



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet



SZAKDOLGOZAT

A második világháborús magyarországi bombázások megjelenítése webes felületen

OE-AMK

2022

Hallgató neve: Bertli Tibor Attila

Hallgató törzskönyvi száma: T000654/FI12904/A-G



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Geoinformatikai Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Bertli Tibor Attila
Szakdolgozat száma: SZD22011414065322
Törzskönyvi száma: T000654/FI12904/A-G

Neptun kódja: XXXXXXXXXX

Szak: geoinformatikai szakember
Specializáció:

A dolgozat címe: A második világháborús magyarországi bombázások megjelenítése webes felületen

A dolgozat címe angolul: Web map for WWII bombings in Hungary

A feladat részletezése:

Tervezzon térinformatikai adatbázist a második világháborús magyarországi bombázások adatainak tárolására. Hozzon létre és töltsön fel (minta)adatokkal egy ilyen adatbázist. Készítsen webes felületű megjelenítő felületet az adatbázishoz. Mutassa be dolgozatában részletesen az elvégzett munkát.

Intézményi konzulens neve: Nagy Gábor József

A kiadott téma elővívési határideje: 2024. 12. 15.

Beadási határidő: 2022. 05. 15.

A szakdolgozat: Nem titkos.

Kiadva: Székesfehérvár, 2022. 04. 07. P.H.

Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat/diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozat/diplomamunkában található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: Budapest, 2022 december. 14.

.....
hallgató aláírása

Tartalomjegyzék

1.	Bevezetés	1
2.	Térinformatika a történettudományban	4
3.	Adatgyűjtés	7
3.1	Szükséges adatok meghatározása.....	9
3.1.1	Magyarország határa 1944-ben.....	9
3.1.2	Budapest belterületének határa 1944-ben	9
3.1.3	A 15. US Air Force fellelhető adatai	9
3.1.4	Az adatbázishoz kapcsolható egyéb források	10
3.2	Adatfeldolgozás.....	10
3.3	Az adatfeldolgozás folyamata	12
3.3.1	Adatbázis tervezés	12
3.3.2	Adattáblák.....	12
3.3.3	Geometria táblák.....	17
4.	Alkalmazásfejlesztés.....	23
4.1	Alkalmazás tervezés.....	23
4.2	Alkalmazás fejlesztés	24
4.2.1	Python	25
4.2.2	Django [30]	26
4.2.3	Leaflet JS	40
4.3	Az alkalmazás működése	59
4.4	Tapasztalatok és a továbbfejlesztés lehetőségei.....	62
5.	Összefoglaló.....	64
6.	Summary.....	66

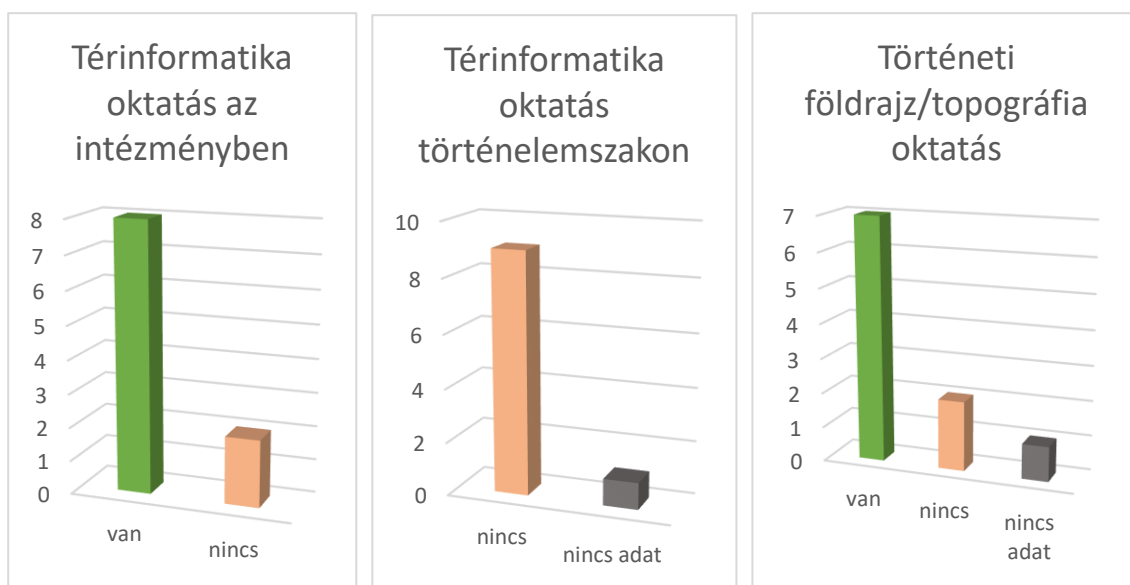
Irodalomjegyzék	68
7. Ábrajegyzék	74

1. BEVEZETÉS

A történettudományi kutatómunka során gyakran szembesülhetünk a térbeliség problematikájával. „A 20. században a történettudománynak több olyan részterülete is kialakult, amely a térbeliség vizsgálatát kiemelten kezeli. Ilyen például a történelem- és földrajztudomány közé ékelődő történeti földrajz.” [1, p. 103]

Történelem szakos tanulmányaimat a Miskolci Egyetemen végeztem 1999 – 2004 között. Mivel egy műszaki egyetemről beszélünk, ahol a Műszaki Földtudományi Karon már oktatnak térinformatikát, a történelem szakon csak elméleti síkon, történeti földrajzot hallgattunk, e mellett informatikai ismereteket is tanultunk.

Ennek apropóján átnéztem a magyarországi történész képzések tantárgyleírásait (1. ábra).



1. ábra Térinformatikai/topográfiai/történeti földrajzi tárgyak történelem szakon

Intézmény ¹	Térinformatika oktatás az intézményben	Térinformatika oktatás történelemszakon	Történeti földrajz/topográfia oktatás
DE	van	nincs	van
EKE	van	nincs	van
ELTE	van	nincs	van
KJE	nincs	nincs	nincs
KRE	nincs	nincs	nincs
ME	van	nincs	van
NYE	van	nincs	van
PPKE	van	nincs	van
PTE	van	nincs	van
SZTE	van	nincs adat	nincs adat

A fenti összehasonlításból kiderül, a tíz Intézményből nyolc helyen van térinformatika oktatás, történelem szakon kilenc Egyetemen biztosan nem oktatnak Térinformatika tárgyat, de a Történeti földrajz/topográfia oktatása sem minden intézményben valósul meg a Történelem szakon.

Érdekesség, hogy a PPKE Régészettudományi Intézetében működik Régészeti Térinformatikai Laboratórium. [2]

A diploma megszerzése után sok évvel hallottam először térinformatikáról, amikor egy átképzésen vettem részt. Ez egy OKJ rendszerű tanfolyam volt, ahol a nyíregyházi Jósa

¹ Debreceni Egyetem, Eszterházy Károly Egyetem, Eötvös Loránd Tudományegyetem, Kodolányi János Egyetem, Károli Gáspár Református Egyetem, Miskolci Egyetem, Nyíregyházi Egyetem, Pázmány Péter Katolikus Egyetem, Pécsi Tudomány-egyetem, Szegedi Tudományegyetem.

András Múzeum térinformatikusa oktatta a Geodéziai és térinformatikai ismereteket. Mivel a térképek használata elkerülhetetlen a történettudományban, azonnal elkezdett érdekelni, hogyan lehetne hasznosítani térinformatikai ismereteimet a történeti kutatásban, adatfeldolgozásban, elemzésben és megjelenítésben.

Egy történelmi eseménysort szeretnék ábrázolni, amelyben számos kronológiai és térbeli problémán keresztül reprezentálhatom a Geoinformatika eszközei miként támogatják a történettudományi kutatást.

2. TÉRINFORMATIKA A TÖRTÉNETTUDOMÁNYBAN

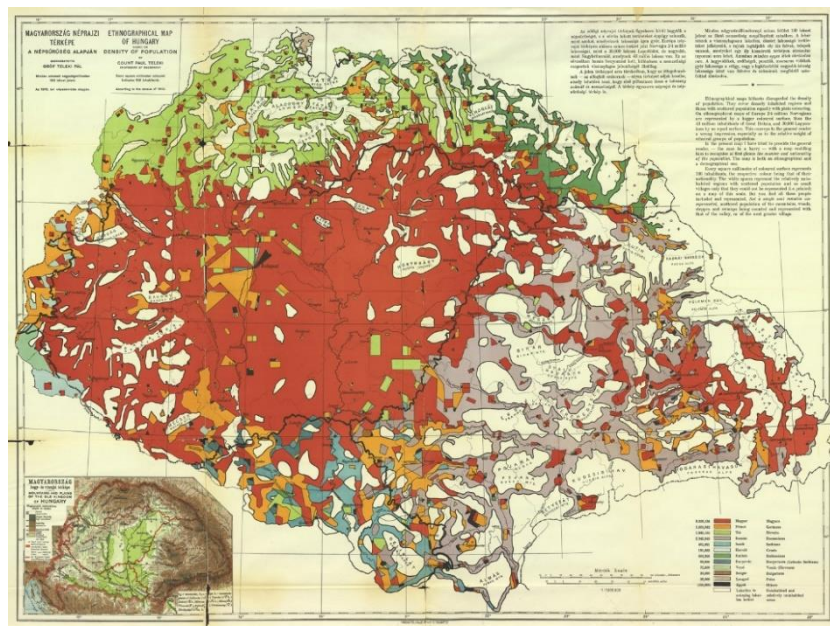
Kétféle felhasználási módja van a térképeknek a történettudományi kutatómunkában.

Az első, amikor a térképet valamilyen eseménysor szemléltetésére használjuk, ebben az esetben a felhasználás egy kimenete a történeti tevékenységnek. Ilyen például egy hadjárat ábrázolása, de ide sorolható egy adott területen élő népesség etnikai, felekezeti és nemzetiségi összetételét ábrázoló térkép korabeli összeírások alapján.

