatások sebességét.

2. A szerver memóriájának beállításai

A GeoServer teljesítménye nagyban függ a futtató környezet (Java webkonténer) dinamikusan foglalt memóriájának méretétől (Java Heap Space). A rendszer alapbeállításai gyakran csak akkora memória felhasználását engedik, ami a szerver működéséhez ugyan elegendő, de sebessége sokszor kifogásolható. Ha a kevés memóriából eredő futásidejű hibákat tapasztalunk, megoldást jelenthet a Java Heap Space növelése. Ennek módja a következő:

- A {tomcat_install}\bin mappában létrehozunk egy setenv.sh nevű és kiterjesztésű fájlt.
- 2. A fájl tartalmát a következőre módosítjuk:

```
export CATALINA_OPTS="$CATALINA_OPTS -Xms512m"
export CATALINA_OPTS="$CATALINA_OPTS -Xmx8192m"
export CATALINA OPTS="$CATALINA OPTS -XX:MaxPermSize=256m"
```

5. Az adatok vizualizációja webtérképen

Dolgozatom végeredménye egy olyan térképes webalkalmazás, amelyen az adatbázisban összegyűjtött és GeoServer által közvetített adatok grafikusan megjeleníthetők, és interaktívan böngészhetők.

A webtérképet tartalmazó weboldal **HTML** (Hypertext Markup Language, hiperszöveges jelölőnyelv) nyelv használatával íródott. A HTML egy egyszerű szövegfájlban írja le a szövegek, képek és videók alkotta dokumentumok (hiperszövegek) szerkezetét, meghatározott szabályok szerinti parancsokkal (HTML kódokkal).

A weblap formai kinézetét a **CSS** (Cascadium Style Sheets) stílusleíró nyelv segítségével alakítottam ki. A CSS megkönnyíti weboldal stílusának kialakítását a szövegformázástól kezdve a színek meghatározásáig, de az weblapok strukturálásában is szerepet játszanak. (Paczona, 2001)

Tekintve, hogy a HTML egy szöveges kódnyelv, nem pedig programozási nyelv, nem képes válaszolni a felhasználó kéréseire, vagy automatikusan végrehajtani különböző feladatokat. (Moncur, 2002) Ezért erre a célra **JavaScriptet** használtam, ami az első és máig legnépszerűbb webes parancsnyelv, így a weboldalam összetettebb funkciói is ezen a nyelven íródtak.

5.1 A weboldal felépítése

A weblap szerkezetének tervezésekor az az irányelv vezérelt, hogy a weboldal leglényegesebb eleme, a webtérkép a rendelkezésre álló legnagyobb felületen megjelenjen, az egyéb kiegészítő elemek pedig ne foglaljanak több helyet a szükségesnél.

Követve a fenti szempontokat az oldal szerkezete 7. *ábrán* látható módon épül fel. A weboldal megjelenése egyszerűséget tükröz, egy fejlécből és menüsorból (1), illetve a térképi felületből (2) áll.

A menüsor elemeire kattintva felugró ablakokban jelennek meg a további lehetőségek. A menüsor elemei a következők:

- 1/a Rétegek: a felugró ablakban a felhasználó a megjeleníteni vagy elrejteni kívánt tematikus rétegeket állíthatja be
- 1/b Jelmagyarázat: a háttértérképi és tematikus rétegek jelkulcsának egyszerűsített kivonata
- 1/c Névjegy: általános információk a weboldalról és készítőjéről
- 1/d Súgó: tanácsok és útmutató az alkalmazás könnyebb használatára vonatkozóan



7. ábra: A weboldalt felépítő egységek

5.2 A webtérkép

A weboldal térképes felületének megalkotásához az **OpenLayers 5.3.0**-ás verzióját használtam. Az OpenLayers egy JavaScript nyelven készült programkönyvtár dinamikus térképek létrehozásához, így felhasználása is JavaScript környezetben lehetséges.

