

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR
TÉRKÉPTUDOMÁNYI ÉS GEOINFORMATIKAI TANSZÉK

DIPLOMAMUNKA TÉMABEJELENTŐ

Hallgató adatai:

Név: Tóth András

Neptun kód: C4B5XJ

Képzési adatok:

Szak: térképész, mesterképzés (MA/MSc)

Tagozat: Nappali

Belső témavezetővel rendelkezem

Témavezető neve: Dr. Ungvári Zsuzsanna

munkahelyének neve: ELTE IK Térképtudományi és Geoinformatikai Tanszék

munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/A.

beosztás és iskolai végzettsége: adjunktus

A diplomamunka címe: Webes összehasonlító névrajzi kereső kialakítása térképekhez

A diplomamunka témája:

(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben diplomamunka témájának leírását)

A diplomamunka elkészítése során fő cél egy olyan webes kereső és megjelenítő weboldal létrehozása, amelyen különböző térképek egyes névrajzi elemek csoportjait lehet megjeleníteni, és vizsgálni az egyes nevek időbeli változását. A munka során szükséges lesz a weboldalhoz kapcsolódó adatbázis létrehozására. Az adatbázis létrehozása során a kézi elemfelvétel mellett megvizsgálom az automatikus karakterfelismerés lehetőségét is.

Budapest, 2021. 11. 30.

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR

Webes összehasonlító névrajzi kereső kialakítása térképekhez

DIPLOMAMUNKA
TÉRKÉPÉSZ MESTERSZAK

Készítette:
Tóth András

Témavezető:
Dr. Ungvári Zsuzsanna
adjunktus
ELTE Térképtudományi és Geoinformatikai Intézet



Budapest, 2022

Tartalomjegyzék

1. Bevezetés.....	4
2. Előzmények.....	5
3. Alapanyagokról.....	6
3.1. Letöltött településállomány	7
3.2. 1906-os térkép	7
3.3. 1911-es térkép	7
3.4. 1927-es térkép	7
3.5. 1948-as térkép	8
3.6. 1961-es térkép	8
3.7. 2004-es térkép	8
4. Településnevekről	10
4.1. Településnevekről általánosságban	10
4.2. Törzskönyvezésről	10
4.3. A felhasznált településnevek típusai	11
5. OCR karakterfelismerés	12
6. Adatnyerés végső folyamata	14
6.1. Adatfelvétel	14
6.2 Az adatbáziskezelő	16
7. A Weboldal felépítése	19
7.1. A weboldal tartalma	19
7.2. Keresés menete.....	19
8. A weboldal fejlesztések.....	22
8.1. A felhasznált programozási nyelvek	22
8.1.1. HTML.....	22
8.1.2. PHP.....	22
8.1.3. JavaScript.	22
8.1.4. AJAX.....	23
8.2. A településnév szerinti keresés létrehozása	23
8.2.1. A településnév szerinti keresés létrehozása – HTML kód	23
8.2.2. A településnév szerinti keresés létrehozása – JS kód.....	24

8.2.3. A településnév szerinti keresés létrehozása – PHP kód	26
8.3. A terület szerinti keresés létrehozása – HTML kód	27
8.3.1. A terület szerinti keresés létrehozása – HTML kód	27
8.3.2. A terület szerinti keresés létrehozása – JS kód.....	28
8.4. A térkép létrehozása	30
8.4.1. A térkép létrehozása – HTML	30
8.4.2. A térkép létrehozása – JS	30
8.5. A weboldal végleges elérhetősége	36
9. Összegzés	37
Köszönetnyilvánítás	38
Irodalomjegyzék.....	39
Mellékletek.....	40

1. Bevezetés

A térkép Föld vagy más égitest felszínének, vagy a felszínre vonatkoztatott természeti és társadalmi típusú tárgyaknak és jelenségeknek meghatározott matematikai szabályok vagy mértani törvények szerint síkba vetített, méretarányosan kisebbített, általánosított, és sajátos grafikai jelrendszerrel bemutatott felülnézeti ábrázolása. [1] A térképek jelentős szerepet töltenek be az emberek tájékozódásában már az ókor óta. A geoid alakú Föld síkba fejtése mindig is problémát okozott, ezért különböző vetületek segítségével igyekeznek a szakemberek síkba fejteni a geoidot úgy, hogy a lehető legkisebb torzulással mutassa be környezetünket.

A térkép azonban nem csak egy navigációs vagy tájékoztató céllal készített földfelszín bemutató alkotás, hanem egyben egy lenyomat is a korról, amiben készült. Egy régi térképet kezünkbe véve meglepődhetünk, hogy akár rövid idő alatt mennyi változáson ment keresztül egy-egy település neve. Ez utalhat a nyelv helyesírásában bekövetkezett változásokra, ideológiai változásokra, de történelmi eseményekre is, például határmódosításokra. Magyarország elmúlt 120 évében számtalan változás következett be, amik leképződtek a településneveken is. Ez egy érdekes kutatási terület és érdekelt ezeknek a változásoknak a nyomon követése. Ennek céljából döntöttem egy webes névrajzi összehasonlító weboldal összeállítás mellett, amin nyomon követhetővé válik az elmúlt évszázadban bekövetkező változások a névrajzi anyagon. Céлом a kereső kialakításakor nem csak egy adatbázis létrehozása volt, hanem egy olyan interaktív felület kialakítása is, ami alapul szolgál a névváltozások áttekintéséhez, nyomon követéséhez. Ezen a felületen látható a XX. század eleji törzskönyvezés, a településösszevonások, Trianonban elcsatolt területek és a kommunista diktatúra ideológiai alapú névadásai.

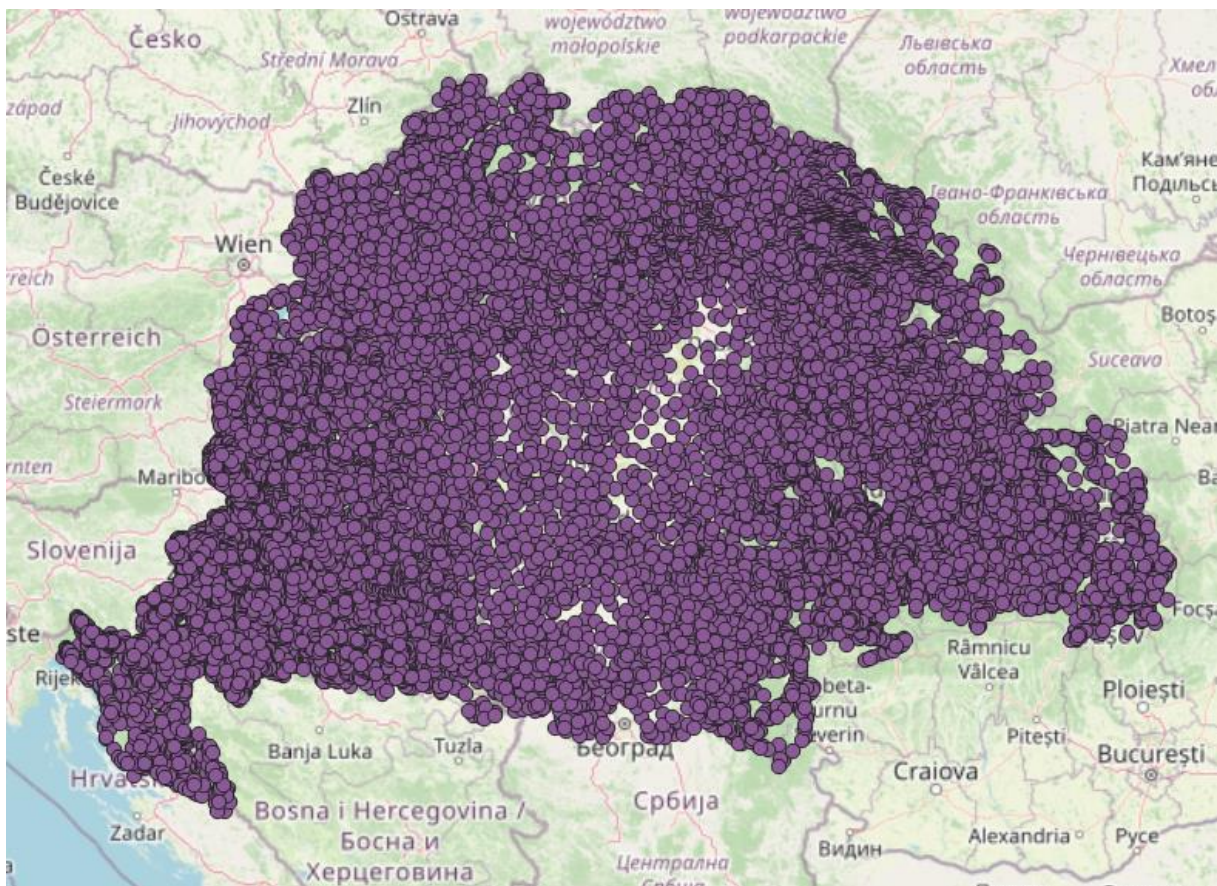
2. Előzmények

Korábbi években Beszkid Arthur: Webes összehasonlító névrajzi kereső kialakítása glóbuszokhoz és Kuris Zoltán: Webes összehasonlító névrajzi kereső fejlesztése glóbuszokhoz című szakdolgozataikban foglalkoztak hasonló témákkal. Különbség, hogy az ő esetükben glóbuszokhoz alakítottak ki keresőt, míg én térképekhez, de a munkájukat fel tudtam használni kiindulási pontnak. Az OCR alkalmazásához Vassányi Gergely: Pontszerű jelek automatikus felismerése archív térképeken konvolúciós neurális hálózat használatával és Patkó Máté: Interaktív térképes földrajzi névmutató készítése című diplomamunkáik láttak el alapszolgáltatásokkal.[2][3][4][5]

3. Alapanyagokról

3.1. Letöltött településállomány

A munka elvégzéséhez szükség volt a magyar korona országainak településhálójáról egy alap adatbázisra. Mivel ennek az adatbázisnak az előállítása rendkívül időigényes lett volna, ezért ingyenes forrásból történő letöltésre esett a választás. A geofabrik.de oldalon minden ország településhálózata elérhető shapefile formátumban, így Magyarország, Szlovákia, Ukrajna, Románia, Szerbia, Horvátország, Szlovénia és Ausztria shapefile-jait töltöttem le. Ezek a fájlok nem csak településeket tartalmaztak, hanem egyéb külterületi neveket is, amiket ki kellett törölni. További probléma volt, hogy ezek a teljes államterületekre vonatkoztak, nem csak az egykor Magyarországhoz tartozott településekre, ezért ezeket a területeket le kellett vágni. Ehhez a QGIS-be behívott az egykori magyar területeken kívül eső településeket töröltem. Miután az összes ország shapefile-ján megtörtént a vágás, következhetett a shapefile-ok összevonása, a művelet végén pedig elkészült az alapadatbázis, amire épült az egész projekt (1. ábra). [12]



1. ábra A letöltött és leválogatott települések QGIS programban

3.2. 1906-os térkép

Az első térkép, amit feldolgoztam, az 1906-os Nagy Magyar Atlaszban jelent meg, ami a Magyar Földrajzi Társaság megbízásából készült. Az atlasz szerkesztője Dr. Brózik Károly főreáliskolai tanár volt, valamint közreműködött Erődi Béla a Magyar Földrajzi Társaság volt elnöke és Berecz Antal a Magyar Földrajzi Társaság tiszteletbeli főtitkára is. A könyv kiadója Lampel R. Könyvkereskedése (Wodianer F. és Fiai) Részvénytársaság volt. A térkép az atlasz 58. és 59. oldalán található Ausztria és Magyarország című Osztrák-Magyar Monarchiát ábrázoló térkép, amiből én a magyar korona alá tartozó területeket dolgoztam fel. A méretarány 1 : 3 750 000. A térkép érdekessége, hogy az 1898 és 1912 közt az Országos Községi Törzskönyvbizottság helységnévrendezése, amely során az ország valamennyi településnevét megvizsgálták és 2693 név egybeírásáról, illetve 227 név esetén történt egyéb helyesírást érintő változtatásról döntöttek. Ekkor még a törzskönyvezés eredményeit nem használták fel az atlasz névanyagához, így a térképen is a településnevek nem a mai helyesírás szabályai szerint szerepelnek. Például Zala-Egerszeg, Marczali vagy Jankovác (mai Jánoshalma). [13]

3.3. 1911-es térkép

Időben a következő felhasznált térkép Kogutowicz Teljes Földrajzi és Történelmi Atlaszában lelhető fel, melyet 1911-ben a Magyar Földrajzi Intézet Részvény Társaság adott ki. Ennek az atlasznak ez volt az ötödik bővített kiadása és a címben szereplő Kogutowicz Manó szerkesztésének köszönhetően jött létre.