„A térkép „output” felhasználási esete tehát nem azt jelenti, hogy ez a történeti munka legutolsó fázisa – hiszen annak elemzése és a megfelelő következtetések levonása még a kutató feladata –, hanem arra utal, hogy a térkép a kutatómunka során a történész által keletkezik.” [1, p. 104]

A másik felhasználási terület amikor a térképet forrásként használják a történészek, kutatómunkájuk során.

„A térképek – alapvető tulajdonságaik figyelembevételével mellett – igen értékes források, hiszen a korabeli viszonyokat visszatükröző „fényképeknek” tekinthetők, vagyis objektív információkat, kvantitatív jellegű adatokat szolgáltatnak.” [1, p. 104]



2. ábra Magyarország néprajzi térképe népsűrűség alapján [https://www.ogyk.hu/uploads/ogyk/tartalom/mo-neprajzi-terkepe-hu.jpg]

Példaként említhetők a katonai felmérések térképei, vagy a gróf Teleky Pál irányítása alatt 1919-re elkészült „Magyarország néprajzi térképe népsűrűség alapján”, ami „Vörös térkép” [3] néven vált ismertté (2. ábra).

„A digitális térképészet, valamint a térinformatikai rendszerek megjelenése az „input” és az „output” felhasználási mód esetében is komoly áttörést jelent. Az adatok feldolgozása egyszerűbbé és gyorsabbá vált. Olyan kutatási részfeladatok, amelyek korábban több személy igen hosszan tartó munkáját igényelték, egy – az új eljárásokat ismerő és alkalmazni tudó – személy által jóval rövidebb idő alatt is elvégezhetők lettek.” [1, pp. 104 - 105]

Jelentős mérföldköve volt a történeti térinformatikának („GISTória”² [4]) a GISa Hungarorum [5] létrejötte, egy történeti térinformatikai rendszer, amely Magyarország és az Osztrák-Magyar Monarchia történetének tanulmányozását segíti, az 1857 – 1910 közötti időszakban. Az elkészült rendszer az OTKA K 111766 projekt keretében jöhetett létre. A projektben résztvevő intézmények között szerepel a Debreceni Egyetem, az Eötvös Lo-

² A kifejezést Demeter Gábor használta [4]

ránd Tudományegyetem, a Pécsi Tudományegyetem és a Nyíregyházi Egyetem, mely intézmények képzési tematikájában mégsem jelenik meg a térinformatika tantárgy a történelem hallgatók részére.

A megvalósítandó cél egy webalkalmazás, a II. világháború magyarországi hadműveleteit kutató történészek munkájának segítéséhez és egy tájékoztató felület létrehozása, a téma iránt érdeklődőknek, illetve gyakorlati példán keresztül, annak szemléltetése, mennyiben támogatják a térinformatikai megoldások a történettudományi kutatómunkát, illetve a publikációt.

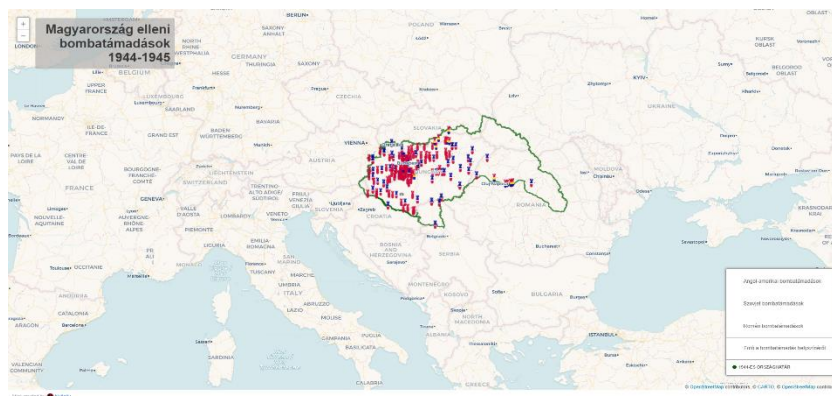
Elsőként az térinformatikai adatbázist kell felépíteni. Meg kell határozni milyen adatokat szeretnék megjeleníteni, azokat milyen részletességben ábrázolom.

A térinformatikai megoldás esetén vektoros megjelenítést alkalmazok. „A vektoros GIS a hagyományos térképi elemzésből alakult ki a hetvenes években. A megoldás nagyon hasonlít az ember hagyományos szemléletéhez. Ebben az esetben a valós világot pontok, vonalak és foltok sorozatával és a hozzájuk kapcsolódó leíró adatokkal jellemezzük. Az elemzések során ezeknek a kapcsolatát vizsgáljuk, vagy egymásra vetítésükkel az azonos helyeken valamilyen feltételek teljesülését határozzuk meg.” [6, p. 9]

Az adatbázisba betöltjük az alap adatokat, országhatár, településhatár elemosztályait. El kell dönteni, hogy az adatokhoz tartozó attribútumokat közvetlenül az adattáblában, vagy kapcsolható segédtáblákban tároljuk és milyen egyéb adatot szeretnénk kapcsolni az adatainkhoz?

3. ADATGYŰJTÉS

Az adatgyűjtés során az Interneten fellelhető adatokra szorítkoztam. A kutatás során derült ki számomra, hogy a Nemzeti Emlékezet Bizottsága is készített hasonló interaktív térképet. [7] (3. ábra)

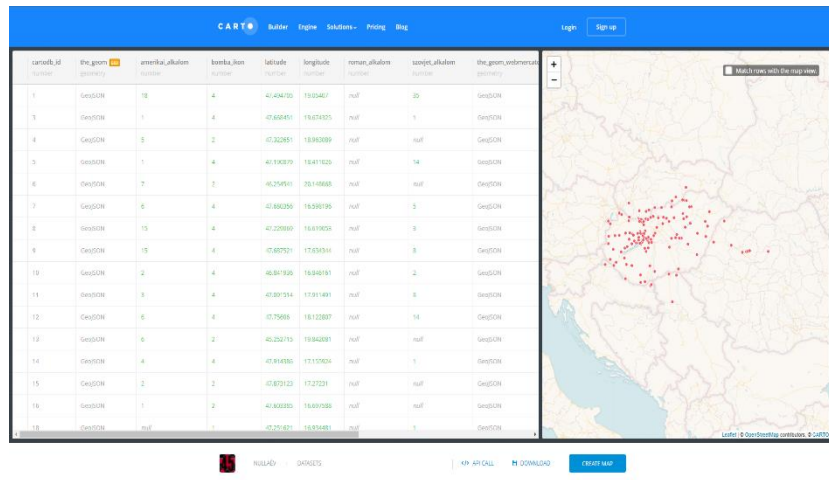


3. ábra A Nemzeti Emlékezet Bizottsága térképének nyitólapja [https://marvinn.carto.com/viz/f44798e8-183e-11e5-8537-0e018d66dc29/embed_map#zoom_in]

Az itt készült térkép az összes Magyarország ellen intézett bombatámadás helyszínét igyekszik bemutatni, hasonló elven, mint ahogy az általam készített interaktív térkép is.

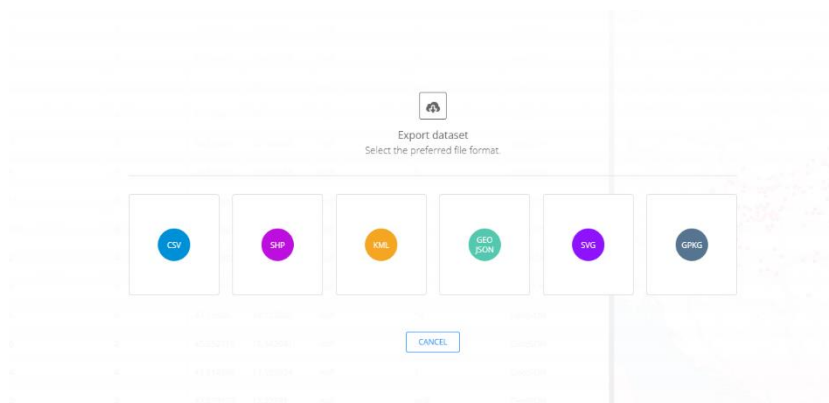
A két térkép közötti különbséget az elkészítés módja adja. A NEB egy CARTO [8] nevű online webalkalmazás készítő szoftvert használva hozta létre a térképet. Az adatbázist GeoJSON formátumban tárolja, ahonnan azok az adatok néhány kattintással szabadon letölthetőek.

A CARTO oldalán a térkép adatbázisa böngészhető (4. ábra), a DOWNLOAD gombra kattintva különböző formátumokban tárolhatók. (5. ábra)



cartodb_id	the_geom	area	area	area	area	area	area	area	area	area
1	GeoJSON	18	4	47.424770	19.024407	na	20	GeoJSON		
2	GeoJSON	1	4	47.404441	19.074331	na	1	GeoJSON		
3	GeoJSON	3	2	47.322851	18.943889	na	na	GeoJSON		
4	GeoJSON	1	4	46.943119	18.417024	na	14	GeoJSON		
5	GeoJSON	7	2	46.254541	18.148845	na	na	GeoJSON		
6	GeoJSON	6	4	47.880250	18.580190	na	5	GeoJSON		
7	GeoJSON	15	4	47.220840	18.418054	na	9	GeoJSON		
8	GeoJSON	15	4	47.487721	17.934314	na	9	GeoJSON		
9	GeoJSON	2	4	46.947826	18.942181	na	2	GeoJSON		
10	GeoJSON	3	4	47.891314	17.911491	na	5	GeoJSON		
11	GeoJSON	6	4	47.750858	18.122807	na	14	GeoJSON		
12	GeoJSON	6	2	46.252170	18.942091	na	na	GeoJSON		
13	GeoJSON	4	4	47.814385	17.137020	na	1	GeoJSON		
14	GeoJSON	2	2	47.870122	17.27221	na	na	GeoJSON		
15	GeoJSON	1	2	47.888385	18.891988	na	na	GeoJSON		
16	GeoJSON	na	1	47.221021	18.934481	na	1	GeoJSON		

4. ábra A CARTO oldalán a térkép adatbázisa [https://marvinn.carto.com/tables/bombatamadasok_uj/public]



5. ábra Adatbázis exportálás a CARTO felületéről [https://marvinn.carto.com/tables/bombatamadasok_uj/public]

3.1 Szükséges adatok meghatározása

3.1.1 Magyarország határa 1944-ben

Az 1944-es országhatárt az 1938-as és a 1940-es Bécsi döntések, illetve a 1941 es Délvidéki hadművelet során visszacsatolt területek alapján lehet meghúzni.