5.2.1 A webtérkép alapvető jellemzői

A kivágat megválasztása

Az ábrázolni kívánt terület lehatárolását két lépésben választottam meg. Az első lépésben egy olyan téglalapot kívántam meghatározni, amely a Tisza-tó teljes területét magába foglalja és délkelet-északnyugati kiterjedése a Kisköre-Abádszalók vonaltól Tiszadorogmáig tart. Tiszadorogma ugyan a Tisza-tótól valamelyest távolabb fekszik, de jelentősége mégis számottevő a kompközlekedés miatt. A továbbiakban ezt a téglalapot úgy bővítettem különböző kitörésekkel, hogy a tó partján vagy közvetlen közelében lévő minden település a teljes felületével szerepeljen a kivágatban. Így született meg a *8. ábrán* lévő lehatárolás. A teljes kivágat a weboldal nézetében a *3. sz. mellékleten* látható.



8. ábra: A kivágat meghatározása északra tájolt állásban

A térkép vetülete

A webtérkép a 3857-es EPSG számú **Web Mercator** (Google Web Mercator, WGS 84/Pseudo-Mercator) vetületben készült, ami a térképes webalkalmazások egyik leggyakrabban használt vetülete (pl.: OpenStreetMap, Google Maps, MapQuest). A Web Mercator rendkívül jól használható az interaktív, tetszőlegesen nagyítható világtérképekhez, mert ugyan kis méretarányokban nagyfokú torzulások jelentkeznek, de belenagyítva egy kisebb területeken relatíve kicsik lesznek a torzulások annak köszönhetően, hogy ez a Mercator-változat is közel szögtartó.

A Web Mercator vetületnél egy, a WGS84 ellipszoid fél nagytengelyével egyenlő sugarú gömbről vetítünk a (szögtartó) gömbi Mercator-vetület segítségével a síkba. Ez a művelet pedig végeredményben nem szögtartó, hiszen az első lépésben (ellipszoid-gömb leképezés) a WGS84-es ellipszoidi koordinátákból a gömbi koordinátákat átalakítás nélkül kapjuk és az ellipszoid lapultsága miatt ez a leképezés nem szögtartó. A magasabb szélességeken emiatt szögtorzulás lép fel. (Lapaine-Usery)

A térkép tájolása

A korábbi ábra jól mutatja, hogy az ábrázolt terület hosszanti főtengelye délkelet-északnyugati irányban halad. Annak érdekében, hogy a térkép keretvonalait párhuzamossá tegyem a képernyő keretével, 52°-os elforgatást alkalmaztam. Így a térkép tájolása megközelítőleg északnyugati.

5.2.2 A háttértérképi rétegek

Ahogy az a korábbi fejezetekben már említésre került, a webtérkép háttereként szolgáló térképi rétegek WMS-en keresztül raszteres állományként érkeznek az OpenLayers-hez. Az OpenLayers az effajta rétegeket csak megjeleníteni képes, módosítani vagy a stílusát szerkeszteni nem. Viszont tekintve, hogy ezekhez a rétegekhez interaktív funkció nem kapcsolódik, ez alapvetően szükségtelen.

OpenLayers-ben a GeoServer által szolgáltatott WMS rétegek beágyazása a térképbe a következőképpen lehetséges.

Ezután a rétegek azzal a jelkulccsal fognak megjelenni a térképen, amit korábban GeoServerben SLD-vel definiáltunk.

A fenti kódrészletből fontos kiemelni a format paramétert. A GeoServer lehetőséget biztosít arra, hogy megválasszuk a kimenő raszteres kép formátumát. Sokszor azonban nehéz eldönteni, hogy melyik formátum a legideálisabb különböző tulajdonságaik miatt. A JPEG formátum hatékony tömörítést biztosít csökkentve ezzel a kimenő állomány méretét, ugyanakkor az átlátszóságot, mint pixeltulajdonságot nem ismeri. A PNG formátum már képes értelmezni az átlátszóságot is, de tömörítése kevésbé hatékony a JPEG-hez képest. Ha szükségünk van az átlátszóság kezelésére, de a képek méretét is optimalizálni szeretnénk, jó megoldás a 'image/vnd.jpeg-png' formátumtípus választása. Ez a formátum képes JPEG és PNG formátumban is előállítani a térképmozaikokat, annak függvényében, hogy adott mozaik milyen képi tartalommal rendelkezik. Amennyiben a mozaik teljes mértékben átlátszatlan (vagyis az átlátszóság figyelembevételére nincs szükség), akkor a kisebb méretű JPEG formátumban kerül előállításra. Ha azonban a mozaik átlátszó pixeleket is tartalmaz, PNG képet generál a GeoServer. Ezzel a beállítással csak azok a képrészletek használják a nagyobb méretű PNG formátumot, amelyek esetén az átlátszóság miatt ez valóban indokolt, minden más esetben a hatékonyabb tömörítésű JPEG használatos. Ez az eljárás a **Smart Transparency** (okos átlátszóság) módszere.