Ebből az atlaszból a Magyarország politikai térképe című térképet választottam, mivel ez ábrázolja egyben a magyar koronához tartozó területeket. A térkép 1 : 2 400 000 méretarányban készült. A névanyagról elmondható, hogy a törzskönyvezés eredménye már lenyomatott hagyott rajta, annak ellenére, hogy még folyamatban volt. A korábbi térképen még külön írt településnevek itt már egybeírva szerepelnek, illetve a Magyarországhoz tartozó települések esetében megtörtént a településnevek magyarosítása is pl.: Jankováczból Jánoshalma lett. A cz-vel írt nevek írása azonban még ezen a térképen is megmarad például Czegléd esetében is.[14]

3.4. 1927-es térkép

A harmadik felhasznált térkép Cholnoky: Földrajzi és Statisztikai Atlaszban található meg, mely Győző Andor kiadásában jelent meg 1927-ben és Cholnoky Jenő szerkesztette.

Ebben az atlaszban a Magyarország című térképet választottam, ami a 15. oldalon található, mivel ez a térkép egyben ábrázolja a Kárpát-medencét. A méretaránya 1 : 5 000 000. A térkép érdekessége, hogy a felhasznált térképek közül ez az első, amely Trianon után készült. A készítő a térképet olyan ábrázolási módszerrel alkotta meg, hogy tisztán kivehetőek az 1920 előtti határok. A névrajzi anyagáról elmondható, hogy a helyesírása megfelel a mai normáknak, illetve a térkép egynyelvű, az államnyelvi névalakok nem találhatóak meg rajta. [15]

3.5. 1948-as térkép

A negyedik felhasznált térkép az 1948-as Bibliotheca Világatlasz Dr. Kéz Andor egyetemi tanár és Dr. Takács József térképész által szerkesztett atlaszban található meg. Ebben az atlaszban nem található a Kárpát-medencét egyben bemutató térkép, mivel a második világháború elvesztése után kialakuló új politikai rendszer politikailag inkorrektnek tekintette a Kárpát-medence egy térképlapon történő ábrázolását, mivel a történelmi Magyarország határai nagyrészt egybe estek a Kárpát-medence határaival. Az atlasz készítője ezt a problémát úgy küszöbölte ki, hogy Magyarországról és minden környező országról található benne térkép, ezért kutatásom során ezeket használtam fel. Az atlaszban található Magyarország térkép 1 : 2 500 000 méretarányú. [16]

3.6. 1961-es térkép

Az ötödik térkép A Politikai és Gazdasági Világatlaszban található meg, ami a Kartográfiai Vállalat 1961-es kiadásában jelent meg. Ebben az atlaszban szintén nem jelent meg a Kárpát-medencét egyben ábrázoló térkép, tekintve hogy ekkoriban is hasonló ideológiai környezet állt fent, mint az 1948-as atlasz születésekor. A helyzet annyit változott, hogy ebben az atlaszban már nem is jelent meg olyan térkép, amin a szomszédos országokat önmagában ábrázolná a térképszerkesztő, hanem a szomszédos országok más országokkal együtt szerepelnek, például Románia Bulgáriával szerepel egy térképlapon. [17]

3.7. 2004-es térkép

Az utolsó térkép, amivel foglalkoztam a Világatlasz országlexikonnal és tematikus fejezetekkel volt, amit a TOPOGRÁF Térképészeti Kft. készített és a NYÍR-KARTA Bt. kiadásában jelent meg 2004-ben. Az 1961-es és 2004-es térkép közt 43 év telt el, ennek ellenére nem választottam más térképet a köztes időszakból, mivel a szocializmus utolsó évtizedeiben kevés változás történt a névanyagban, így nem láttam értelmét. Inkább településösszevonások voltak jellemzőek, de helyesírásban nem történt változás, ideológiai

helységnévadás is csak Tiszaszederkény Leninvárossá átnevezése esetében történt. Így, hogy kimaradt 43 év több változásnak volt ideje történni, ilyen változás a rendszerváltás utáni korábban összevont települések különválása. 2004 után inkább csak közigazgatási változások történtek (várossá nyilvánítás, megye vagy járásváltások).[18][23]

4. Településnevekről

4.1. Településnevekről általánosságban

Munkám során az 1914 előtti Magyar és Horvát Királyság területének településnév anyagát gyűjtöttem össze 5 térkép alapján. Településnév alatt értjük azokat a társadalom által megalkotott objektumok neveit, amelyeket az ember lakó-, munka és pihenőhelyül szolgálnak. Alapvetően az adatbázis építése közben a térképen fellelhető magyar nyelvű névalakot (ha volt), illetve a térképen fellelhető államnyelvi alakot, ami minden esetben latin betűs volt a mai cirill ábécét használó országok közül, és a mai államnyelvi alakot gyűjtöttem össze, hogy összeköthető legyen az alapadatbázissal. Az államnyelvi alak az a névalak, ami minden ország közigazgatásában hivatalosan megállapított és dokumentált nevek, amelyek második és harmadik névalakként nem államnyelvi név is lehet. Ezek jelentős része a terepen táblákon is jelezve van a települések belterületének kezdetén és végén, amit a közlekedés-rendszetben meghatározott formátáblák jeleznek. A településnevek a térkép méretarányától függően pontra vagy pontként viselkedő felületre vonatkozó nevek, melyek fogalmi csoportjai szorosan kötődnek hivatalosan megállapított településkategóriákhoz. Azért is választottam ezt a kategóriát, mert ez a csoport a legállandóbb a térképi megjelenítésben, így ebben az esetben válogathattam a legszélesebb körű térképválasztékból. [6]

4.2. Törzskönyvezés

A magyar hivatalos településnév-megállapítások időszaka 1898-tól 1912-ig tartott és négy nem magyar többségű (Árva, Fogaras, Hunyad és Liptó) vármegye kivételével az egész ország területén hivatalossá vált. Ahol a törzskönyvezés elvégzésre került, ott a törzskönyvbizottság által meghatározott névalakot kellett hivatalosan használni. A névmegállapításokat és azok nyilvántartását a Központi Statisztikai Hivatalhoz tartozó Országos Községi Törzskönyvbizottság végezte, mely ellenőrizte az egyes vármegyék minden településének nevének előforduló alakjait, helytörténeti és névtörténeti kutatásokat végzett a helyi előjárók és hozzáértők megkérdezésével. A kutatások után a Bizottság javaslatot tett a település új nevére, aminek a szentesítése a belügyminiszter feladata volt. A törzskönyvezés végére maradtak a legkevésbé magyarok lakta vármegyék települései. A többségében nem magyarok lakta vármegyék tiltakozása hatására a miniszterelnök 1912-ben hatályon kívül helyezte a törzskönyvezésről szóló 1898-as törvényt, de ahol végeztek a törzskönyvezéssel, ott hivatalos maradt annak eredménye. Csupán Árva, Liptó, Fogaras és Hunyad vármegyében maradt meg a korábbi állapot. A törzskönyvezettnek tekintett neveket az 1913.

évi helységnévtár tartalmazza. A törzskönyvezés nem vonatkozott a Horvátországhoz tartozó vármegyékre, mivel a településnévgondozás Horvátország saját hatáskörébe tartozott, melynek következtében a horvát kormányzat magyar helységnevek nagy részét szlávósította vagy jelzővel látta el. Ezzel Horvátországban gyakorlatilag megszűnt a magyar helységnév, melyek archaizálódtak.

Aváltoztatások csak az akkor önálló települések neveit érintették, tehát helységrészek, külterületi lakott helyek neveit nem változtatták. A törzskönyvezési folyamat vármegyéenkénti sorrendben történt. A Magyar Királyság 1920 előtti területén összesen 12 537 településből 4662 település neve változott meg. Változások főként kötőjelezés helyett egybeírást (Zala-Egerszeg → Zalaegerszeg), hangjelölési módosulást (*Szegzárd* → *Szekszárd*), jelzővel differenciálást (*Szent-György* → *Baranyaszentgyörgy*) és fordítást (*Tergovesty* → *Vásáros*) jelentett. [6]

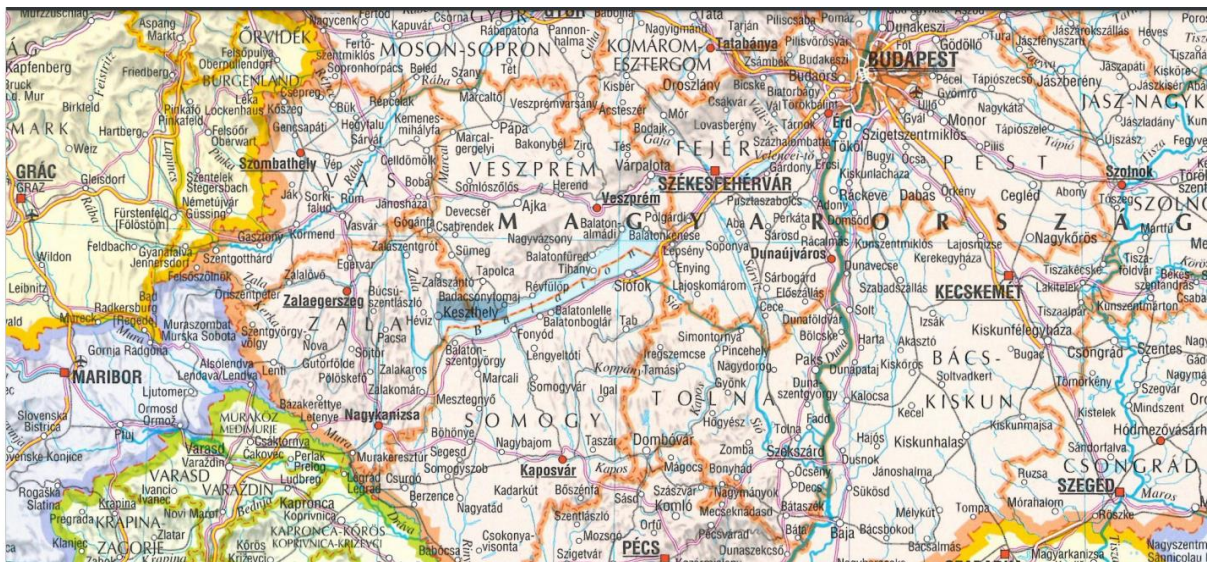
4.3. A felhasznált településnevek típusai

A települések a letöltött adatbázis szerint négy kategóriába sorolódtak: falu (*village*), kisváros (*town*), város (*city*) és főváros (*capital_city*). Ezekben a kategóriákban én sem változtattam. Az adatbázist átvizsgálva arra a megállapításra jutottam, hogy a falu (*village*) kategória meghatározása falvak esetében jogállás szerint történik, tehát azok a települések kerültek a falu kategóriájába, amelyek jogilag falvak. A kisváros (*town*) és város (*city*) elkülönítése lélekszám alapján történik. Az adatbázis vizsgálata alapján a határ 25 ezer fő, azok a települések, amelyek lélekszáma ez alatt a szám alatt van kisváros (*town*) kategóriába sorolódnak, amely települések több lakóval rendelkeznek, azok a város (*town*) kategóriába. Fővárosnak (*capital_city*) az adatbázisban három város minősül: Budapest, Pozsony és Zágráb. Az adatbázisban összesen 16 938 település található meg, de a településnevek változása ezek közül csak azok esetében lehetséges, amelyek megjelentek a térképeken.

5. OCR karakterfelismerés

Munkám során felmerült, hogy a munka felgyorsítása érdekében OCR technológiát is bevetek. Az OCR optikai szövegfelismerést jelent, amely technológia lehetővé teszi, hogy a képek, szkennelt vagy digitálisan fotózott dokumentumok, jelen esetben térképek szerkeszthető és szövegesen kereshető formába alakítását. Ennek előnye lett volna, hogy az így kapott anyag kivágható és Excelbe másolva jelentősen gyorsíthatta volna az adatnyerés folyamatát. Az OCR lényege, hogy a program elemzi a dokumentum kép szerkezetét. Az adott oldalt elemeire bontja és szöveg, kép, illetve táblázat blokkokat készít. A sorokat szavakra bontja, a szavakat karakterekre. Amikor már karakter szinten jár, akkor azokat egy belső mintával hasonlítja össze, és különféle hipotéziseket állít fel arra nézve, hogy mi lehet ez a karakter. A hipotézisek alapján megvizsgálja a különféle szó és betűalkotási lehetőségeket az adott sorban. Miután nagyszámú lehetséges hipotézist lefuttatott, eldönti, hogy mi lesz a végleges szöveg.

Annak érdekében, hogy kipróbáljam ezt a technológiát, szétnéztem az interneten, hogy milyen OCR technológiák érhetőek el. Sajnos ennek a technológiának a nagy része fizetős én viszont mindenképp ingyen elérhető programmal szerettem volna dolgozni. Végül az online elérhető tools.pdf24.org oldalra esett a választásom. [19] Ennek az oldalnak nagy előnye, hogy online interneten keresztül futtatható a karakterfelismerés böngészőn keresztül. Az oldalon lehetséges nem csak pdf fájlformátumú dokumentumokat is feltölteni, valamint nyelvi beállításoknál a magyar nyelv is kiválasztható, ami szintén komoly előny, mivel számos program estén nem elérhető a magyar nyelv, ami lehetetlenné teszi, a településnevek pontos felismerését. A 2004-es térkép feltöltése után futtattam a karakterfelismerést (2.ábra), ami kiábrándító eredményt hozott (3.ábra).