A feladat nehézségét az adta, hogy pontos leírást nem sikerült fellelnem a bécsi döntésekhez kapcsolódóan. Korabeli közigazgatási térképek, alapján meghatározható volt az országhatár.

Az országhatárhoz szükséges alapadat forrásai:

- GISa Hungarorum [5]
 - MO_Telepules.shp [9]
- Hungaricana [10]
 - Magyarország közigazgatási térképe 1918-ban és 1944-ben. [11]

3.1.2 Budapest belterületének határa 1944-ben

Itt hasonlóan, mint az országhatár elkészítésénél, korabeli településtérképet vettem alapul:

- Arcanum [12]
 - Budapest (1945) - Budapest közigazgatási térképsorozata [13]

3.1.3 A 15. US Air Force fellelhető adatai

A rendelkezésre álló történeti tanulmányokból kiderül, hogy a Magyarország elleni bombatámadásokban az Egyesült Államok 15. Légierője, az angol Királyi Légierő 205. Bombázó csoportja, valamint román és szovjet bombázók vettek részt.

Sajnos a RAF, a szovjet és a román bombázók forrásokkal alátámasztott tevékenységéről online információt nem találtam, így ezek további kutatást igényelnek.

Az Egyesült Államok Légierője nagyon jól dokumentált weboldalakgal rendelkezik a különböző háborúkban részt vett alakulatokról, így elsősorban erre koncentráltam.

A 15. Légió 2. világháborús tevékenységeinek kutatásához jelentős forrásanyag áll rendelkezésre. [14] A térkép alapját ez az adatbázis képezi. Ebben megtalálhatóak, hogyan épült fel a légió, milyen ezredek, csoportokból és századokból állt össze a légió. Az egyes alakulatok oldalain megtalálhatók a bevetések dokumentációi, a bevetési parancsokból kiderül, mikor milyen célpontokat, mekkora erővel támadtak, a bevetési jelentések pedig a küldetés sikerességéről, valamint a veszteségekről is beszámol.

3.1.4 Az adatbázishoz kapcsolható egyéb források

Nagyon hasznos forrás a Fortepan [15] adatbázisa, ahonnan fényképeket csatolhatunk az adatainkhoz.

Hasznos források a korabeli napilapok, helyi lapok, térképek melyek többek között a Hungaricana adatbázisában kutathatók. [10]

Az oldal működésének bemutatásához ennyi információt is elegendőnek tartok.

3.2 Adatfeldolgozás

A feladat megoldása során törekedtem a Free and Open-Source Software (FOSS) eszközök adatai lehetőségei kihasználására.

Az Open Source, vagy „nyílt forráskódú szoftverek szabadon használható, másolható, terjeszthető, tanulmányozható és módosítható programok.” [16]

Ezek a szoftverek ingyen használható, sokoldalú és biztonságos megoldást nyújtanak a felhasználóknak, a nyílt forráskód miatt gyorsabban fejleszthető és egy probléma esetén a közösségi jellege miatt gyors megoldást biztosítanak.

Az alábbi szoftvereket és eszközöket használtam a Web alkalmazás elkészítése során:

- QGIS [5] : „A QGIS egy felhasználóbarát, nyílt forráskódú térinformatikai rendszer (GIS), amely GNU Általános nyilvános licenc alapján használható. A QGIS az Open Source Geospatial Foundation (OSGeo) hivatalos projektje. Futtatható Linux, Unix, Mac OSX, Windows és Android környezetben, és számos vektoros, raszteres és adatbázis formátumot és funkciót támogat.” [17]

- PostgreSQL [18]: A PostgreSQL egy nyílt forráskódú SQL nyelvet használó objektum-relációs adatbázis kezelő rendszer. Ehhez telepítettem a PostGIS és PostGIS_topology bővítményeket a geometriai adatok kezeléséhez.
- pgAdmin 4 [19]: A PostgreSQL adatbáziskezelő rendszer grafikus kezelőfelülete.
- Visual Studio Code [20]: Asztali forráskód szerkesztő alkalmazás, amely több programozási nyelvet támogat.
- Python 3. 10 [21]: A Python egy értelmezett objektum-orientált magasszintű programozási nyelv, mely többféle programozási paradigmát támogat. Az értelmezett azt jelenti, hogy a program egy köztes fordító által, futás közben fordul számítógép által értelmezhetővé.
- Django [22]: A Django egy ingyenes, nyílt forráskódú Python alapú keretrendszer, amely Model-View-Template (MVT) szoftver tervezési mintán alapul. A Model segít az adatbázis elérésben, a Template egy megjelenítési réteg, amely kezeli a felhasználói felületet, a View hozza létre a kapcsolatot a Model és a Template között.
- Leaflet JS [23]: A Leaflet egy nyílt forráskódú Javascript könyvtár, interaktív térképekhez.
- GeoJSON [24]: A GeoJSON különböző geometriai adatok kódolására szolgáló formátum. Feature (Pont, Vonallánc, Poligon), valamint FeatureCollection (Multipont, Multi Vonallánc és Multi Poligon) kódolására alkalmas.

3.3 Az adatfeldolgozás folyamata

3.3.1 Adatbázis tervezés

A rendelkezésre álló információkból meg kellett határozni milyen adatokat fogok felhasználni.

Elsőként leképeztem a 15. Légierő alakulatait Osztály (Group) szintig. Az Osztály, egy az Amerikai Egyesült Államok hadseregének egy egysége, aminek magyarországi megfelelője nincs, „általában három-négy századból álló repülő alakulat [25]”.

3.3.2 Adattáblák

Airforce – a Légierő Ezred szintű táblája [26]

A 15. Légerő felépítése

- Bombázó ezredek
 - 5. Bombázó ezred
 - 47. Bombázó ezred
 - 49. Bombázó ezred
 - 55. Bombázó ezred
 - 304. Bombázó ezred
- XV. Vadászrepülő parancsnokság
 - 305. Vadászrepülő ezred
 - 306. Vadászrepülő ezred

A fenti adatokból az következő SQL paranccsal létrehoztam a Légierő tábláját.

```
CREATE TABLE IF NOT EXISTS public.airforce
```

```
(
```

```
  id integer NOT NULL DEFAULT nextval('airforce_id_seq'::regclass),
```

```
  objectid bigint,
```

```
-- Az ezred egyedi azonosítója
```

```
  wing_id character varying(255) COLLATE pg_catalog."default",
```

```

-- Az ezred neve
    name character varying(255) COLLATE pg_catalog."default",
-- Az ezredparancsnokság helye
    headq character varying(255) COLLATE pg_catalog."default",
-- Ezredparancsok neve
    commander character varying(255) COLLATE pg_catalog."default",
-- Az ezred weboldalának linkje
    wing_link character varying(255) COLLATE pg_catalog."default",
    CONSTRAINT airforce_pkey PRIMARY KEY (id)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS public.airforce

```

```

    OWNER to postgres;

```

Group – a Légierő Osztály szintű táblája [26]

Az egyes ezredek felépítése:

- 5. Bombázó ezred
 - 2. Bombázó csoport
 - 97. Bombázó csoport
 - 99. Bombázó csoport
 - 301. Bombázó csoport
 - 463. Bombázó csoport
 - 483. Bombázó csoport
- 47. Bombázó ezred
 - 98. Bombázó csoport
 - 376. Bombázó csoport
 - 449. Bombázó csoport
 - 450. Bombázó csoport
- 49. Bombázó ezred
 - 451. Bombázó csoport
 - 461. Bombázó csoport
 - 484. Bombázó csoport
- 55. Bombázó ezred
 - 460. Bombázó csoport
 - 464. Bombázó csoport
 - 465. Bombázó csoport
 - 485. Bombázó csoport
- 304. Bombázó ezred
 - 454. Bombázó csoport
 - 455. Bombázó csoport
 - 456. Bombázó csoport
 - 459. Bombázó csoport
- 305. Vadászrepülő ezred

- 1. Vadászrepülő csoport
- 14. Vadászrepülő csoport
- 82. Vadászrepülő csoport
- 306. Vadászrepülő ezred
 - 31. Vadászrepülő csoport
 - 52. Vadászrepülő csoport
 - 325. Vadászrepülő csoport
 - 332. Vadászrepülő csoport

CREATE TABLE IF NOT EXISTS public."group"

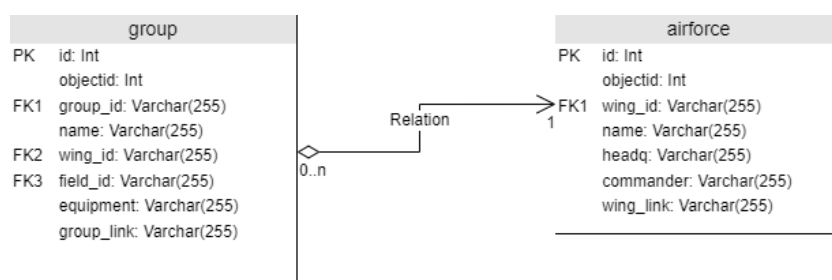
```
(
  id integer NOT NULL DEFAULT nextval('group_id_seq'::regclass),
  objectid bigint,
  -- Az csoport egyedi azonosítója
  group_id character varying(255) COLLATE pg_catalog."default",
  -- A csoport neve
  name character varying(255) COLLATE pg_catalog."default",
  -- Az ezred egyedi azonosítója
  wing_id character varying(255) COLLATE pg_catalog."default",
  -- A légibázis egyedi azonosítója
  field_id character varying(255) COLLATE pg_catalog."default",
  -- A csoport által repült repülőgép típusa
  equipment character varying(255) COLLATE pg_catalog."default",
  -- A csoport weboldalának linkje
  group_link character varying(255) COLLATE pg_catalog."default",
  CONSTRAINT group_pkey PRIMARY KEY (id)
```