Egyéb raszteres hétterek

A webtérkép lehetőséget nyújt az adatbázis tartalmából generált háttértérképi rétegen kívül más raszteres rétegek megjelenítésére is. A rétegválasztó menüre kattintva a felhasználó kérésére megjeleníthető az 1 : 20 000-es méretarányú infra ortofotó vagy a fekete-fehér műszaki vízügyi térkép, amely térképeket szintén a GeoServer közvetít WMS-en keresztül.

5.2.3 A tematikus rétegek

A térkép vízügyi, idegenforgalmi, horgászati tematikával bíró rétegei döntően interaktív funkciókkal ellátva jelennek meg a webalkalmazásban. Az interaktivitás biztosítja, hogy ezen rétegek láthatósága a felhasználó igényei szerint ki- és bekapcsolható legyen. Ezen felül lehetőség nyílik a megjelenített objektumok attribútumainak lekérdezésére is. Ha ilyen funkciókkal kívánunk felruházni egy réteget, célszerű WFS szolgáltatáson keresztül lekérni azt az adatbázisból, így minden adat és attribútum a rendelkezésünkre áll.

WFS réteget a következő módon adhatunk hozzá OpenLayers térképünkhöz.

```
. . .
           /*A WFS réteg forrása*/
var benzinkutSource = new ol.source.Vector({
    format: new ol.format.GeoJSON(),
    url: function(extent) {
      return 'http://localhost:8080/geoserver/wfs?service=WFS&' +
          'version=1.0.0&request=GetFeature&' +
          'typename=tiszato:toltoallomasok&' +
          'outputFormat=application/json&srsname=EPSG:3857&' +
          'bbox=' + extent.join(',') + ',EPSG:3857';
    },
    strategy: ol.loadingstrategy.bbox
      });
           /* Réteg deklarálása */
var benzinkut = new ol.layer.Vector({
     source: benzinkutSource,
     visible: false,
    maxResolution: 8,
     style: new ol.style.Style({
           image: new ol.style.Icon({
              src: 'png/benzinkut.png',
               scale: 0.33
           })
     })
});
. . .
```

Ez a kódrészlet a *benzinkutak* réteg példáján keresztül mutatja be a WFS rétegek deklarálásának módját, de hasonlóképpen történik minden más pontszerű tematikus objektum megjelenítése is.

A tematikus rétegek egyedi ikonjai

A fenti forráskódban az is látható, hogy a pontszerű elemek stílusbeállítása külső forrásból származó raszteres képet használ a térképi ábrázoláshoz.

A pont típusú tematikus objektumok megjelenítésére önálló ikonokat készítettem Corel Draw program segítségével (9. *ábra*). Az ikonok tervezésekor sok esetben az alapötletet a *www.flaticon.com* ingyenes ikongyűjtemény szolgáltatta.



9. ábra: Egyedileg tervezett ikonok a teljesség igénye nélkül

Az attribútumok megjelenítése

Az tematikus rétegek interaktivitását tovább gazdagítja a térképi elemekhez tartozó attribútumok lekérdezésének lehetősége. Az attribútumadatok a térképi elem ikonjára kattintva jelennek meg egy felugró ablakban. Az ablak tartalma annak függvényében változik, hogy az adott elemhez milyen megjeleníthető adatok tartoznak. Az alábbi kódrészletek a popup ablak létrehozását szemléltetik adott típusú térképi elemhez.