2. ábra Településnév kijelölése a tools.pdf24.org oldalon az OCR karakterfelismerés futtatása után

A1						
	A	B	C	D	E	F
1	gkeséttele					
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						

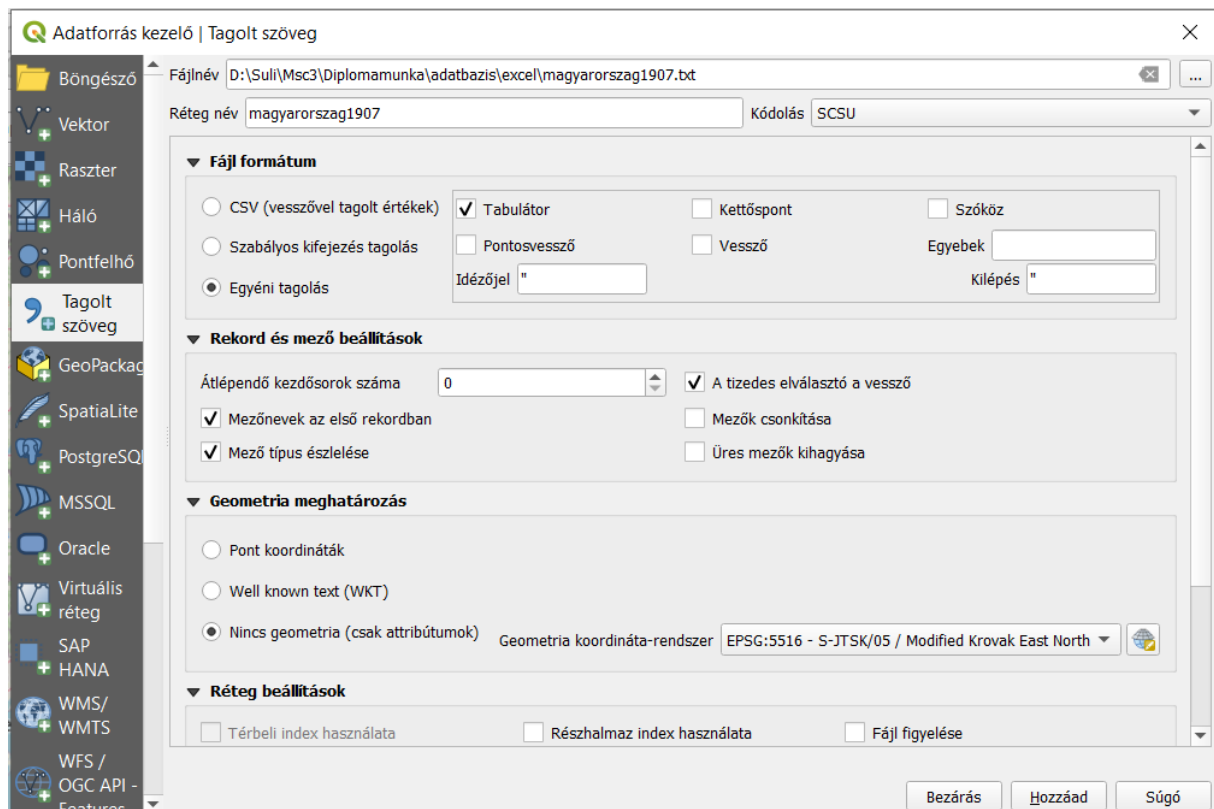
3. ábra A karakterfelismerés után kimásolt Keszthely településnév beillesztés után

A program bár szerkeszthetővé és kivághatóvá konvertálta a térképen található szöveget, amit egyszerű kijelöléssel és másolással be tudtam illeszteni az Excelbe, de a program által létrehozott szövegnek közes sem volt az eredeti térképre írt tartalomhoz, így ejtettem ezt az adatnyerési módot.

6. Adatnyerés végső folyamata

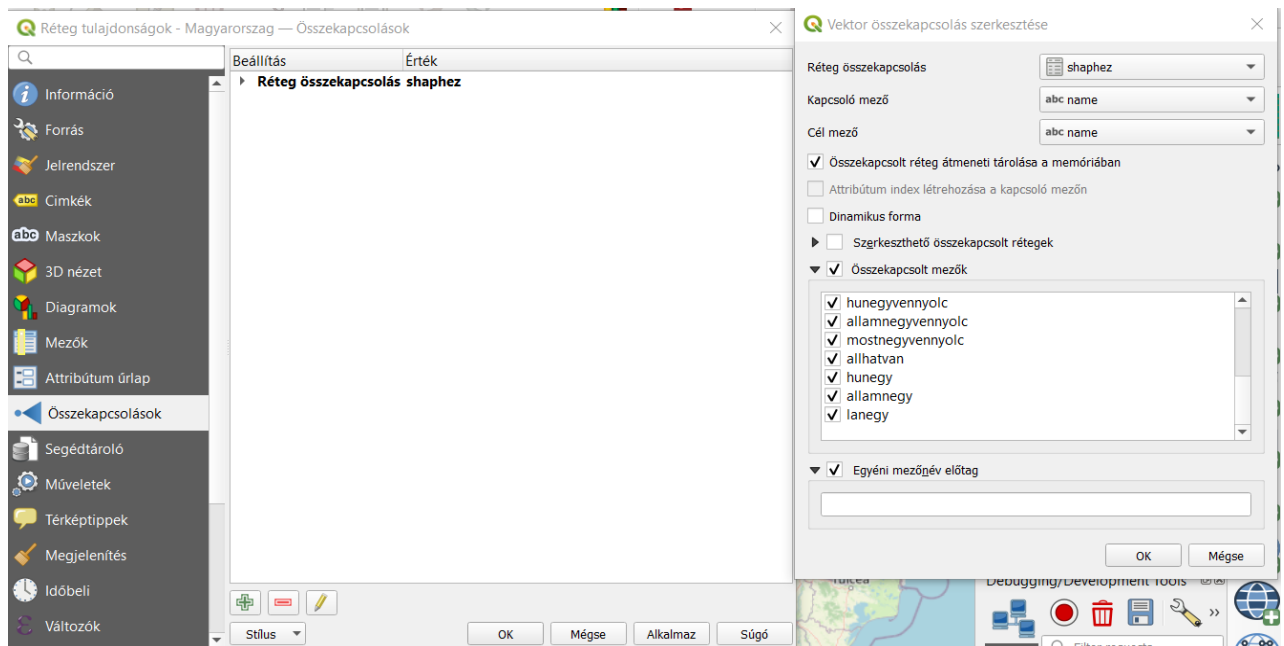
6.1. Adatfelvétel

Miután kiderült, hogy az OCR technológia alkalmatlan a térképeken szereplő településnevek felvételére, a kézi névfelvétel mellett döntöttem. Ennek során egyesével megnyitottam a beszkenelt térképeket és elkezdtem beírni Excel táblázatkezelőbe a térképen található településneveket. Ha a térképen voltak jelölve közigazgatási egységek, akkor azok alapján haladtam, hogy ne hagyjak ki neveket. Amennyiben a térképen nem csak magyar névalakok voltak megtalálhatóak, akkor egy új oszlopba felvettem az államnyelvi névalakokat is. Miután végeztem egy térképpel következett a következő szakasz, amikor az összerendelhetőség érdekében a felvett nevek mellé felvettem a mai államnyelvi névalakokat is, eredeti betűrendszer szerint, mivel cirill betűket használó országok esetén a letöltött adatbázisban cirill betűs írással voltak megtalálhatóak ezek a településnevek. Annak érdekében, hogy megtaláljam ezeknek a településeknek az államnyelvi névalakját a Wikipédiát hívtam segítségül, mivel itt megtaláltam minden térképen szereplő település államnyelvi névalakját, ha a magyar nyelvű oldalon nem találtam, akkor átállítottam a nyelvet államnyelvire és onnan másoltam be a táblázatba. Miután végeztem térképekkel, mindegyiket külön mentettem térképenként. Hogy a QGIS térinformatikai szoftverben fel tudjam használni ezeket a fájlokat, a mentés másként menüben az „Unicode-os szöveg (*.txt)” lehetőséget választottam, mivel csak ilyen formátumban kimentve lehetséges úgy behívni a QGIS szoftverbe, hogy a karakterkódolás minden névalak esetében sértetlen maradjon. QGIS programban a behívás a következőképp zajlott: meg kellett nyitni a „Tagolt szöveg hozzáadása” gombot (add delimited text layer), majd a „Fájlnév” opciónál kell megadni a txt fájl elérési útvonalát, valamint a „kódolás” opciónál az „ucs-2,, lehetőséget kell beállítani, hogy rendben legyen a karakterkódolás (4.ábra).



4. ábra A txt. fájl hozzáadása a QGIS programban

Miután behívtuk a szövegfájlt következhet a letöltött shapefile állományhoz csatolás. Ezt a műveletet a rétegkezelőben tehetjük meg, ha jobb gombbal kattintunk a rétegen és a tulajdonságok menüpontban megnyitjuk az összekapcsolások menüpontot (5.ábra). Itt a zöld plusz gombbal adhatunk hozzá új kapcsolatot. Ebben a menüben meg kell adni a behívott szövegfájlt, amit hozzá akarunk adni a réteghez, valamint a kapcsolódó mezőket. Emiatt a művelet miatt volt fontos adatbázisépítés közben az Excelben megadni a mai államnyelvi névalakokat, mivel ezzel a mezővel lehet összekapcsolni a letöltött shapfile adatbázisában lévő „name” mezőt.



5. ábra Szöveg réteg összekapcsolása a shapefile-al

Az eredmény pedig a honlap adatbázisa számára feldolgozható fájlformátumba történő kimentés. Ez nem megy egy lépésben. A QGIS rétegkezelőjében a rétegre jobb gombbal kattintva export-> elem mentése másként opcióval tudjuk az eredményt menteni esetünkben egyszer MS Office Open XML munkafüzet [XLSX] formátumba, másodszor pedig GeoJSON formátumban. Az XLSX fájl az MySQL adatbáziskezelőhöz szükséges, a GeoJSON pedig a Leaflet térképhez. Az XLSX fájl Excelben történő megnyitása után menthetjük el, az adatbáziskezelő számára elfogadható formátumba a CSV UTF-8 (pontosvesszővel tagolt) (*.csv) opcióra kattintva.

6.2 Az adatbáziskezelő

Ha elkészült a csv fájl szükséges lesz az adatbáziskezelő program megnyitása. Én a honlap futtatásához szükséges webszervernek az USBWebservert használtam (6.ábra). Ez a program tartalmazza a honlap futtatásához szükséges környezetet, az adatbázishátteret. A program telepítése után a mappájában megnyitva az usbwebserver.exe futtatásával nyitható meg. [20] Később ez a tesztkörnyezet könnyen migrálható akár másik, hasonló beállításokkal rendelkező szerverre.



6. ábra Az USBWebserver az exe fájl futtatása után

A phpMyAdmin gombra kattintva nyitható meg az adatbáziskezelő a böngészőben. A felületen bal oldalon találhatóak az adatbázisok, ahol létrehoztam a „honlapok” nevű adatbázist. Ebbe az adatbázisba importáltam be az adataimat. Ez a honlapok fülre kattintva importálás műveletre kattintással végezhető el. Itt a „Fájl kiválasztása” gombbal történik meg a csv behívása. A karakterkészletnek UTF-8-nak lennie. A fájl formátumot a program automatikusan felismeri és a „Formátum specifikus beállítások” menüben az „Oszlopok elválasztása ezzel” pontban az automatikusan megjelenő vesszőt pontosvesszőre kell átírni, hogy lefusson az importálás. Be kell pipálni, hogy a fájl első sora a tábla oszlopneveit, hogy az első sor ne legyen az adat része (7.ábra).