A popup ablak létrehozása (stílusbeállítások külön CSS fájlban):

Alább látható a függvény meghatározása, amely az ikonra kattintva létrehozza a felugró ablakot, benne a megfelelő tartalommal. A kódban a *feature4* változó azt az elemet deklarálja, amely a kattintás helyén a *szallashelyek* rétegre esik. Ezután a popup tartalmát (*content*) feltölti az elem megfelelő adataival. Ha léteznének más rétegek is, amelyeknek *popup-content*-je a szálláshelyekével megegyezően kerülne kiíratásra, akkor azok elemei is a *feature4* változó alatt lennének deklarálva.

```
map.on('click', function(evt){
    ...
var feature4 = map.forEachFeatureAtPixel(evt.pixel, function
      (feature, layer) {
        if (layer == szallashelyek){
            return feature;
            }
        });
    ...
```

```
if (feature4) {
  var geometry = feature4.getGeometry();
  var coord = geometry.getCoordinates();

  var content = '<h3>' + feature4.get('nev') + '</h3>';
    content += 'szálláshely';
    content += '<h4><u>Elérhetőségek</u></h4>';
    content += 'ob>Cím:</b> ' + feature4.get('cim') + '';
    content += 'ob>Telefon:</b> ' + feature4.get('telefon') + '';
    content += 'ob>E-mail:</b> ' + feature4.get('email') + '';
    content += 'ob>E-mail:</b> ' + feature4.get('email') + '';
    content += 'ob>E-mail:</b> ' + feature4.get('email') + '';
    content_element.innerHTML = content;
    overlay.setPosition(coord);
    console.info(feature.getProperties());
    }
});
....
```

Értelemszerűen annyiféle *feature* típus került meghatározásra (*feature1*, *feature2* stb.) ahány különbözőféle *popup-content* létezik.

A tematikus rétegek és a hozzájuk kapcsolódó információk térképi megjelenítését a 4. sz. mellékleten látható példa szemlélteti.

A rétegválasztó ablak

Az egyes rétegek ki-, és bekapcsolhatóságának lehetősége a felhasználó számára biztosított. A rétegek láthatóságának beállítása a *Rétegek* menüelemre kattintva lehetséges. Kattintásra egy ablak jelenik meg, ahol rádiógombos megoldással lehet a rétegeket megjeleníteni vagy elrejteni. Ugyanitt lehet a raszteres hátterek közül választani (alap WMS, ortofotó, műszaki vízügyi térkép). A rádiógombokkal a következőképpen befolyásolható a térképi tartalom (következő példán: tömegközlekedés réteg ki/be kapcsolása).

A fenti rádiógombok minden vektoros és raszteres rétegre beállításra kerültek, amik a következő elrendezésben helyezkednek el a rétegválasztó panelen (*10. ábra*).



10. ábra: Rétegválasztó ablak

6. Továbbfejlesztési lehetőségek

Az elkészült térinformatikai rendszer számos olyan lehetőséget rejt még magában, amelyek kidolgozásával az adatbázis tovább pontosítható, a webtérkép funkciói pedig tovább bővíthetők.

Az adatbázis tematikus adatai jórészt internetes adatgyűjtés útján vagy külső forrásból származó anyagok alapján lettek kialakítva. A térképekkel szemben alapvető elvárás, hogy pontos legyen, és friss adatok legyenek rajta, ez pedig sokszor csak terepi helyszíneléssel lehetséges. Figyelembe véve azonban az ábrázolt terület nagyságát és a pontosítani kívánt adatok mennyiségét, látható, hogy ez a lépés ennek a munkának a keretein belül nem valósulhatott meg rendkívüli idő-, és munkaerő-igényessége miatt.