A fájl lehet tömörített (gzip, zip) vagy tömörítetlen.
A tömörített fájl nevének **[formátum].[tömörítés]** végződésűnek kell lennie. Például: **.sql.zip**

Számítógép tallózása: **Fájl kiválasztása** magyaror...adatok.csv (Max: 2,048KB)

A fájl karakterkészlete: **utf-8**

Részleges importálás:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)

Az elejétől kihagyandó sorok száma:

Formátum:

CSV

Megjegyzés: Ha a fájl több táblát tartalmaz, akkor azok egyesítve lesznek

Formátum specifikus beállítások:

☐ A táblaadatok lecserélése fájlra

Oszlopok elválasztása ezzel:

Oszlopok körbezárása ezzel:

Oszlopok escape-elése ezzel:

Sorok végződése ezzel:

☒ A fájl első sora tartalmazza a tábla oszlopneveit (ha ez nincs kiválasztva, az első sor az adat része lesz)

☐ Ne legyen megszakítás INSERT hiba esetén

Indítás

7. ábra A phpMyAdmin adatbáziskezelő beállításai .csv fájl importálása esetén

Ha hibaüzenet nélkül lefutott a művelet, akkor létrejön a tábla, ami átnevezhető bárhogyan, én esetemben data lett a tábla neve (8. ábra). Esetemben a táblának 16940 sora lett, de természetesen a tábla számtalan üres mezőt tartalmaz, mivel csak azok az oszlopok lettek kitöltve, amelyekhez az adott térképen szerepelt a megfelelő névalak.

phpMyAdmin interface showing the 'data' table. The table contains 16940 rows. The SQL query displayed is: `SELECT * FROM `data` ORDER BY `data`.`hu48` DESC LIMIT 0, 30`. The table structure is as follows:

fclass	population	name	layer	hu06	most06	hu11	most11	hu27	most27	hu48	all48	most48	all61	hu04
city	81494	Žilina	sk	Zsolna	Žilina	Zsolna	Žilina	Zsolna	Žilina	Zsolna	Žilina	Žilina		Zsolna
town	10198	Jibou	ro	Zsibó	Jibou	Zsibó	Jibou	Zsibó	Jibou	Zsibó	Jibou	Jibou		Zsibó
town	6393	Žarnovica	sk							Zsarnóca	Žarnovica	Žarnovica		Zsarnóca
village	1484	Zsáka	hu							Zsáka	Zsáka	Zsáka		
town	9107	Жабаль	sr							Zsabyia	Žabali	Жабаль		Zsabyia
city	51471	Comboor	sr			Zombor	Comboor	Zombor	Comboor	Zombor	Sombor	Comboor		Zombor
village	3163	Slovenská Ľupča	sk							Zolyomlpcse	Slovenská Ľupča	Slovenská Ľupča		
village	4135	Podbrezová	sk			Zolyombrezo	Podbrezová			Zolyombrezo	Podbrezová	Podbrezová		
village	333	Breziny	sk							Zolyomberezna	Breziny	Breziny		
town	42778	Zvolen	sk			Zolyom	Zvolen	Zolyom	Zvolen	Zolyom	Zvolen	Zvolen		Zolyom
town	6691	Zirc	hu			Zircz	Zirc			Zirc	Zirc	Zirc	Zirc	
city	56202	Zaláu	ro	Zilah	Zaláu	Zilah	Zaláu	Zilah	Zaláu	Zilah	Zaláu	Zaláu		Zilah
town	20363	Сента	sr	Zenta	Сента	Zenta	Сента	Zenta	Сента	Zenta	Senta	Сента		Zenta
village	1493	Uhrovec	sk			Zavugrócz	Uhrovec			Zayugrócz	Uhrovec	Uhrovec		

8. ábra Adatbázistábla a phpMyadmin adatbáziskezelőben

7. A Weboldal felépítése

7.1. A weboldal tartalma

A weboldal három fő részre bontható.

Első fő rész az oldal felső része, ahol bal oldalon található a weboldal címe, jobb oldalon pedig a weboldal készítője, témavezető, link a weboldal leírásához, az intézet honlapjához, valamint egy Google Térképhez, ami az intézet elhelyezkedését jelöli.

A következő egysége a honlapnak a sárgával jelölt középső rész, ami a keresőket tartalmazza. Itt található meg a településekhez a kereső, ami gépelés alapú, az országrészekhez, valamint az egyes térképeknek a névanyagához a kereső, amelyek esetében a keresés gombkattintással végezhető el.

A harmadik része a weboldalnak az eredményeket szemléltető rész, ahol bal oldalon található a táblázat helye, jobb oldalon pedig a térkép, amin a keresett települések jelennek meg.

7.2. Keresés menete

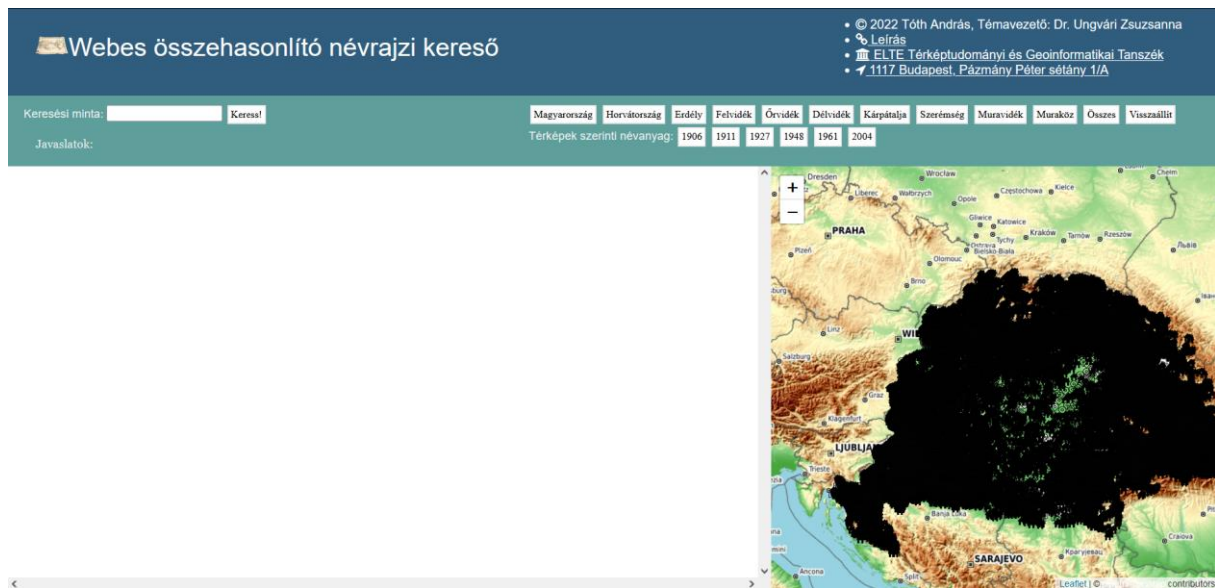
A weboldal kialakításakor célként fogalmazódott meg, hogy a weboldal tartalmazzon egy olyan keresőt, amely lehetőséget nyújt településnév szerinti keresésre. Ezért elsőként a weboldalra elhelyezésre került egy kereső, amibe beírva a településnevet rá lehessen keresni, egy adott településnévre és eredményként egy olyan táblázatot kapjunk, amely tartalmazza a 6 térképen előforduló településneveket, illetve a letöltött adatbázisból vett mai államnevet. Gépelés közben a kereső alatt a **Javaslatok:** kezdetű sorban feltűnnek javaslatok arról, hogy milyen településnevek találhatók meg az adatbázisban, ezzel is elkerülve az esetleges felesleges keresést. A **Keress!** gombra történő kattintás után megjelenik a táblázat az adatokkal. Az adatbázis első sorában az eredmények gomb formájában jelennek meg, amik azt a célt szolgálják, hogy a gombra kattintva az eredmény rész jobb oldalán elhelyezett térkép fókusza a Kárpát-medencéről a keresett településre helyeződjék át, megjelenik a település településkategória szerinti jele és egy felugró ablakban megjelenik a település államneve, illetve a népessége.

A következő keresési forma, ami kialakításra került lehetőséget nyújt területi adatok lekérdezésére. Az adatbázisépítéskor az országhatárok változása miatt aszerint kerültek felvételre a településnevek, hogy mely országhoz tartoznak napjainkban. Éppen ezért a keresőben nagyrészt a mai határok mentén különülnek el a területi lekérdezés lehetőségei, kivétel Horvátország környezete, mivel Horvátországot az I. világháború előtti térképek

esetén is külön kezeltem. Így lett végül 10 területi lekérdezési lehetőség, amely magába foglalja a mai Magyarországot, Erdélyt (azokat a területeket, ami ma Romániához tartoznak); Felvidéket (azokat a területeket, amelyek ma Szlovákiához tartoznak), Kárpátalját (ma Ukrajnához tartozó települések), Délvidéket (ma Szerbiához tartozó, korábban Magyarországhoz tartozó települések), Őrvidéket (ma Ausztriához tartozó települések), Muravidéket (ma Szlovéniához tartozó települések), Muraközt (ma Horvátországhoz, korábban Magyarországhoz tartozó települések), Szerémséget (ma Szerbiához, korábban Horvátországhoz tartozó területeket), valamint Horvátországot (ma is és korábban is Horvátországhoz tartozó területek). Ezekre a gombokra kattintva a kiválasztott területen fekvő települések névalakjai kerülnek kilistázásra, illetve a térkép fókusza ezekre a területekre kerül át, valamint megjelennek a települések kategória szerinti szimbólumai, amikre kattintva egy felugró ablakban látható a település mai államneve és népessége. A táblázatban a települések mai államnyelvi névalakjai szintén kattinthatóak, amivel a térképen a kiválasztott település válik a fókuszt középpontjává, a nézet felnagyítódik, majd felugranak a település adatai.

Abban a sorban, ahol a területi keresés található megtalálható még az **Összes**, valamint a **Visszaállít** gombok is. Az **Összes** gomb segítségével előhívhatjuk a térképen az összes település szimbólumát és a szimbólumra kattintva a település államnevét és népességét, illetve a táblázatba kilistázhatjuk az adatbázisban szereplő összes névanyagot. A **Visszaállít** gomb segítségével az oldal eredeti állapotát állíthatjuk vissza (9. ábra).

A következő sorban található keresési lehetőség a felhasznált térképek szerinti keresés. Itt a térképek kiadásának az évszáma a keresőgombok felirata. Az évszámra kattintva a táblázatban megjelenik a lista azokról a nevekről, amik az ebben az évben kiadott általam felhasznált térképen találhatóak meg. Ebben az esetben a térkép változatlan marad.



9. ábra A weboldal külleme

8. A weboldal fejlesztése

A weboldal fejlesztéséhez szükséges környezetet a már korábban említett USBWebsever nevű program biztosította, ami olyan népszerű webes szoftverek kombinációja, mint az Apache, MySQL, PHP és MyAdmin. Az USBWebserver-rel bárhol és bármikor lehetőség nyílik PHP weboldalak fejlesztésére és megjelenítésére, mindezt úgy, hogy az egész mozgatható. [20]

8.1. A felhasznált programozási nyelvek

8.1.1. HTML

A HTML (Hyper Text Markup Language) egy szabványos jelölőnyelv, amely elemek sorából a weboldal szerkezetét írja le. Az elemek leírják a böngészőnek a tartalom megjelenítés módját. A HTML-elemek jelölik a weboldal olyan elemeit, mint a címsor, a bekezdés vagy hivatkozás. Tulajdonképpen a HTML kód írja le a weboldal külsőjét. Ezt a CSS nyelv segítségével teszi, ami megadja a HTML fájlok stílusát. A CSS-el történő formázás három módon történhet: *external*, ha a stílust külön .css fájlban adjuk meg, *internal* módon, ha egy style elemként a HTML fájlban, illetve *inline* módon, ha egy elem létrehozásakor adjuk meg a stílusbeli információkat. A kódban mindhárom módszerre található példa. Az index.php fájl tartalmazza az oldal HTML kódját. [7] [8]

8.1.2. PHP

A PHP egy olyan szerveroldali szkriptnyelv, amely egy hatékony eszköz dinamikus és interaktív weboldalak készítéséhez. A PHP széles körben használt, ingyenes és hatékony alternatívája a versenytársaknak, például a Microsoft ASP-jének. A weboldalon a keresőmotor létrehozása is PHP-ben történt. A php kódokat az orszagiir.php, a kereses.php és a gethint.php fájlok tartalmazzák. [9]

8.1.3. JavaScript

A JavaScript vagy JS leginkább webes céllal létrehozott interpretált nyelv, amelyet általában arra használnak, hogy interaktív effektusokat hozhassunk létre a böngészőnkben. C nyelven alapul és egy ideje támogatja az objektumorientáltságot. Az interpretált nyelv az azt jelenti, hogy futás közben értelmezett, nincs szükség fordítóra. Az objektumorientáltság úgy jelenik meg benne, ahogy számos, egyéb programnyelv esetén: lehet osztályokat létrehozni, azokat származtatni egymásból. A JavaScript nyelven nem adhatjuk meg deklaráláskor a változók típusait, illetve azoknak nincs is típusa. Az adatoknak van típusa, de a változóknak nincs, és egy változóban tárolt adat típusa változhat. Ebből kifolyólag gyengén típusos nyelv.