A webtérkép grafikai megjelenítéséhez használt eszközök (OpenLayers, SLD) kelléktára ugyan folyamatosan bővülni látszik, ennek ellenére hiányosságaik ritkán még mindig megmutatkoznak. Esetemben ez a névrajz kialakításánál okozott gondot. A felületi nevek (területnevek) ábrázolása ugyanis nem, vagy csak igen pontatlanul lehetséges olyan formai jellemzők hiányában, mint a betűszórás. További nehézségeket jelent, ha a név ívelten kerül megírásra. A probléma jelentőségét mutatja, hogy a legnépszerűbb webtérképes alkalmazásokban (Google Maps, OpenStreetMap, Bing Map) sem találkozunk ehhez hasonló névrajzzal.

7. Összegzés

Diplomamunkámban egy olyan térinformatikai rendszer tervezését és kivitelezését valósítottam meg, amelynek kliens oldali végterméke egy webtérkép, amely a Tisza-tavat és környékét mutatja be turisztikai, vízügyi, és horgászati tematikával. Dolgozatom ugyanakkor nem pusztán a térkép kialakításának módját tárgyalja, de különös hangsúlyt fektet azokra a szerveroldali munkafolyamatokra is, amelyek egy ehhez hasonló webalkalmazás kialakításá- nak hátterében zajlanak. Mivel munkám eredménye egy teljesen önálló rendszer, a dolgozat a munka kereteit szolgáló szoftveres infrastruktúra kialakításától kezdve vezet végig a rend-szerépítés folyamatán egészen a kliensoldali megjelenítésig. Ennek a munkafolyamatnak a lépéseit taglalják a diplomamunka egyes fejezetei.

A dolgozat első részében részletesen ismertetem annak a fejlesztői környezetnek a kialakítását, amely az egész rendszer működésének alapját képezi. Ennek a környezetnek az egyik kulcsfontosságú egysége a diplomamunka címében is megfogalmazott adatbázis. A továbbiakban ennek létrehozását, a benne tárolt adatok forrását és kialakításuk módját ismertetem. Ezt követően mutatom be az adatbázis szerveren keresztül történő publikálásának lehetőségét a GeoServer segítségével. Részletesen kitérek a WMS szolgáltatás sajátosságaira, és a WMS stílusának kialakítási módjára. Végül a GeoServer által szolgáltatott adatokra építettem egy mindenki által böngészhető interaktív webtérképet az adatbázis adatainak grafikus szem-léltetésére.

Munkámmal célom, hogy egy átfogó képet nyújtsak az adatbázis-alapú térképes webalkalmazások építésének (egy lehetséges) módszeréről.

A dolgozat eredményeként létrejött weboldal a http://tisza.elte.hu címen érhető el.

8. Irodalomjegyzék

- HUPWIKI (2009): *PostGIS*, https://wiki.hup.hu/index.php/PostGIS. Utolsó elérés: 2019. 05. 10.
- GeoServer Felhasználói Kézikönyv (ver. 1.1): Fordította: dr. Bata Teodóra, Dolleschall János. http://docplayer.hu/1709222-Geoserver-felhasznaloi-kezikonyv.html. Utolsó elérés: 2019. 05. 10.
- Miljenko LAPAINE, E. Lynn USERY: *Térképvetületek és koordinátarendszerek*, fordította és átdolgozta: Gede Mátyás. https://icaci.org/files/documents/wom/09_IMY_WoM_hu.pdf Utolsó elérés: 2019. 05. 10.
- Michael MONCUR (2002): *Tanuljuk meg a JavaScript használatát 24 óra alatt*, Kiskapu Kiadó, Budapest, p. 13.
- PACZONA Zoltán (2001): *HTML technikák a gyakorlatban*, Camputer Panoráma Kiadó, Budapest, p. 40.
- NGUYEN Thai Binh (2012): *GDAL, PostgreSQL- PostGIS*, Eötvös Loránd Tudományegyetem, Informatikai Kar. http://tarsadalominformatika.elte.hu/tananyagok/gdal/index.html. Utolsó elérés: 2019. 05. 10.
- NYÍRI Imre (2003): A PostgreSQL adatbázis-kezelő használata. Linuxvilág, 2003/1, p. 50
- OSGEOLIVE: *PostGIS Overview*, https://live.osgeo.org/hu/overview/postgis_overview.html. Utolsó elérés: 2019. 05. 10.
- ÓNODI Zsolt (2017): *QGIS WFS réteg használata*, AdatTérKép. https://adatterkep.com/qgiswfs-reteg-hasznalata. Utolsó elérés: 2019. 05. 10.
- PŐDÖR Andrea (2010): *Kartográfia* + *Webmapping* 7., Nyugat-magyarországi Egyetem, Geoinformatikai Kar, Székesfehérvár
- SIKI Zoltán (2010): WMS kliens használata (1.6 vezió), Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest. http://www.agt.bme.hu/gis/qgis/qgis_wms.pdf. Utolsó elérés: 2019. 05. 10.