A weboldalon létrehozott webtérkép egy Leaflet-térkép, ami szintén JavaScript nyelven alapszik. A Leaflet a vezető nyílt forráskódú JavaScript-függvénykönyvtár a mobilbarát

interaktív térképekhez. Mindössze 39 KB méretű, mégis rendelkezik minden leképezési funkcióval, amelyre a legtöbb fejlesztőnek szüksége van. A Leafletet az egyszerűség, a teljesítmény és a használhatóság szem előtt tartásával tervezték. Hatékonyan működik az összes főbb asztali és mobilplatformon, sok pluginnal bővíthető, könnyen használható és jól dokumentált API-ja, valamint egyszerű, olvasható forráskódja van. A JavaScript kódok az index.php fájl `<script>` és `</script>` címkék közti részekben találhatóak. [11][21][22]

8.1.4. AJAX

Az AJAX az **A**synchronous **J**avascript **A**nd **X**ml rövidítése. Ezzel a technikával úgy tudjuk a honlapon lévő adatainkat frissíteni, hogy nem kell a teljes weboldalt újra betölteni (szinkron mód), hanem aszinkron módon, egy esemény bekövetkeztére (pl. megnyomunk egy gombot, vagy leütünk egy billentyű stb.) töltődik be vagy frissül a tartalom. Ehhez kombináltan használtam a JavaScriptet és a PHP-t. Működését egy, a honlapomon előforduló példával illusztrálnám. Amikor a szövegmezőbe beírok egy településnevet, és rányomok a keress gombra, akkor a kattintással meghívok egy JS függvényt, amely elküld egy XHR (XMLHttpRequest-et) kérést. Ez a háttérben meghívja a PHP szkriptet, amellyel lefut a keresés az adatbázisban és visszatér az eredmény (ennek lépései: kapcsolódik az adatbázishoz, keres az adatbázisban, majd a keresési eredményeket feltölti egy HTML táblázatba). Az eredményt pedig betöltjük ebben az esetben a weboldal egy div-jébe (ahol az eredménytáblát szeretnénk megjeleníteni). [10]

8.2. A településnév szerinti keresés létrehozása:

8.2.1. A településnév szerinti keresés létrehozása – HTML kód

Munkám során az első keresési funkció, amit létrehoztam az a településnév szerinti keresés volt. Ez, mint említettem két részből áll. Az első az, amikor begépel a felhasználó a keresendő kifejezést, és minden egy leütésre (onkeyup esemény), megmutatja a hasonló neveket az adatbázisból. A második része az, hogy a felhasználó a név vagy névrészlet beírása után a keress gombra kattint, és visszakapja az adatbázisból a hasonló vagy egyező találatokat. Ennek a két funkciónak a kódrészleteit három részre bontva magyarázom meg. Először a keresőt HTML-ben formáztam meg az alábbi kód szerint:


```

<form method="get">
Keresési minta: <input type="text" id="minta"
onkeyup='showHint(this.value)' />
    <input type="button" value="Keress!" onclick="betoltes();" />
    <p style="text-align:left; background-color:
#FFFF00">Javaslatok: <span id="txtHint" style='overflow-y:scroll;
background-color: #FFFF00;height: 100px; width:50%;padding-
left:20px;float:left;text-align:left;position:absolute;z-index:
10;'></span></p>
</form>

```

A kód először megadja, hogy a kereső mely HTTP (Hypertext Transfer Protocol) módszert alkalmazza. A HTTP teszi lehetővé a kommunikációt a kliens és szerver között kérés-válasz protokollként. A választás ez esetben a GET módszer mellé esett. Az `<input type="text" id="minta" onkeyup='showHint(this.value)' />` elem megadja a keresőnek, hogy a bevétel módja szöveg lesz, ami a „minta” értéket kapja és a **showHint** funkció segítségével végez keresést. Megadtam az `<input type="button" value="Keress!" onclick="betoltes();" />` elemmel a **Keress!** gombot is, aminek a segítségével zajlik a **betoltes()** funkció futtatása, ami kiírja az eredményeket. A `</p>` elem pedig a txtHint funkció futtatására szolgál, ami a beírt szöveg alapján az adatbázisban található lehetséges településneveket írja ki. A HTML kód ezen kívül még a stíuselemeket tartalmazza.

A HTML-kódban történik az eredménytáblázat div elemként történő megformázása is az alábbi kód szerint:

```

<div id="tablazat" style='width: 63%;height: 558px;margin-right:
37%;overflow:scroll;position:absolute;index-z:10;'>

```

Az eredménytáblázat a következő formában jelenik meg (10. ábra):

Mai államnév	Mai magyarnév	Mai latinbetűs államnév	1961-es nevek	1948-as magyar névalakok	1948-as államnevek	1927-es településnevek	1911-es településnevek	1906-os településnevek
Miskolc	Miskolc	Miskole		Miskolc	Miskolc	Miskolc	Miskolcz	Miskolcz

10. ábra Az eredménytábla Miskolcra történő keresés esetén

8.2.2. A településnév szerinti keresés létrehozása – JS kód

A következő lépésben a JS kódok olvashatók. Ha begépelünk valamit a szövegmezőbe, a **showHint** függvény segítségével keressük meg a hasonló találatokat (az onkeyup eseményre).

```

function showHint(str) {
    var xhttp;
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    }
    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("txtHint").innerHTML =
this.responseText;
        }
    };
    xhttp.open("GET", "gethint.php?q="+str, true);
    xhttp.send();
}

```

Itt az történik, hogy a függvény beolvassa a keresőmezőbe beírt szöveget, megnyitja a *gethint.php* fájlt aszinkron módon, és lefuttatja a keresést, az eredményt pedig elküldi a böngészőnek. A felhasználó a hasonló neveket fogja látni a „Javaslatok” alatt.

A következő a **betoltes függvény**, ami a következő JS kódból áll (ez tölti be az eredménytáblázatot):

```

function betoltes() {
    if (window.XMLHttpRequest) { // code for IE7+, Firefox,
    Chrome, Opera, Safari
        xhttp=new XMLHttpRequest();
    }

    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var valasz=document.getElementById("result");
            valasz.innerHTML = xhttp.responseText;
        }
    };

    var minta =document.getElementById("minta").value;
    xhttp.open("GET", "kereses.php?varos=" + minta,

```

```

true);

        xhttp.send();

    }

```

A **betoltes** funkció az, ami a **Keress!** gombra történő kattintás után a beírt szöveget elküldi a kereses.php fájlba, lefuttatja a keresés és kiírja az eredményt a weblapon.

8.2.3. A településnév szerinti keresés létrehozása – PHP kód

A fent említett JavaScript funkciók az alábbi PHP fájlokkal dolgoznak együtt.

Elsőként a *kereses.php*:

```

<?php
    $minta=$_GET["varos"];

    $db=mysqli_connect('localhost','root','usbw','honlapok');
    mysqli_query($db,'set names utf8');

    $eredmeny=mysqli_query($db,"select * from data where name
like '{$minta}%' or hu06 like '{$minta}%' or hu11 like '{$minta}%'
or hu27 like '{$minta}%' or hu48 like '{$minta}%' or hu04 like
 '{$minta}%' order by name");

    print mysqli_error($db);
    if (mysqli_num_rows($eredmeny)>0)
    {
        print "<table><tr><th>Mai államnév</th><th>Mai
magyarnév</th><th>Mai latinbetűs államnév</th><th>1961-es
nevek</th><th>1948-as magyar névalakok</th><th>1948-as
államnevek</th>
        <th>1927-es településnevek</th><th>1911-es
településnevek</th><th>1906-os teepülésnevek</th>";

        while ($sor=mysqli_fetch_assoc($eredmeny))
            print "<tr><td><input type='button' id='zoomolj'
value='\".$sor['name'].\"'
onclick='zoomToVaros(\"\".$sor['name'].\"'\");'></td><td>\".$sor['hu04']
.\"</td><td>\".$sor['la04'].\"</td><td>\".$sor['all61'].\"</td><td>\".$sor
['hu48'].\"</td>

            <td>\".$sor['all48'].\"</td><td>\".$sor['hu27'].\"</td><td>\".$sor['
hu11'].\"</td><td>\".$sor['hu06'].\"</td></tr><td></td>";

        print "</table>";
    }

```

```

        else
            print "<b>Nincs találat!</b>";
        mysqli_close($db);
    ?>

```

A *kereses.php* tartalmazza az adatbázishoz történő kapcsolódást és az abban való keresés lefuttatását. A `$minta=$_GET["varos"]` értékadással kiolvassuk, a betöltés függvényben elküldött nevet vagy névrészletet, amire keresni fogunk az adatbázisban. Az adatbázishoz történő csatlakozást biztosítja az `$db=mysqli_connect('localhost','root','usbw','honlapok')` elem. A következő sor a kódolást állítja UTF-8-ra. Ezután történik a **minta** mező értéke alapján az adatbázisból a megfelelő attribútumok kiválasztása, szükséges oszlopok elnevezése és importálása, valamint, ha nem talál a keresésnek megfelelő elemet, akkor a Nincs találat! kiírása. A definiáltak szerint a kereső elküldi a weboldalnak az eredményeket, ami a HTML-ben megadottak szerint egy táblázatban jeleníti meg azt.

8.3. A terület szerinti keresés létrehozása

8.3.1. A terület szerinti keresés létrehozása – HTML kód

A terület szerinti kereső esetében is a weblapon látható kereső kialakítása HTML-ben történt a következő kóddal:

```

<form method="get">
<input type="button" id="bttnmagyar" value="Magyarország"
onclick="terkepurit();orszag('hu');"/>
<input type="button" id="bttnhorvat" value="Horvátország"
onclick="terkepurit();orszag('hr');"/>
<input type="button" id="bttnerdely" value="Erdély"
onclick="terkepurit();orszag('ro');"/>
</form>

```

Ebben az esetben is GET módszerrel történik a szerverrel a kommunikáció. Az kereső típusát az *input type* elem értéke *button*, tehát gomb, mivel ezt a fajta keresés pont szerinti kereséssel szerettem volna véghez vinni. Az *id* értékével definiáltam a gomb azonosítóját, a *value* pedig megadja a gomb értékét, a feliratot, ami a gombon szerepel. Az *onclick="orszag('Magyarország')* megadja, hogy a gombra kattintva az oldal milyen értéket küldjön az ezt felhasználó JS funkciónak.

Az eredménytáblázat ugyanabban a div elemben jelenik meg, mint amit a szöveges keresőnél már definiáltam.

Példa a táblázatban történő megjelenítésre (11. ábra):

Mai államnév	Mai magyarnév	Mai latinbetűs államnév	1961-es nevek	1948-as magyar névalakok	1948-as államnevek	1927-es településnevek	1911-es településnevek	1906-os településnevek
Бала					Đala			
Бурђево								
Бурђин								
Јабланка								
Јабука								
Јазово								
Јанков Мост								
Јаношиќ								
Јарковац	Árkod	Jarkovac						
Јасеново				Karasjeszenő	Jasenovo			
Јаша Томић	Módos	Јаша Томић		Módos	Jaša Tomić			
Јерменовци								
Љутово								
Његошево								
Ада	Ada	Ada			Ada		Ada	Ada

11. ábra Eredménytábla "Délvidék" gombra történő kattintás esetén

8.3.2. A terület szerinti keresés létrehozása – JS kód

```
function orszag(orszagnev) {
    if (window.XMLHttpRequest) { // code for IE7+, Firefox,
    Chrome, Opera, Safari
        xhttp=new XMLHttpRequest();
    }
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var valasz=document.getElementById("result");
            valasz.innerHTML = xhttp.responseText;
        };
        xhttp.open("GET", "orszagkiir.php?orszag=" +
    orszagnev, true);
        xhttp.send();
        telepBetolt(orszagnev);
    }
}
```

Az **ország** függvény az, ami a számunkra megfelelő gombra történő kattintás után a beírt szöveget elküldi az *orszagkiir.php* fájlba, lefuttatja a keresés és kiírja az eredményt a weblapon. A településeket **telepBetolt(orszagnev)** függvény tölti be.

8.3.2. A terület szerinti keresés létrehozása – PHP kód

A gombbal történő keresés esetén az *orszagkiir.php* fájl olvasódik be. Ez a korábban részletezett módon hozzákapcsolódik az adatbázishoz, ki keresi a megfelelő attribútumokat és a kódban definiált oszlopokba rendszerezve elküldi a weboldalnak az eredményt, ami a HTML kódban definiáltak szerint táblázatban megjeleníti azt.

```
<?php
$ország=$_GET['ország'];
if (isset($ország)) {
    switch ($ország) {
        case "hu":
        {
            $db=mysqli_connect('localhost','root','usbw','honlapok');
            mysqli_query($db,'set names utf8');
            $result=mysqli_query($db,"select * from data where layer
like 'hu' order by name");

            print mysqli_error($db);

            if (mysqli_num_rows($result)>0)
            {
                print "<table><tr><th>Mai államnév</th><th>Mai
magyarnév</th><th>Mai latinbetűs államnév</th><th>1961-es
nevek</th><th>1948-as magyar névalakok</th><th>1948-as
államnevek</th>
                <th>1927-es településnevek</th><th>1911-es
településnevek</th><th>1906-os teepülésnevek</th>";
                while ($sor=mysqli_fetch_assoc($result))
                    print "<tr><td><input type='button' id='zoomolj'
value='".$sor['name']."'
onclick='zoomToVaros(\"".$sor['name']."\");'></td><td>".$sor['hu04']
.</td><td>".$sor['la04'].</td><td>".$sor['all61'].</td><td>".$sor
['hu48'].</td>
                    <td>".$sor['all48'].</td><td>".$sor['hu27'].</td><td>".$sor['
hu11'].</td><td>".$sor['hu06'].</td></tr><td></td>";
                print "</table>";
            }
        }
    }
}
```

```

    }
    else
        print "<b>Nincs találat!</b>";
    mysqli_close($db);
}
break;
}
}
?>

```

8.4. A térkép létrehozása

A weboldal létrehozásakor alapvető igényként merült fel, hogy az oldal tartalmazzon egy szemléltető térképet, ami képes térben elhelyezni a keresés eredményeit, ami hasznos arra nézve, hogy a felhasználónak nem kell elhagynia az oldalt, amikor rákeres egy településre és meg szeretné tudni, hogy az pontosan hol van. Ennek érdekében döntöttem a Leaflet mellett, ami egy olyan nyílt forráskódú, ingyenesen használható JavaScript könyvtárat takar, amit a legtöbb asztali és mobil platform támogat. A Leaflet-et felhasználása általában HTML oldalakon történik úgy, hogy először a html kód *head* részében meghívásra kerül a *leaflet* JS könyvtár és a hozzá tartozó stílusfájl (.css) az alábbi kódban látható módon, majd div elemként adják hozzá a kódban a JavaScriptben megírt Leaflet-et.

```

<link rel="stylesheet" href="leaflet/leaflet.css" />
<script src="leaflet/leaflet.js"></script>

```

8.4.1. A térkép létrehozása – HTML

A térkép helyét egy div elemként adtam hozzá az oldal kódjához, és a lap bal alsó oldalán helyeztem el.

```

<div id="terkep_helye" style='width: 37%;height: 538px;margin-left: 63%;position:absolute;index-z:11'></div>

```

A *terkep_helye*-nek nevezett div elem a teljes oldal szélességének 37%-a, magassága pedig 538px. A bal oldali szélétől való távolságát 63%. Szükség volt a pozíció *absolute*-ra történő állítására és *index-z* érték megadásra is, hogy a térkép a táblázattal egy sorban tudjon megjelenni és ne az alatta következő sorban.