- TARCSI Ádám, HORVÁTH Győző (2012): Webadatbázis-programozás, Eötvös Loránd Tudományegyetem, Informatikai Kar https://www.tankonyvtar.hu/hu/tartalom/tamop412A/ /2010-0011_web_adatbazisprog/index.html. Utolsó elérés: 2019. 05. 10.
- Ubuntu dokumentáció: *Apache Tomcat*. http://sugo.ubuntu.hu/10.10/html/serverguide/ /hu/tomcat.html. Utolsó elérés: 2019. 05. 10.

9. Ábrajegyzék

A jegyzékben feltüntetett ábrák minden esetben saját képek és képernyőmentések.

1. ábra: A rendszer architektúra diagramja	9
2. ábra: Shapfile-ok importálása az adatbázisba a shp2pgsql segítségével	. 10
3. ábra: Ugyanazon földrészlet légifotón és digitalizálva saját jelkulccsal	. 11
4. ábra: Nádas kitöltőminta a 1:0000-nél nagyobb méretarányokban	. 16
5. ábra: A nádas növényzeti kategória ábrázolása különböző méretarányokban	. 17
6. ábra: A főutak útkategória ábrázolása különböző méretarányokban	. 18
7. ábra: A weboldalt felépítő egységek	. 21
8. ábra: A kivágat meghatározása északra tájolt állásban	. 22
9. ábra: Egyedileg tervezett ikonok a teljesség igénye nélkül	. 25
10. ábra: Rétegválasztó ablak	. 28

10. Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Elek Istvánnak, a téma megválasztásában nyújtott segítségéért, továbbá mindenkori támogatásáért, ami nélkül a dolgozat nem jöhetett volna létre.

Köszönettel tartozom továbbá Kovács Bélának, aki munkám internetes publikálásában nyújtott segítséget.



1. sz. melléklet: A növényzeti fedettség teljes jelkulcsa a méretarány függvényében

11. Mellékletek

-						
	1 : 5 000 - - 1 : 10 000	1 : 10 000 - - 1 : 20 000	1 : 20 000 - - 1 : 40 000	1 : 40 000 - - 1 : 80 000	1 : 80 000 - - 1 : 160 000	1 : 160 000 - - 1 : 320 000
Főút						
Másodrendű út						
Egyéb burkolt út						
EuroVelo 11						
Dózerút						
Szekérút						
Ösvény						
Gyalogút						

2. sz. melléklet: Az úthálózat teljes jelkulcsa a méretarány függvényében







4. sz. melléklet: Tematikus réteg (szálláshelyek) megjelenítése. attribútumokkal

12. Nyilatkozat

DIPLOMAMUNKA LEADÁSI és EREDETISÉG NYILATKOZAT

Alulírott Eigner Péter, Neptun-kód: JR2ZI3 az Eötvös Loránd Tudományegyetem Informatikai Karának, Térképtudományi és Geoinformatikai Tanszékén A Tisza-tó idegenforgalmi célú térinformatikai adatbázisa című diplomamunkámat a mai napon leadtam.

Témavezetőm neve: dr. Elek István

CD-t / DVD-t mellékelek (aláhúzandó): <u>igen</u> nem

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

Tudomásul veszem, hogy szakdolgozat/diplomamunka esetén plágiumnak számít:

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

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

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2019. május 15.

hallgató aláirása