8.4.1. A térkép létrehozása – JS

Az index.php fájlban script elemként van hozzáadva a Leaflet kód része. Először létrehoztam a térképet, mint változót és beállítottam a kiindulási nézet középpontjának koordinátáját és a hozzá tartozó nagyítást:

```
var map=L.map('terkep_helye').setView([47.475,19.062], 6.4);
```

Következő lépésként az alaptérképnek az openTopomap webes térképét használtam.

```
L.tileLayer('https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors'
}).addTo(map);
```

Ezután történt meg a térképen megjelenítendő kategorizált településikonok és a rájuk kattintva megnyíló popup ablakok definiálása. Először létrehoztam az ikonokat, amelyek az adatbázisban előforduló településkategóriáknak megfelelően hozzárendeltem a településkategóriák szimbólumait, melyeket korábban png formátumban mentettem ki.

```
var ikonok={
  'village': 'falu.png',
  'null': 'falu.png',
  'town': 'varos.png',
  'locality': 'varos.png',
  'county': 'varos.png',
  'city': 'city.png',
  'national_capital': 'city.png'
}
```

Az ikonok létrehozása után történt meg az adatbázishoz való kapcsolódás és a telep változó definiálása. A **telepBetolt(layerName)** funkciót is itt definiáltam.

```
var telep;
function telepBetolt(layerName) {
  map.removeLayer(telep);
  fetch('nevek.geojson').then(r=>r.json()).then(d=>{
    telep= L.geoJSON(d,{
      pointToLayer: function (f,ll) { if
(f.properties.layer==layerName) {
        return L.marker(ll,{icon: L.icon({
          iconUrl: ikonok[f.properties.fclass],
          iconSize: [10,10],
          iconAnchor: [0,0],
```



```

        popupAnchor: [0,0]
    }));
    }
    }

    }).addTo(map).bindPopup(function (telep) {return
"Településév: "+telep.feature.properties.name+"\<br>Népesség:
"+telep.feature.properties.population+"\nFő";
    })

    map.fitBounds(telep.getBounds());
    });
    }
    fetch('nevek.geojson').then(r=>r.json()).then(d=>{
        telep=      L.geoJSON(d,{
        pointToLayer: function (f,ll) {
        return L.marker(ll,{icon: L.icon({
            iconUrl: ikonok[f.properties.fclass],
            iconSize: [10,10],
            iconAnchor: [0,0],
            popupAnchor: [0,0]
        }}});
        }

        }).addTo(map).bindPopup(function (telep) {return
"Településév: "+telep.feature.properties.name+"\<br>Népesség:
"+telep.feature.properties.population+"\nFő";
        })
    });

```

A Leaflet esetében új adatbázisra volt szükség, mivel a MySQL nem tartalmazta a szükséges koordinátákat. Ezért QGIS szoftverben a korábban exportált réteget ezúttal GEOJSON fájlformátumban mentettem ki, ami már felhasználható volt a Leaflet térképhez is.

A kódban a **telep** változó létrehozása után *fetch* függvénnyel kapcsolódtam a **nevek.geojson** adatbázis fájlhoz. Az adatbázisból szerzett adatok segítségével a településekhez hozzárendeltem a megfelelő szimbólumokat, beállítottam azok méretét és a *bindPopup* funkcióval egy kattintásra felugró ablakba kiírtam a település nevét és lélekszámát.

Ezután következett a **zoomToVaros(nev)** függvény definiálása. Ez a funkció felel ezért, hogy a táblázatban a település államnévére kattintva a térkép a kiválasztott településre nagyítson.

```

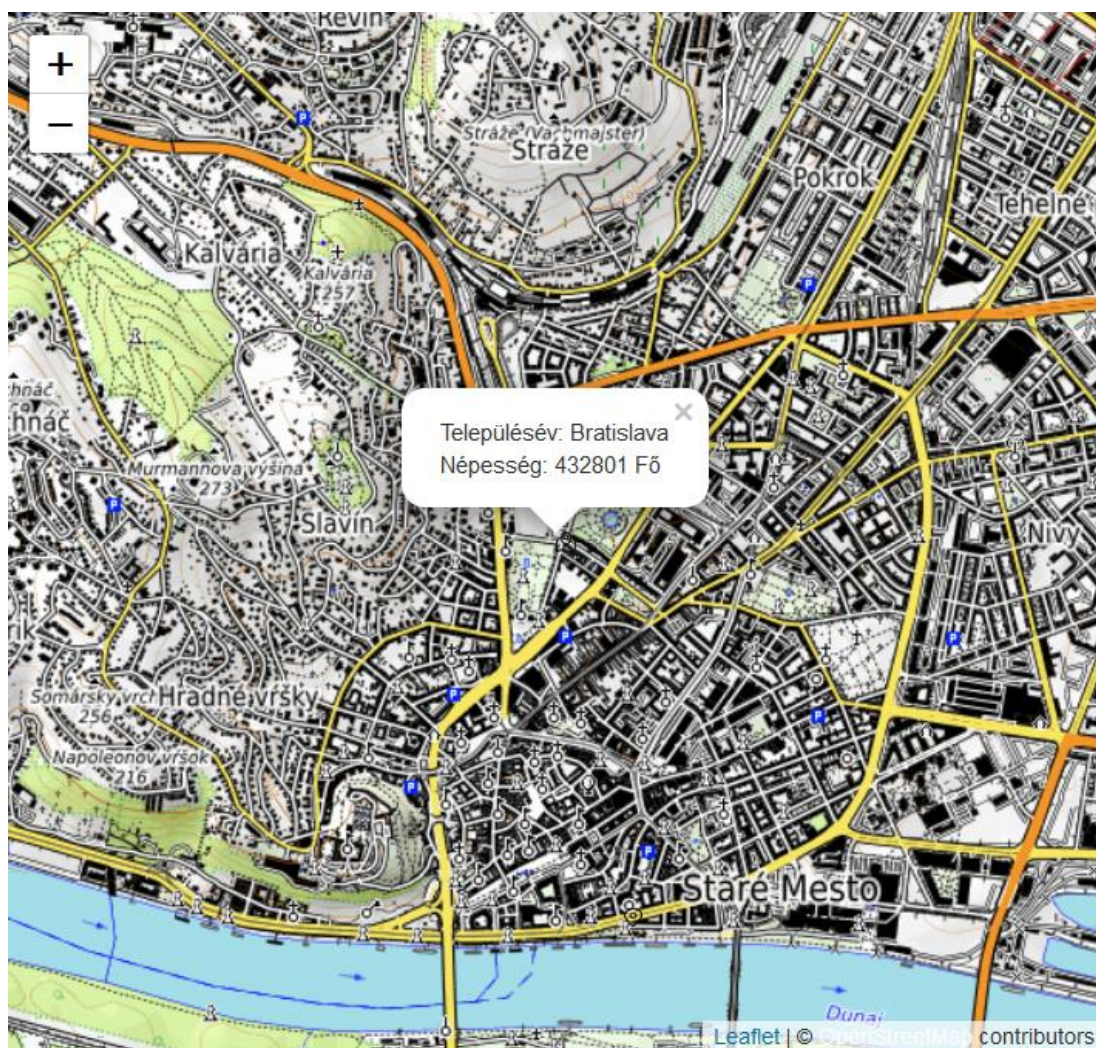
function zoomToVaros(nev) {
map.removeLayer(telep);

```

```

var data = fetchJSON('nevek.geojson').then(function(data) {
    data.features.forEach(function(feature) {
        if (feature.properties.name==nev){
            var coord=feature.geometry.coordinates;
            var
popupData="Településév:"+feature.properties.name+"\<br>Népesség:
"+feature.properties.population+"\nFő";
            var popup = L.popup().setLatLng([coord[1],coord[0]], 14)
                .setContent(popupData)
                .openOn(map);
            map.setView([coord[1],coord[0]],14);
            telep.addTo(map);
        }
    });
});
}

```



12. ábra A felugró popup ablak a településnévvel és népességgel

A **map.removeLayer(telep)** funkció felel a korábban behívott települések szimbólumainak az eltávolításáért. A **fetchJSON** kifejezéssel hozzákapszolódik a **nevek.geojson** adatbázishoz és megvizsgálja, hogy a táblázatban kiválasztott név az adatbázis mely sorának **name** attribútumával egyenlő. Végül, ahol megtalálja a keresett települést a térkép nézetét az adatbázisban található koordináták szerinti helyre állítja és a nagyítást pedig a 14-es zoomszintre.

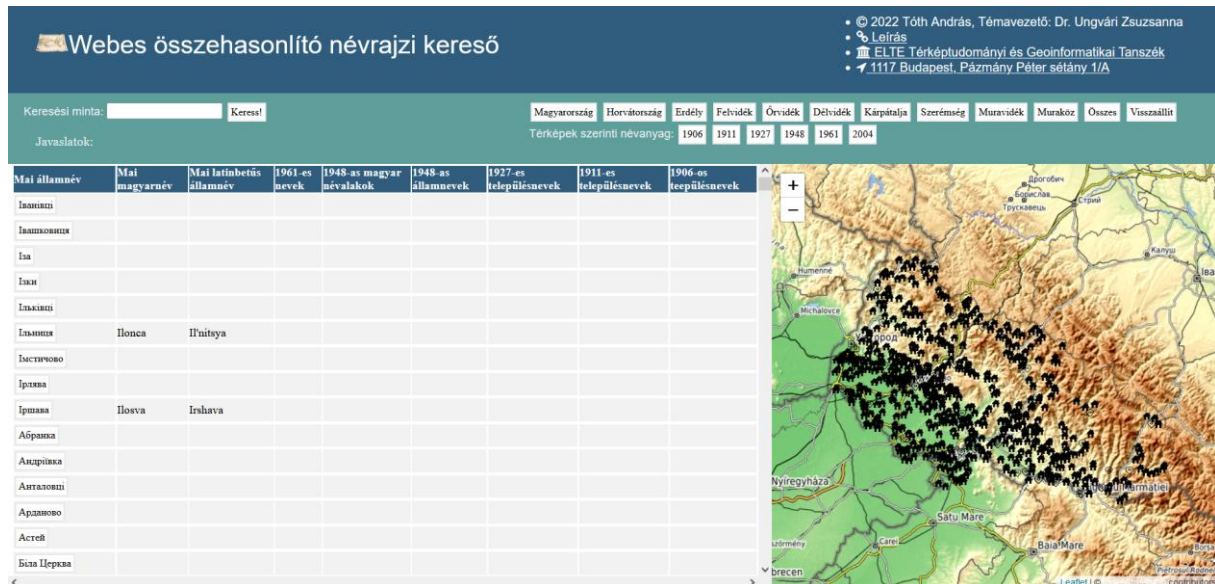
A terület szerinti keresésnél a térképi nézet középpontjába a megadott terület kerül térkép és emellett betölti a településszimbólumokat is. A fókusz áthelyezéséhez a **map.flyTo** eseményt használtam. Erre az alábbi kód szolgál:

```
document.getElementById("bttnmagyar").addEventListener("click",
function () {

    map.removeLayer(telep);
```

```
map.flyTo([46.475,19.062], 7, {
    animate: true,
    duration: 2,

});
```



13. ábra A térkép főkusza "Kárpátalja" gombra kattintás után

Az összes szimbólum betöltése esetén pedig a következő:

```
document.getElementById("bttosszes").addEventListener("click",
function () {

    map.removeLayer(telep);
    telep.addTo(map);
    map.flyTo([46.475,19.062], 6.4, {
        animate: true,
        duration: 2, // in seconds

    });

});
```

Itt adódik hozzá a térképhez a telep réteg.

Webes összehasonlító névjaji kereső

- © 2022 Tóth András, Témavezető: Dr. Ungvári Zsuzsanna
- Leírás
- ELTE Térképtudományi és Geoinformatikai Tanszék
- 1117 Budapest, Pázmány Péter sétány 1/A

Keresési minta:

Keres!

Magyarország

Horvátország

Erdély

Fehvidék

Órvidék

Délvidék

Kárpátalja

Szerémség

Muravidék

Muraköz

Összes

Visszaállítás

Térképek szerinti névanyag:

1906

1911

1927

1948

1961

2004

JavaSlatok:

Mai államnév

Mai magyarnév

Mai latinbetűs államnév

1961-es nevek

1948-as magyar névalakok

1948-as államnevek

1927-es településnév

Aba	Aba	Aba				
Abádszalók	Abádszalók	Abádszalók		Abádszalók	Abádszalók	
Abaliget						
Abasár	Abasár	Abasár				
Abaujfalvár						
Abaujkér						
Abaujlak						
Abaujszántó	Abaujszántó	Abaujszántó	Abaujszántó	Abaujszántó	Abaujszántó	
Abaujszabolcs						
Abaujvár						
Abda						
Ábelovár						
Abod						

14. ábra Az oldal "Összes" gombra kattintás esetén

8.5. A weboldal végleges elérhetősége

A weboldal offline módon a diplomamunkával leadott dokumentumok közül az *USBWebserver_v8.6* fájl megnyitásával az *usbwebserver.exe* fájl futtatása után a felugró ablakban a *Localhost* gombra kattintva nyitható meg. A weboldal online, bárholnan elérhető verziója a Mercator intézményi szerveren, az én személyes fiókom *public_html* mappájában kapott végleges helyet. Felkerültek az oldalt és a kiegészítő fájlokat tartalmazó fájlok és mappák. Elérhető a <http://mercator.elte.hu/~c4b5xj/diplomamunka/index.php> cím alatt.

9. Összegzés

Dolgozatomban hat magyar térkép a történelmi Magyarország területére eső településneveinek összehasonlításához hoztam létre egy webes keresőt. A feladat teljesítéséhez használtam térinformatikai szoftvert, adatbáziskezelőt, webszervert és több programozási nyelvet. A dolgozat írása közben bemutattam a weboldal elkészítésének és a hozzá szükséges adatok megszerzésének a menetét.

A dolgozat elején bemutattam azt a hat térképet, amiket településnévanyag forrásként használtam, majd rövid áttekintést nyújtottam a magyar településnévadás történetéről. Részletesen kifejtettem a törzskönyvezés lényegét és menetét.

A dolgozat középső részén részletesen leírtam az adatnyerés folyamatát. Ismertettem az adatok összekötésének és az eredmények exportálásának menetét és azok beimportálását az adatbáziskezelőbe.

Készítettem egy felhasználói és egy fejlesztői dokumentációt is: ebben a keresési lehetőségeket és a felhasznált programozási nyelveket és kódrészleteket is megmagyaráztam.

A létrejött adatbázis méretéről szólva a munka kiindulását jelentő adatbázis 16938 települést tartalmazott. Ebből 3204 település található a mai Magyarország területén, 238 Ausztriában, 2810 Szlovákiában, 599 Ukrajnában, 5045 Romániában, 468 Szerbiában, 4406 Horvátországban és 168 település Szlovéniában található. Az 1906-os térképről 279 nevet vettem fel, az 1911-esről 520 nevet, az 1927-esről 232 településnevet, az 1948-asról 635 nevet, az 1961-es térképről 143 nevet és a 2004-es térképről pedig 1048 településnevet tudtam felvenni. A letöltött településekből 15438-hoz egyik térképen se találtam meg, míg 1500 legalább az egyikben előfordult.

Munkám eredményeként létrejött egy felület, ahol lehetőség nyílik áttekinteni, hogy 1906 és 2004 között milyen változásokon ment keresztül a magyar településnévanyag.

Köszönetnyilvánítás:

Ezúton is szeretnék köszönetet mondani a témavezetőmnek, Ungvári Zsuzsannának, aki az adatbázis létrehozása és a weboldal fejlesztése folyamán mindig hasznos és értékes tanácsokkal látott el. Rengeteg segítséget nyújtott a felmerülő problémák megoldásához, ami nagyban hozzájárult diplomamunkám elkészítéséhez.

Irodalomjegyzék

- [1] Térkép fogalma: <https://hu.wikipedia.org/wiki/T%C3%A9rk%C3%A9p>
- [2] Beszkid Arthur, 2017: Webes összehasonlító névjajzi kereső kialakítása glóbuszokhoz. Diplomamunka, ELTE, Budapest, 2017
- [3] Kuris Zoltán, 2017: Webes összehasonlító névjajzi kereső kialakítása glóbuszokhoz. Szakdolgozat, ELTE, Budapest, 2017
- [4] Patkó Máté, 2019: Interaktív térképes földrajzi névmutató készítése, Diplomamunka, ELTE, Budapest, 2019
- [5] Vassányi Gergely, 2022: Pontszerű jelek automatikus felismerése archív térképeken konvolúciós neurális hálózat használatával, Diplomamunka, Budapest, 2022
- [6] Faragó Imre, 2014: Földrajzi nevek, ELTE, Budapest, 2014 (145-148, 165-171.o)
- [7] W3Schools HTML: <https://www.w3schools.com/html/>
- [8] W3Schools CSS: <https://www.w3schools.com/css/>
- [9] W3Schools PHP: <https://www.w3schools.com/php/>
- [10] W3Schools PHP AJAX: https://www.w3schools.com/xml/ajax_intro.asp
- [11] W3Schools PHP JAVASCRIPT: <https://www.w3schools.com/js/default.asp>
- [12] GEOFABRIK: <http://download.geofabrik.de/>
- [13] Dr. Brózik Károly: Nagy Magyar Atlasz: Ausztria és Magyarország (1906)
- [14] Kogutowicz Manó: Teljes Atlasz: Magyarország politikai térképe (1911)
- [15] Cholnoky: Földrajzi és Statisztikai Atlasz: Magyarország (1927)
- [16] Bibliotheca Világatlasz: Magyarország, Ausztria és a Felső-Adria vidéke, Csehszlovákia, Szlovákia fürdő és bányavidéke, Jugoszlávia, Románia, A Szovjetunió nyugati vidékei (1949)
- [17] Politikai és Gazdasági Világatlasz: Magyarország (1961)
- [18] Világatlasz Topográf Térképészeti Kft. A Kárpát-medence közigazgatása (2004)
- [19] OCR karakterfelismerés: <https://tools.pdf24.org/hu/>
- [20] USBWebserver letöltési helye: <http://www.usbwebserver.net/en/>
- [21] JavaScript leírás: <https://gremmedia.hu/javascript-nyelv-alapjai-elo-peldakkal-bemutatva>
- [22] Leaflet: <https://leafletjs.com/index.html>
- [23] KSH településváltozások: https://www.ksh.hu/teruletiatlasz_telepulesek
- [24] W3Schools CSS Buttons: https://www.w3schools.com/css/css3_buttons.asp
- [25] W3Schools CSS:nth-child() Selector: https://www.w3schools.com/cssref/sel_nth-child.asp

Honlapok utolsó látogatása: 2022.05.14.

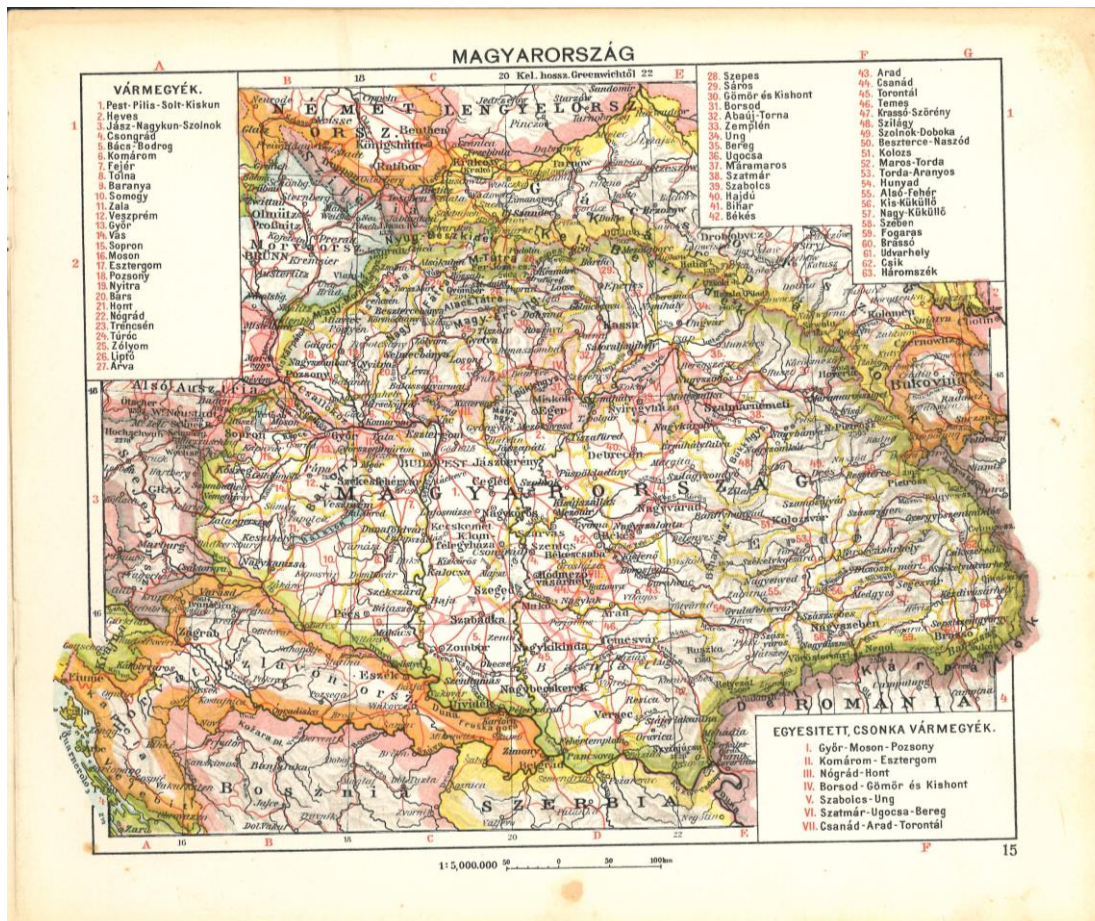
Mellékletek



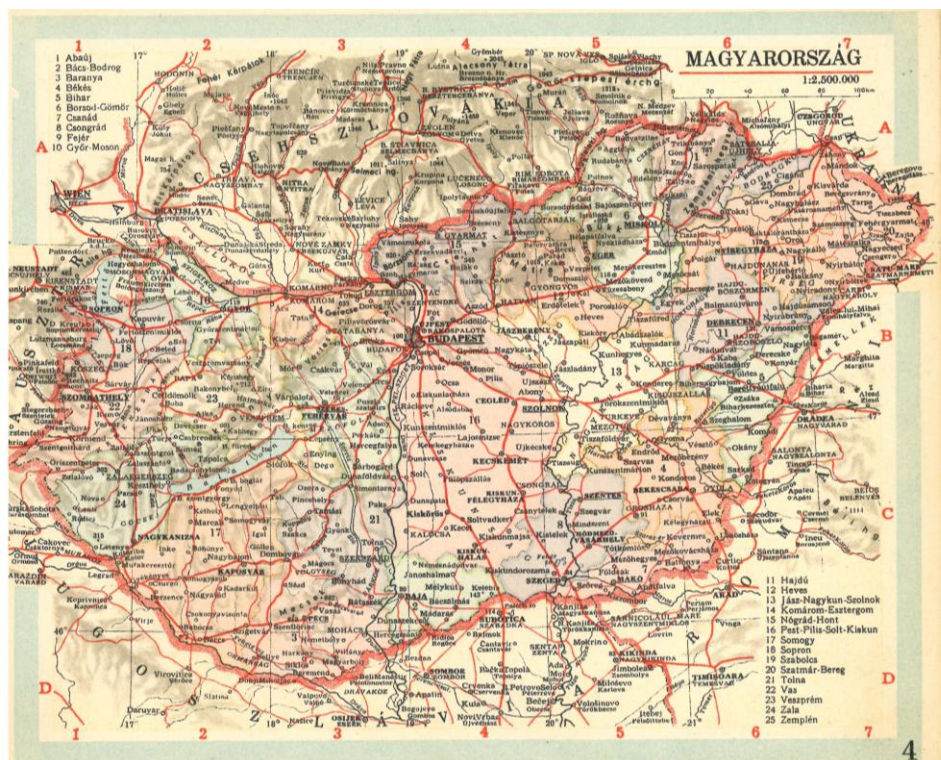
1. Melléklet: Nagy Magyar Atlasz: Ausztria és Magyarország (1906)



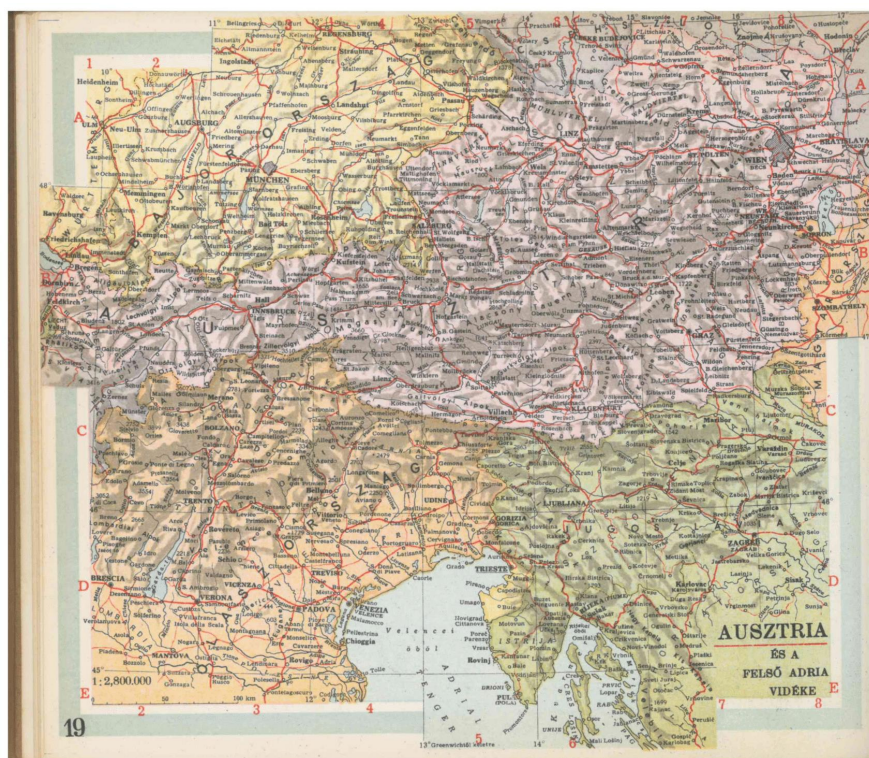
2. Melléklet: Teljes Atlasz: Magyarország politikai térképe (1911)



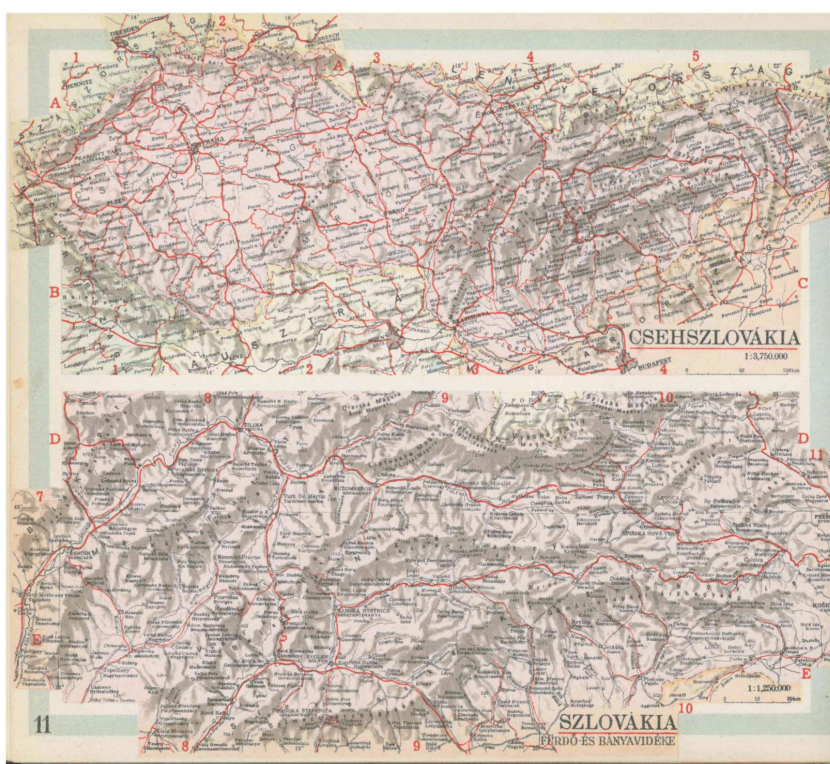
3. Melléklet: Chohnoky: Földrajzi és Statisztikai Atlasz: Magyarország (1927)



4. Melléklet: Bibliotheca Világatlasz: Magyarország (1949)



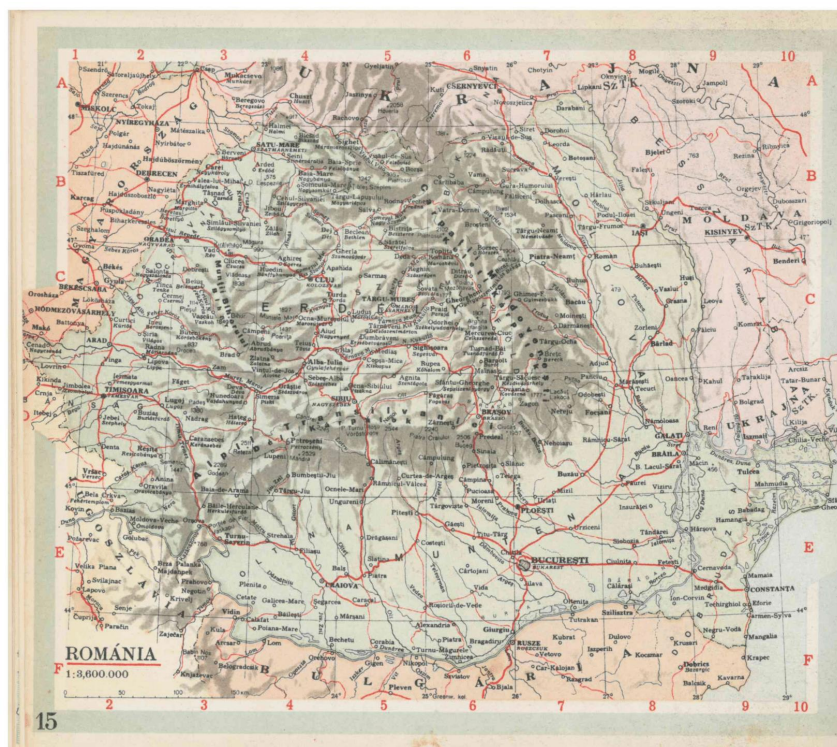
5. Melléklet: Bibliotheca Világatlasz: Ausztria és a Felső-Adria vidéke (1949)



6. Melléklet: Bibliotheca Világatlasz: Csehszlovákia, Szlovákia fürdő és bányavidéke(1949)



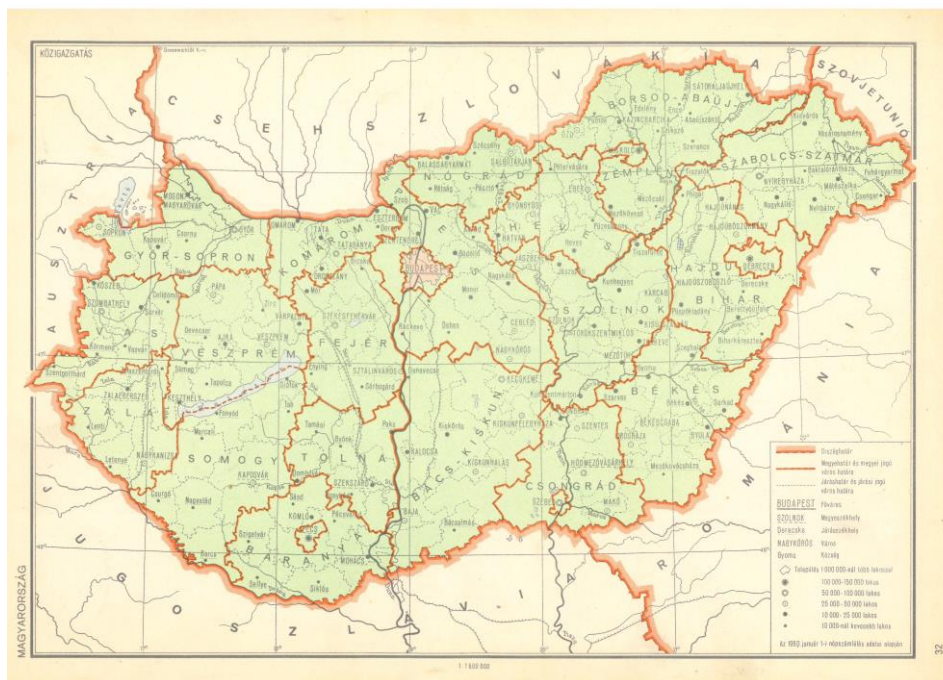
7. Melléklet: Bibliotheca Világatlasz: Jugoszlávia (1949)



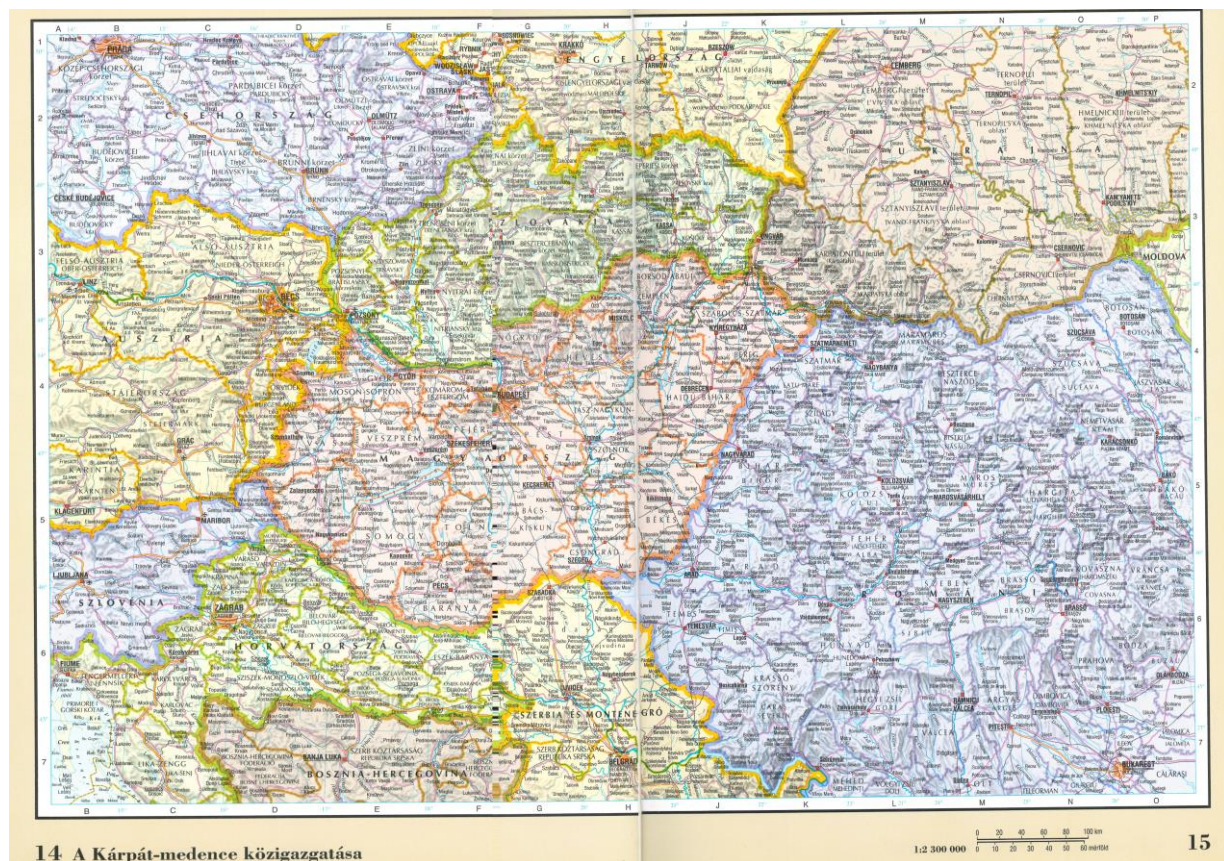
8. Melléklet: Bibliotheca Világatlasz: Románia (1949)



9. Melléklet: Bibliotheca Világatlasz: A Szovjetunió nyugati vidékei (1949)



10. Melléklet: Politikai és Gazdasági Világatlasz: Magyarország (1961)



11. Melléklet: Világatlasz Topográf Térképészeti Kft. A Kárpát-medence közigazgatása (2004)

**DIPLOMAMUNKA LEADÁSI
és
EREDETISÉG NYILATKOZAT**

Alulírott TÓTH ANDRÁS.....Neptun-kód: C4B5X3.....

az Eötvös Loránd Tudományegyetem Informatikai Karának, Térképtudományi és
Geoinformatikai Tanszékén

WEBES ÖSSZEHASZNÁLÓ NEVRAJZI KERESŐ KIALAKÍTÁSA
TÉRKÉPEKHEZ.....

című diplomamunkámat a mai napon leadtam.

Témavezetőm neve: DR. UNGVÁRI ZSUZSANNA.....

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

Büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom, hogy jelen
szakdolgozatom/diplomamunkám saját, önálló szellemi termékem; az abban hivatkozott
szakirodalom felhasználása a szerzői jogok általános szabályainak megfelelően történt.
Tudomásul veszem, hogy szakdolgozat/diplomamunka esetén plágiumnak számít:

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

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

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2022. május 15.

Tóth András.....
hallgató aláírása