)

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS public."group"
```

```
OWNER to postgres;
```



6. ábra Az *airforce* és a *group* táblák

A két táblába a pgAdmin segítségével betöltöttem az adatokat a korábban létrehozott .csv fájlokból. A táblák kapcsoló mezője a **wing_id**. (6. ábra)

A két tábla relációjából létrehoztam az **airforcegroup** táblát (7. ábra) a következő SQL paranccsal:

```
CREATE TABLE airforcegroup AS
```

```
SELECT public.group.group_id, public.group.name AS grname, public.group.equipment, group_link AS grlink, airforce.name AS afname, airforce.headq, airforce.commander, airforce.wing_link AS wlink, public.group.field_id
```

```
FROM public.airforce
```

```
INNER JOIN public.group ON public.group.wing_id = public.airforce.wing_id
```

ORDER BY public.group.group_id;

airforcegroup	
FK1	group_id: Varchar(255)
	grname: Varchar(255)
	equipment: Varchar(255)
	grlink: Varchar(255)
	afname: Varchar(255)
	headq: Varchar(255)
	commander: Varchar(255)
	wlink: Varchar(255)
FK2	field_id: Varchar(255)

7. ábra Az *airforcegroup* tábla

3.3.3 Geometria táblák

Airfields – A 15. Légerő bázisai Olaszországban (8. ábra).

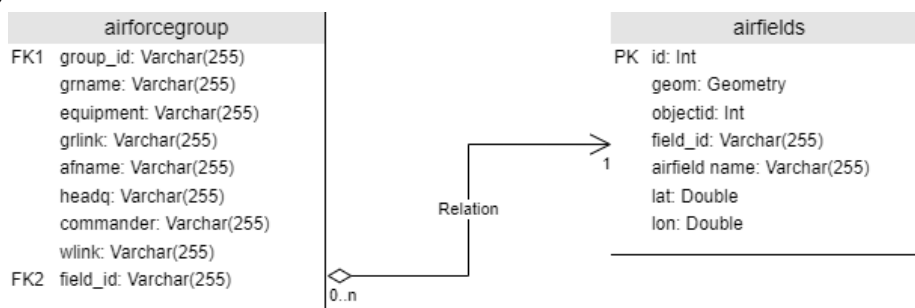
Ehhez az adatot a Wikiwand [27] oldalon találtam meg a repülőterek koordinátaival, amiből előállítottam egy .csv fájlt.

A QGIS-ben a .csv-ből pont réteget generáltam, majd ezt a DB Manager eszközzel feltöltöttem a PostgreSQL adatbázisba. A Target SRID-t beállítottam EPSG:4326, WGS84 vetületi rendszerbe.

airfields	
PK	id: Int
	geom: Geometry
	objectid: Int
	field_id: Varchar(255)
	airfield name: Varchar(255)
	lat: Double
	lon: Double

8. ábra *airfields* tábla

Az *airfields* tábla és az *airofrocegroup* tábla a ***field_id*** nevűvel kapcsolható össze (9. ábra).



9. ábra Az *airfields* és az *airforcegroup* táblák kapcsolata

A két tábla összekapcsolásával elkészítettem az **airfieldsag** táblát (10. ábra) az alábbi SQL parancsot futtatva:

```
CREATE TABLE airfieldsag AS
```

```
SELECT public.airfields.geom, public.airfields.airfield_name AS afdname, public.airforcegroup.group_id, public.airforcegroup.grname, public.airforcegroup.equipment, public.airforcegroup.grlink, public.airforcegroup.afname, public.airforcegroup.headq, public.airforcegroup.commander, public.airforcegroup.wlink
```

```
FROM public.airfields
```

```
INNER JOIN public.airforcegroup ON public.airforcegroup.field_id = public.airfields.field_id
```

```
ORDER BY public.airforcegroup.group_id;
```

Az elkészült táblához hozzáadtam egy elsődleges kulcs mezőt:

```
ALTER TABLE airfieldsag ADD COLUMN id SERIAL PRIMARY KEY;
```

airfieldsag	
	geom: Geometry
	afdname: Varchar(255)
FK1	group_id: Varchar(255)
	grname: Varchar(255)
	equipment: Varchar(255)
	grlink: Varchar(255)
	afname: Varchar(255)
	headq: Varchar(255)
	commander: Varchar(255)
	wlink: Varchar(255)
PK	id: Int

10. ábra Az **airfieldsag** tábla

Az **airfieldsag** tábla mezői:

- geom: A geometriai adatokat tartalmazza
- afdname: A légibázis neve
- group_id: A légibázison állomásozó Osztály azonosítója
- grname: A légibázison állomásozó Osztály neve
- equipment: Az Osztály által repült repülőgép típusa
- grlink: Az Osztály weboldalának linkje

- **afname:** Az Ezred neve, ahova az Osztály tartozik
- **headq:** Az Ezredparancsnokság helye
- **commander:** Az Ezred parancsnoka
- **wlink:** Az Ezred weboldalának linkje

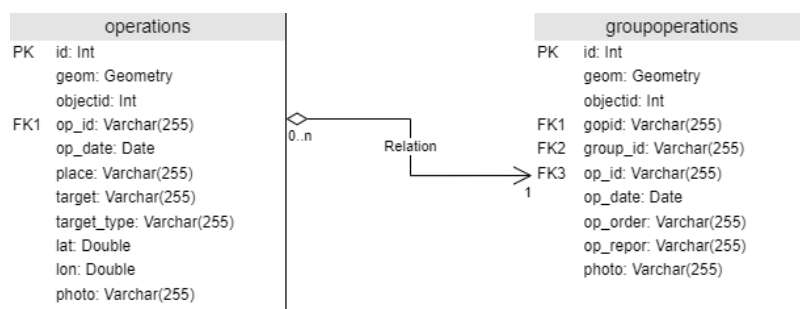
Operations – Bevetések

A bevetések adatbázisát a 15. Légierő honlapján található bevetési parancsokból és a „Légi háború Magyarország felett” [28] című könyvből gyűjtöttem ki.

1944 április és 1944 december között 93 alkalommal hajtottak végre bombatámadást magyarországi célpontok ellen. Elsősorban az ipari és vasúti infrastruktúrák, valamint a repülőterek estek áldozatul a bombázásoknak. Sok esetben, mivel az ipari létesítmények, nagyvárosokban találhatók, a közelben lévő civil épületek is áldozatul estek a támadásoknak. Jelen munkában a bevetések helyszíneire, a tervezett célpontok elleni bevetésekre koncentráltam. További kutatást igényel a bombatalálatok térképi ábrázolása, a bombatámadások pontossága.

A támadások helyszínének pontjait manuálisan, műholdkép alapján raktam le. Az elkészült pont réteget feltöltöttem a PostgreSQL adatbázisba, szintén EPSG:4326, WGS84 vetületi rendszerben.

Az **operations** tábla a célpontokat tartalmazza, készítettem egy **groupoperations** nevű kapcsolótáblát is, amiben az Osztályok azonosítójához rendeltem a küldetéseket. (10. ábra)



11. ábra Az **operations** és **groupoperations** táblák kapcsolata

A két tábla összekapcsolásával létrejött a **bombsites** tábla:

CREATE TABLE bombsites AS

SELECT public.operations.geom, public.operations.op_date AS opdate, public.groupoperations.group_id, public.groupoperations.op_order AS oporder, public.operations.place, public.operations.target, public.operations.target_type AS targettype, public.groupoperations.op_report AS opreport, public.groupoperations.photo AS opphoto, public.operations.photo AS targetphoto

FROM public.operations

INNER JOIN public.groupoperations ON public.operations.op_id = public.groupoperations.op_id

ORDER BY public.operations.op_date

bombsites	
	geom: Geometry
	opdate: Date
FK1	group_id: Varchar(255)
	oporder: Varchar(255)
	place: Varchar(255)
	target: Varchar(255)
	targettype: Varchar(255)
	opreport: Varchar(255)
	opphoto: Varchar(255)
	targetphoto: Varchar(255)
PK	id: Int
	targtypeen: Varchar(255)
	grname: Varchar(255)

12. ábra A bombsites tábla

Az **bombsites** tábla mezői:

- geom: A geometriai adatokat tartalmazza
- opdate: A bombatámadás dátuma (Év-Hónap-Nap)
- group_id: A bombatámadásban részvevő Osztály egyedi azonosítója
- oporder: Bevetési parancs (web link)
- place: A bombatámadás helye (Településnév)
- target: A bombatámadás célpontja
- targettype: A célpont típusa
- opreport: Jelentés a küldetésről (web link)
- opphoto: A bevetés során készített fénykép (web link)
- targetphoto: A bombázásról készült fénykép (web link)

A lekérdezés eredménye nem tartalmazta a végleges tábla teljes adatait, a **bombsites** táblához utólag hozzáadtam további 3 mezőt:

- **id** mező: *ALTER TABLE airfieldsag ADD COLUMN id SERIAL PRIMARY KEY;*
- **targypeen**: Angol nyelven is megadtam a célpont típust, a programozás során felmerülő karakterkódolási probléma elkerülése miatt
- **grname**: A küldetésben részt vevő osztály nevét is hozzáadtam a táblához.

A **targypeen** és **grname** mezőket a QGIS mező kalkulátorával töltöttem fel adattal.

Budapest – Budapest belterületének határa 1944-ben.

Budapest határához az Arcanum weboldalon található „Budapest (1945) • Budapest közigazgatási térképsorozata” [13] című térképet használtam fel.

A geometria szerkesztését az Open Street Map alaptérképen végeztem el, a weboldalon található térképet követve.

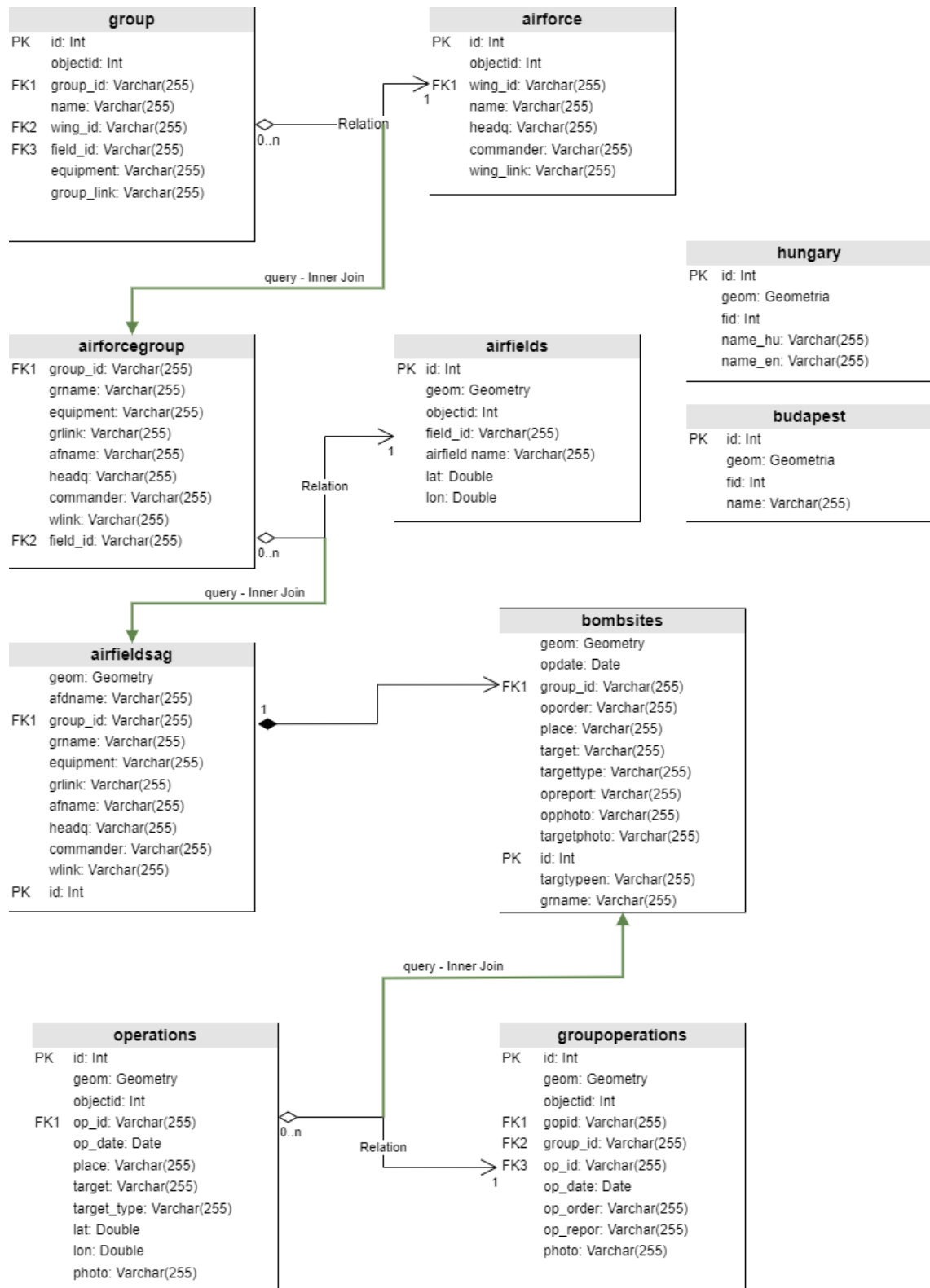
A szerkesztés végeztével a geometriát feltöltöttem a PostgreSQL adatbázisba EPSG:4326, WGS84 vetületi rendszerben.

Hungary – Magyarország határa 1944-ben.

Az 1944-es állapot szerinti országhatárt, Budapest határához hasonló módszerrel pontosítottam. Először a „MO_Telepules.shp” [17] állományból a „Magyarország közigazgatási térképe 1918-ban és 1944-ben. [B XV a 381]” [11] alapján leválogattam a határmenti településeket, ezeket a Merge eszközzel összevontam, majd a külső határokat megtartva létrehoztam az országhatár polygon elemét.

Az elkészült geometriát feltöltöttem az adatbázisba EPSG:4326, WGS84 vetületi rendszerben.

A táblák PostgreSQL adatbázisba történt feltöltése után elkészült az adatbázis sémája, amit a 13. ábrán látható UML diagram szemléltet.



13. ábra Az elkészült adatbázis UML diagramja

4. ALKALMAZÁSFEJLESZTÉS

Az alkalmazásfejlesztés megkezdése előtt térinformatikai rendszerben megterveztem az alkalmazás kinézetét és működését.

4.1 Alkalmazás tervezés

Az alkalmazás tervezés folyamatában is a QGIS szoftvert használtam. Egy üres projektet hoztam létre, ahova az adatbázisból betöltöttem a szükséges rétegeket.

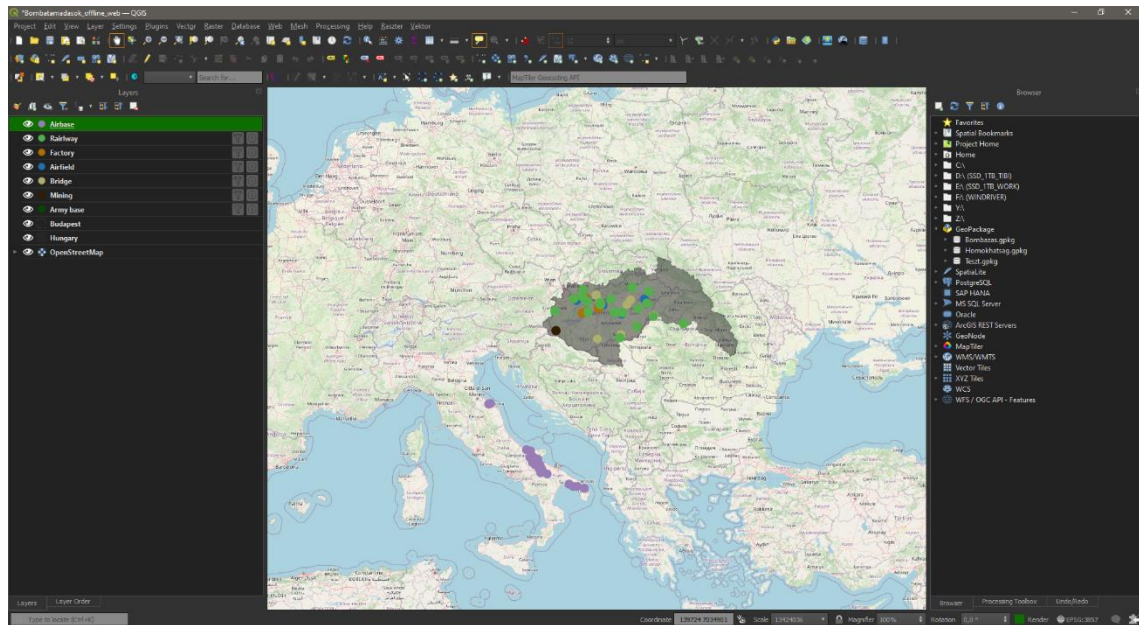
- Alaprétegek:
 - Budapest határa 1944-ben
 - Magyarország határa 1944-ben
- Légibázisok
- Célpontok
 - Vasúti infrastruktúra
 - Ipari létesítmények
 - Katonai létesítmények
 - Repülőterek
 - Bányák

A 10. ábrán szemléltetem az alkalmazás megjelenésének tervét.

Egy egyszerű, jól átlátható felület megvalósítása volt a célom, ahol ábrázolva vannak a 15. Légierő bázisai, a támadások kiindulópontjai, valamint a bombázások célpontjai. Ez már egy térbeli megjelenítése annak, hogy honnan indultak a repülőgépek, és hová tartottak. Alaptérképnek az Open Street Map térképét használtam a tervezés során. Ezt használtam alapfelületként, amikor a térképi rétegeket szerkesztettem.

A QGIS lehetőséget biztosít egy offline internet kapcsolat nélkül is megjeleníthető böngészőben megnyitható alkalmazás elkészítésére, ahol tesztelni lehet a térkép működését,

annak célközegében. Ennek a webalkalmazásnak a forráskódja is szerkeszthető, beállítható a rétegkezelés, a megjelenített elemek csoportosítása, nagyítás esetén csoportbontás, kattintásra eseményvezérlés, pl. felugró ablakban megjelenő információk



14. ábra Az alkalmazás terve QGIS-ben

Végül, az alkalmazás elkészítése során teljesen más megközelítést használtam. Azzal a céllal kezdtem el az Open Source alapú alkalmazás fejlesztést, hogy egy új módszert sajátítsak el. A mindennapokban az ESRI ArcGIS előre csomagolt, komplex rendszerét használom, ami kényelmes szolgáltatást biztosít a térkép készítéstől az online publikációig, előre gyártott modulokkal.

4.2 Alkalmazás fejlesztés

Az Open Source megoldásokból hiányzik az a kényelem, amit a „dobozos” szolgáltatások nyújtanak, így más megközelítést igényel egy alkalmazás felépítése. Az ArcGIS által készen nyújtott modulokat, a nyílt forráskódú fejlesztés során, sok esetben magunknak kell legyártani, bár nagyon sok megoldási lehetőséget nyújt az Open Source közösség is, azokat mindig személyre kell szabni, hogy be lehessen építeni a saját eszközünkbe. Ez adja a nyílt forráskódon alapuló alkalmazásfejlesztés szépségét, egyben nehézségét is.

A szoftverfejlesztéshez a Django projektet használtam PostgreSQL adatbázis eléréssel és LeafletJs alapú webalkalmazás fejlesztési módszert választottam, aminek folyamatát a továbbiakban részletesen tárgyalok.

4.2.1 Python

A térinformatikai szoftverek a Python programozási nyelvet használják beépített script nyelvként, amivel automatizálhatók a különböző munkafolyamatok, valamint egyéb műveletek is végrehajthatók a segítségével, többek között mezőszámítási műveletek és modellezés. Mivel a QGIS is ezt a nyelvet használja, a telepítés során ezt a könyvtárat is létrehozza a számítógépen.

Amennyiben nincs Python telepítve, azt meg kell tenni a fejlesztés megkezdése előtt. Mivel Microsoft® Windows operációs rendszert használok így az ehhez készült Python telepítése szükséges, amit a program weboldaláról lehet letölteni. [21]

Az alkalmazás készítéséhez a 3.10.x verziót telepítettem.

A Python egyik nagy előnye, hogy sokféle modul van hozzá, a függvénykönyvtár részeként, vagy letölthető függvénykönyvárakból. A program írása során rengeteg ilyen függvényt telepítünk, amit valahogy kezelni kell, hiszen nem várható el a felhasználoktól, ha használni szeretnék az alkalmazást maguknak telepítsék ezeket a modulokat.

- Virtualenv [29]

A Python futtató környezete, ami egy virtuális, elkülönített környezet, amit azon belül telepítünk, azt le lehet kérdezni és listát lehet készíteni belőlük, amit később fel lehet használni a modulok telepítésére.

A virtuális környezet és egyéb modulok telepítéséhez a ***pip*** csomagkezelőt használjuk.

„A ***pip*** (package installer for Python) egy csomagkezelő rendszer, amely operációs rendszertől függetlenül használható mindegyiken, és sok hasznos dolgot tud. Alapvetően, a Python 3.4 verziótól kezdődően tartalmazza a ***pip*** csomagkezelőt.

A ***pip*** alapvetően a **PyPI** (Python Package Index, <https://pypi.org/>) oldalról telepít csomagokat.” [29]

- Virtuális környezet létrehozása

Először létrehoztam egy mappát, ahova dolgozni fogok, amit **webapp** néven neveztem el. Ez a mappa gyökér könyvtára az alkalmazásomnak, ide mentettem minden szükséges, fájlt, illetve itt hoztam létre a többi könyvtárat.

A virtuális környezet telepítéséhez a Windows parancssorban, amit adminisztrátorként futtattam, beléptem a korábban elkészített mappába, majd az alábbi parancsot futtattam:

```
py -m pip install venv bombenv
```

A parancs futása után létrejött egy **bombenv** könyvtár a következő tartalommal:

Directory: \Webapp\bombenv

Mode	LastWriteTime	Length	Name
d----	2022. 12. 07. 13:05		Include
d----	2022. 12. 07. 13:05		Lib
d----	2022. 12. 07. 13:10		Scripts
-a----	2022. 12. 07. 13:05	118	pyvenv.cfg

Ezzel elkészült a virtuális környeztet, amit az alábbi parancs segítségével aktiváltam:

```
bombenv/Scripts/activate
```

A parancs indításának következménye, hogy zárójelben megjelenik a virtuális környezet neve a parancssorban.

```
(bombenv) PS \Webapp\bombenv>
```

Ettől kezdve bármilyen modult telepítettem az a **bombenv\Lib\site-packages** mappába került.

A virtuális környezetből a *deactivate* paranccsal lehet kilépni.

4.2.2 Django [30]

Ahhoz, hogy egy Django projektet létrehozzunk, telepíteni kell a virtuális környezetbe, a Django modult. Ezt a korábban is használt pip paranccsal lehet megtenni:

```
py -m pip install django
```

Ezzel a **bombenv\Lib\site-packages** mappában megjelent egy *asigref*, egy *django*, egy *sqlparse* és egy *tzdata* mappa.

Most már létre lehet hozni a Django projektet. Ehhez az alábbi parancsot futtattam:

```
django-admin startproject bombing
```

A parancs futása létrehozott egy **bombing** mappát az alábbi tartalommal:

```
bombing/
```

```
    manage.py
```

```
    bombing/
```

```
        __init__.py
```

```
        asgi.py
```

```
        settings.py
```

```
        urls.py
```

```
        wsgi.py
```

„A *manage.py* egy segédprogram, amely minden Django projekt része, parancssorban lehet futtatni. Ezzel lehet többek között alkalmazást készíteni, migrálni az adatbázist, szerveret indítani.” [31]

„A *settings.py* a projekt konfigurációit tartalmazza.” [31]

„Az *urls.py* tartalmazza a projekt URL címeit.” [31]

A **bombapp** Django alkalmazás elkészítéséhez beléptem abba a mappába, ahol a *manage.py* fájl található, majd a következő parancsot futtattam:

```
py manage.py startapp bombapp
```

Az alábbi könyvtárstruktúra jött létre a parancs futtatása:

```
bombing/
```

```
bombapp/  
    __init__.py  
    admin.py  
    models.py  
    tests.py  
    views.py  
    migrations/  
        __init__.py  
manage.py  
bombing/  
    __init__.py  
    asgi.py  
    settings.py  
    urls.py  
    wsgi.py
```

Ezzel elkészült az alkalmazás alapja.

Ahhoz, hogy térképi alkalmazásként működhessen, a Django projekthez további kiegészítők telepítését kellett futtatni.

A PostgreSQL adatbázis kezeléséhez a **psycopg2** nevű modul:

```
pip install psycopg2
```

A GeoJSON formátum értelmezéséhez a **django-geojson** és a **geojson** modulokat:

```
pip install django-geojson
```

```
pip install geojson
```

A Leaflet JS használatához pedig a ***django-leaflet*** modult:

```
pip install django-leaflet
```

A *pip freeze* parancs megmutatja, milyen modulok vannak telepítve az alkalmazáshoz:

```
asgiref==3.5.2
```

```
Django==4.1.4
```

```
django-geojson==4.0.0
```

```
django-leaflet==0.28.3
```

```
geojson==2.5.0
```

```
psycogp2==2.9.5
```

```
sqlparse==0.4.3
```

```
tzdata==2022.7
```

A *pip freeze > requirements.txt* parancs egy szövegfájlba menti a telepített modulokat, amit a *pip install -r requirements.txt* parancs futtatásával aktiválva a szövegfájlban lévő modulok telepíthetők egy virtuális környezetbe. Ez a lehetőség azért hasznos, mert ha egy új virtuális környezetet hozunk létre egy új Django projekttel, a szövegfájlba mentett modulokat egy lépésben telepíthetjük, kiküszöbölve, a hibázás lehetőségét, ha valamely modult elfelejtenénk telepíteni. Bár ebben az esetben hibaüzenet jelzi, hogy hiányzik egy modul az alkalmazás futtatásához.

„A Django-ban az alkalmazások és a projektek elkülönülnek, ezért a projektben regisztrálni kell az alkalmazást, illetve a telepített modulokat.” [31]

Ezt a *settings.py* fájlban lehet megtenni. A bombing mappában megnyitottam a *settings.py* fájlt, ehhez már a Visual Studio Code szerkesztőt használtam.

A fájlban az *INSTALLED_APPS* szekcióhoz hozzáadtam a *bombapp* alkalmazást, valamint a *gis*, a *leaflet*, és a *djgeojson* modulokat:

```
INSTALLED_APPS = [
```

```

'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'django.contrib.gis',
'leaflet',
'djgeojson',
'bombapp',
]

```

A modulok és az alkalmazás mellett a *settings.py* fájlban kell beállítani az adatbázis szerver kapcsolatot is, ezt a *DATABASES* szekcióban lehet megtenni:

```

DATABASES = {
    'default': {
        'ENGINE': 'django.contrib.gis.db.backends.postgis',
        'NAME': 'bombing',
        'USER': 'postgres',
        'HOST': 'localhost',
        'PASSWORD': 'postgres',
        'PORT': 5432,
    }
}

```

Az adatbázis alatt adtam hozzá a *geos_c* és *gdal306* fájlok elérését.

A két fájl az OSGeo4W [32] csomag telepítése után a számítógépen lévő OSGeo4W mappában található. Ez a két fájl szükséges ahhoz, hogy a django-leaflet kezelni tudja a különböző geometriai adatokat, esetünkben a GeoJSON formátumból geometriát állítson elő.

A settings.py fájlban így jelenik meg a két fájl elérési útja:

```
GEOS_LIBRARY_PATH = r'C:\OSGeo4W\bin\geos_c'
GDAL_LIBRARY_PATH = r'C:\OSGeo4W\bin\gdal306'
```

A serializer modult is itt adtam meg, ami a megjelenítéshez szükséges eszköz:

```
SERIALIZATION_MODULES = {
    'geojson': 'djgeojson.serializers',
}
```

Az *Internationalization* szekcióban beállítottam az alapértelmezett nyelvi beállítást és az időzónát:

```
LANGUAGE_CODE = 'hu'
TIME_ZONE = 'CET'
```

Ezzel az oldal adminisztrációs felülete magyar nyelvű lett, az időzóna pedig Közép-Európai.

Beállítottam statikus fájlok elérésének helyét:

```
STATICFILES_DIRS = [ BASE_DIR / "static"]
```

Végül a Leaflet beállításokat adtam hozzá a settings.py fájlhoz:

```
LEAFLET_CONFIG = {
    'DEFAULT_CENTER': (44.4914973, 15.8185459),
    'DEFAULT_ZOOM': 6,
```

```
'MAX_ZOOM': 17,

'MIN_ZOOM': 6,

'MINIMAP': True,

'RESET_VIEW': False,

'ATtribution_PREFIX': 'Bertli Tibor - Szakdolgozat - 2022',

}
```

A *settings.py* fájl beállítása után elkezdtem felépíteni az alkalmazást.

Először a modelleket kellett hozzáadni, a *models.py* fájlhoz.

„A Django modell minden olyan osztály, ami a *django.models.Model* függvényeinek gyűjteményét örökli. A gyűjtemény olyan eljárásokat tartalmaz, amelyekkel lekérdezhető, módosítható és frissíthető az adatbázis. Mezőket definiálhatunk, metaadatok állíthatók be és modellek közti kapcsolatok hozhatók létre.” [33]

GeoJSON fájl használata esetén *gis* modellt kell használni, amit a *models.py* fájlban a következő módon hívtam meg:

```
from django.contrib.gis.db import models
```

Ez lehetőséget ad, hogy az adatbázis kezelni tudja a geometriai adat formátumot, amihez segítségül hívja a **gdal** és a **geos** fordítókat.

A *models.py* fájlban alábbi modell osztályokat készítettem el:

Airfields

```
class Airfields(models.Model):

    Id = models.IntegerField(db_column='id', primary_key=True)

    airfieldName = models.CharField(db_column='afdname', max_length=255)

    groupId = models.CharField(db_column='group_id', max_length=255)

    groupName = models.CharField(db_column='grname', max_length=255)
```

```

equipment = models.CharField(db_column='equipment', max_length=255)
groupLink = models.CharField(db_column='grlink', max_length=255)
airforceName = models.CharField(db_column='afname', max_length=255)
headQ = models.CharField(db_column='headq', max_length=255)
commander = models.CharField(db_column='commander', max_length=255)
airforceLink = models.CharField(db_column='wlink', max_length=255)
geom = models.PointField(db_column='geom', srid=4326)

```

```

def __unicode__(self):
    return self.airfieldName

```

```

class Meta:
    managed = False
    db_table = 'airfieldsag'
    verbose_name_plural = 'Airfields'

```

BombSites

```

class BombSites(models.Model):
    Id = models.IntegerField(db_column='id', primary_key=True)
    opDate = models.DateField(db_column='opdate')
    groupId = models.CharField(db_column='group_id', max_length=255)
    groupName = models.CharField(db_column='grname', max_length=255)
    opOrder = models.CharField(db_column='oporder', max_length=255)
    place = models.CharField(db_column='place', max_length=255)

```

```

target = models.CharField(db_column='target', max_length=255)
targetType = models.CharField(db_column='targettype', max_length=255)
opReport = models.CharField(db_column='opreport', max_length=255)
opPhoto = models.CharField(db_column='opphoto', max_length=255)
targetPhoto = models.CharField(db_column='targetphoto', max_length=255)
targetEn = models.CharField(db_column='targypeen', max_length=255,)
geom = models.PointField(db_column='geom', srid=4326)

```

```

def __unicode__(self):
    return self.opDate

```

```

class Meta:
    managed = False
    db_table = 'bombsites'
    verbose_name_plural = 'Bomsites'

```

Budapest

```

class Budapest(models.Model):
    Id = models.IntegerField(db_column='id', primary_key=True)
    name = models.CharField(db_column='name', max_length=20)
    geom = models.PolygonField(db_column='geom', srid=4326)

    def __unicode__(self):
        return self.name

```

```

class Meta:

    managed = False

    db_table = 'budapest'

    verbose_name_plural = 'Budapest'

```

Hungary

```

class Hungary(models.Model):

    Id = models.IntegerField(db_column='id', primary_key=True)

    nameHu = models.CharField(db_column='name_hu', max_length=25)

    nameEn = models.CharField(db_column='name_en', max_length=25)

    geom = models.PolygonField(db_column='geom', srid=4326)

    def __unicode__(self):

        return self.nameHu

class Meta:

    managed = False

    db_table = 'hungary'

    verbose_name_plural = 'Hungary'

```

Az egyes mezőkben a mező tulajdonságok meghatározásánál a *db_column* tulajdonság a PostgreSQL táblában lévő mezőre mutat.

A **Meta** osztályban definiáltam a PostgreSQL adatbázis tábláját (*db_table*), ahonnan a **Model** osztály az adatot eléri.

„Django tartalmaz egy beépített adminisztrációs felületet, amely az alkalmazásban elérhető adatokelérésére használható, valamint egy teljes hitelesítési és engedélyezési implementációt is tartalmaz, amivel a rendszergazdai webhelyhez való hozzáférés szabályozható.” [34] Az adminisztrációs felület eléréséhez létre kell hozni egy *superuser*-t, aki az oldal adminisztrátora lesz.

A *py manage.py createsuperuser* parancs futtatásával hoztam létre az **admin** nevű felhasználót, aki a **siteadmin** jelszóval tud belépni az adminisztrációs felületre.

Az *admin.py* fájl szerkesztésével hozzáadtam az **Airfields** és **BombSites** osztályokat az adminisztrációs oldalhoz.

Először meghívtam a következő függvényeket, amik az adatmodell és a Leaflet elérést biztosítják az adminisztrációs weboldalnak:

```
from .models import Airfields, BombSites

from leaflet.admin import LeafletGeoAdmin
```

Ezután az alábbi parancssorral regisztráltam a modelleket az adminisztrációs felületen:

```
class AirfieldsAdmin(LeafletGeoAdmin):

    ordering = ('airfieldName', 'groupId')

    list_display = ('airfieldName', 'groupName')
```

```
class BombSitesAdmin(LeafletGeoAdmin):

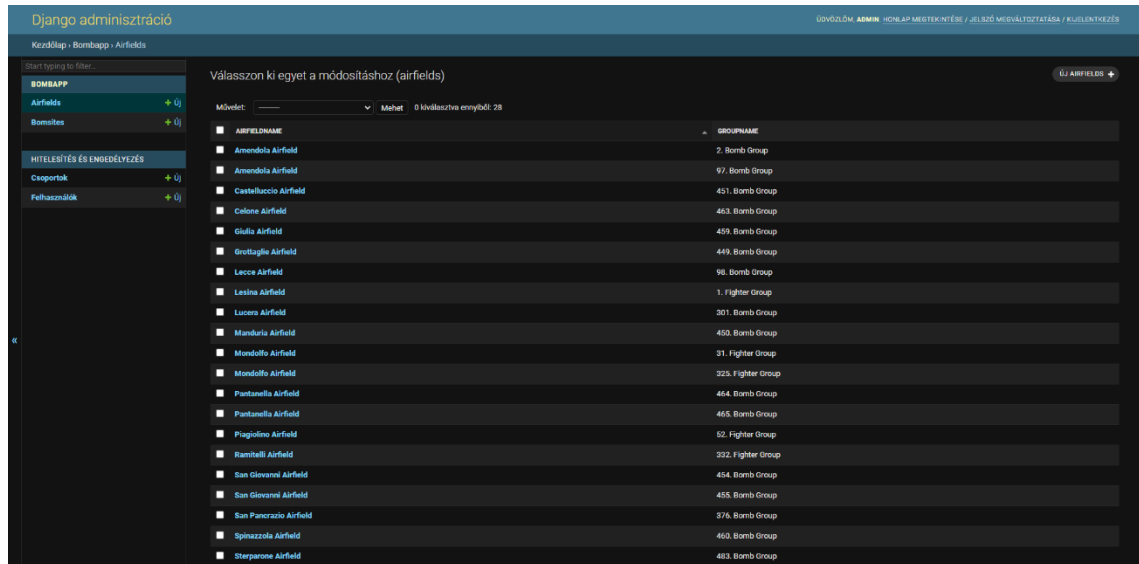
    ordering = ('opDate', 'groupId')

    list_display = ('opDate', 'place', 'target')
```

```
admin.site.register(Airfields, AirfieldsAdmin)

admin.site.register(BombSites, BombSitesAdmin)
```

Az *ordering* beállítja a rendezési sorrendjét az adatoknak, *list_display* pedig a mezőket adja meg, amik megjelennek a Django Admin oldalon. (15. ábra)



15. ábra A Django adminisztrációs felületén az *airfields* modell

A Django Adminisztrációs felületét akkor célszerű így beállítani, ha többfelhasználós szerkeszthető alkalmazást készítünk.

Ez az alkalmazás jelen állapotában nem ad lehetőséget felhasználói regisztrációra, így jogosultságokat sem állítottam be, ettől függetlenül, létrehoztam a később szerkeszthető rétegeket.

Az *admin.py* szerkesztése után megvalósítottam a nézeteket.

A nézet információkat jelenít meg a felhasználó számára. A nézet teljesíti a felhasználói kéréseket az HTML felől, elvégzi a megfelelő adatok betöltését a modellből, vagy módosítja a modellt, majd egy HTML sablont ad vissza válaszként a felhasználónak. [35]

Ehhez a *views.py* fájlt kellett szerkesztenem. Itt szintén függvényhívással kezdődik a folyamat:

```
from django.shortcuts import render

from django.core.serializers import serialize

from django.http import HttpResponse, HttpRequest
```

```
from .models import Airfields, Budapest, Hungary, BombSites
```

A függvényhívás után létrehoztam a nézeteket:

```
def home(request):
```

```
    return render(request, 'index.html', {
```

```
        'title': 'Kezdőoldal'
```

```
    })
```

```
def airfields_dataset(request):
```

```
    airfields = serialize('geojson', Airfields.objects.all())
```

```
    return HttpResponse(airfields, content_type='json')
```

```
def bombsites_dataset(request):
```

```
    bombsites = serialize('geojson', BombSites.objects.all())
```

```
    return HttpResponse(bombsites, content_type='json')
```

```
def budapest_dataset(request):
```

```
    budapest = serialize('geojson', Budapest.objects.all())
```

```
    return HttpResponse(budapest, content_type='json')
```

```
def hungary_dataset(request):
```

```
    hungary = serialize('geojson', Hungary.objects.all())
```

```
    return HttpResponse(hungary, content_type='json')
```

Először az alkalmazás kezdőoldalának nézetét készítettem el, majd a megjelenítendő modelleket.

A nézetek létrehozása még nem teszi azonnal megjeleníthetővé az adatokat, ehhez létre kellett hozni azok elérési útját. A **bombapp** mappában készítettem egy *urls.py* fájlt, ahol beállítottam a nézetben található modellek elérési útját:

```
from django.urls import include, path

from bombapp import views

urlpatterns = [

    path("", views.home, name='home'),

    path('airfields/', views.airfields_dataset, name="airfields"),

    path('bombsites/', views.bombsites_dataset, name="bombsites"),

    path('budapest/', views.budapest_dataset, name="budapest"),

    path('hungary/', views.hungary_dataset, name="hungary"),

]
```

Ennek az *urls.py* fájlnek az elérését beállítottam a Django projektnek is. Ehhez a projekt mappában (bombing) található *urls.py* fájlt módosítottam:

```
from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path("", include('bombapp.urls')),

    path('admin/', admin.site.urls),

]
```

4.2.3 Leaflet JS

A háttér folyamatok beállítása után létrehozhatók a *Template*-ek. Ezek olyan szövegfájlok, amelyek szöveg alapú formátumok, létrehozására használhatók. Esetünkben ez a szövegformátum a HTML.

A *Template*-ekhez a Django projekt mappában létrehoztam egy ***Templates*** mappát és ebben egy *index.html* nevű szövegfájlt.

Ez a fájl valósítja meg a felhasználói felületet. Mivel a frontend alapja egy Leaflet JavaScript könyvtár. Ez a virtuális környezethez van telepítve, a ***bombenv/Lib/site-packages/leaflet*** mappában.

Ennek eléréshez a következő címkéket kell beállítani a HTML fájlban:

A `<head>` jelölő elé:

```
{% load static %}

{% load leaflet_tags %}
```

Majd a `<head>` tárolóban meghívjuk a leaflet.js és leaflet.css fájlokat:

```
{% leaflet_js %}

{% leaflet_css %}
```

A `<head>` tároló tartalmazza a weboldal metaadatait, mint például az oldal címét, stílusát és a weboldalon futó script fájlok elérését.

Itt beállítottam a karakter kódolást *utf-8*-ra, meghívtam stílus fájlokat és beállítottam egy alap stílust az oldal működéséhez, script fájlokat importáltam és címet adtam az alkalmazásnak:

```
<meta charset="UTF-8">

<link rel="stylesheet" href="../static/css/style.css">

<title>Magyarország bombázása 1944-ben</title>
```

A `<head>` tároló alatt található a `<body>` tároló. Ide kerülnek a felhasználó számára megjelenő információk.

A `<body>` tárolóba egy JavaScript kódot írtam, ahol egy függvénybe AJAX kéréssel meghívtam az alkalmazás mappájában lévő `urls.py` fájlban létrehozott linkeket.

Ehhez szükséges a Leaflet AJAX JS könyvtár ezért a `<head>` tárolóban meghívtam a `leaflet.ajax.js`-t. A forráskódot a calvinmetcalf/leaflet-ajax [36] oldalról letöltöttem, majd a benne lévő **dist** mappát átmásoltam a Django projekt mappában létrehozott **static** mappába.

A hívást a következő sor hajtja végre a `<head>` tárolóban:

```
<script type="text/javascript" src="../../static/dist/leaflet.ajax.js"></script>
```

Egy Marker Cluster [37] nevű könyvtárat is hozzáadtam a HTML fájlhoz, ami a pontokat csoportosítja, ehhez stílusfájl is tartozik:

```
<link rel="stylesheet" href="../../static/dist/MarkerCluster.Default.css">
```

```
<script      type="text/javascript"      src="../../static/dist/leaflet.markercluster.js"></script>
```

A JavaScript függvényben változókat hoztam létre a térképen megjelenítendő rétegeknek:

Airfields

A Légi bázisok pont állománya stílussal:

```
var airfldDs = new L.GeoJSON.AJAX("{% url 'airfields' %}", {
  pointToLayer: function(feature, geom) {
    return L.circleMarker(geom, {
      fillColor: '#008080',
      color: '#537898',
      weight: 1,
      fillOpacity: 0.5,
```

```

    })
  },

```

A Légibázisok szövegbuborékokban megjelenő attribútumai:

```

    onEachFeature: function(feature, layer) {

      layer.bindPopup("<h2>" + feature.properties.airfieldName.toString() +
"</h2>" +

        "<table><tr><td><b>Osztály:</b></td><td><a      target='_blank'
href='" + feature.properties.groupLink + "'>" + feature.properties.gro-
upName.toString() + "</a></td></tr>" +

        "<tr><td><b>Repülőgép:</b></td><td>" + feature.properties.equ-
ipment.toString() + "</td></tr>" +

        "<tr><td><b>Ezred:</b></td><td><a target='_blank' href='" + fea-
ture.properties.airforceLink + "'>" + feature.properties.airforceName.toString() +
"</a></td></tr>" +

        "<tr><td><b>Parancsnokság:</b></td><td>" + feature.pro-
perties.headQ.toString() + "</td></tr>" +

        "<tr><td><b>Parancsnok:</b></td><td>" + feature.properties.com-
mander.toString() + "</td></tr></table>");

```

Eseményvezérlő, ha az egérrel a pontra állunk, megváltozik a pont körvonalának a színe:

```

    layer.on("mouseover", function(e) {

      layer.setStyle({

        color: 'yellow'

      });

    });

    layer.on("mouseout", function(e) {

```

```

        layer.setStyle({
            color: '#537898',
        });
    });
},
});

```

Pontok csoportosítása és a csoport hozzáadása a térképhez:

```

var afclusters = L.markerClusterGroup({
    showCoverageOnHover: false,
});

airfldDs.on('data:loaded', function() {
    afclusters.addLayer(airfldDs);
    afclusters.addTo(map);
});

```

BombSites

A bombázások célpontjainál egy lekérdezést is elvégeztem, mielőtt a pont réteget megjelenítettem. A következő rétegeket készítettem el, tárolt eljárásban:

Replülőtér

```

var tgAirfield = new L.GeoJSON.AJAX("{% url 'bombsites' %}", {
    pointToLayer: function(feature, geom) {

```

Lekérdezés a **BombSites** modellből, ahol a **targetEN = Airfield**:

```

    switch (feature.properties.targetEn) {
        case 'Airfield':

```

```

    return L.circleMarker(geom, {
        fillColor: '#02a0e3',
        color: '#60a7c4',
        weight: 1,
        fillOpacity: 0.5,
    })
}
},
onEachFeature: function(feature, layer) {
    //Popup - Repülőtér
    // #region - Popup
    layer.bindPopup("<h2>" + feature.properties.place.toString() + "</h2>" +
        "<p>" + feature.properties.opDate.toString() + "</p>" +
        "<table><tr><td><b>Osztály:</b></td><td>" + feature.pro-
        perties.groupName.toString() + "</td></tr>" +
        "<tr><td><b>Célpont:</b></td><td>" + feature.properties.tar-
        get.toString() + "</td></tr>" +
        "<tr><td><a target='_blank' href='" + feature.properties.opOrder +
        "'><b>Bevetési parancs</b></a></td></tr>" +
        "<tr><td><a target='_blank' href='" + feature.properties.opPhoto +
        "'><b>Bevetésen készült fénykép</b></a></td></tr>" +
        "<tr><td><a target='_blank' href='" + feature.properties.opReport +
        "'><b>Bevetési jelentés</b></a></td></tr>" +

```

```

    "<tr><td><a target='_blank' href='" + feature.properties.targetPhoto +
    "'><b>Fénykép a bombázásról</b></a></td></tr></table>");

```

```

    // #endregion

    layer.on("mouseover", function(e) {

        layer.setStyle({

            color: 'yellow'

        });

    });

    layer.on("mouseout", function(e) {

        layer.setStyle({

            color: '#60a7c4'

        });

    });

},

});

var tgafclusters = L.markerClusterGroup({

    showCoverageOnHover: false,

});

tgAirfield.on('data:loaded', function() {

    tgafclusters.addLayer(tgAirfield);

    tgafclusters.addTo(map);

});

// #endregion

```

Ipari infrastruktúra

```
var tgIndustry = new L.GeoJSON.AJAX("{% url 'bombsites' %}"), {
  pointToLayer: function(feature, geom) {
```

Lekérdezés a **BombSites** modellből, ahol a **targetEN = Industry**:

```
    switch (feature.properties.targetEn) {
      case 'Industry':
        return L.circleMarker(geom, {
          fillColor: '#519681',
          color: '#72968b',
          weight: 1,
          fillOpacity: 0.5,
        })
      }
    },
    onEachFeature: function(feature, layer) {
      layer.bindPopup("<h2>" + feature.properties.place.toString() + "</h2>" +
        "<p>" + feature.properties.opDate.toString() + "</p>" +
        "<table><tr><td><b>Osztály:</b></td><td>" + feature.pro-
        perties.groupName.toString() + "</td></tr>" +
        "<tr><td><b>Célpont:</b></td><td>" + feature.properties.tar-
        get.toString() + "</td></tr>" +
        "<tr><td><a target='_blank' href='" + feature.properties.opOrder +
        "'><b>Bevetési parancs</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opPhoto +
"><b>Bevetésen készült fénykép</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opReport +
"><b>Bevetési jelentés</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.targetPhoto +
"><b>Fénykép a bombázásról</b></a></td></tr></table>");
```

```
layer.on("mouseover", function(e) {
```

```
    layer.setStyle({
```

```
        color: 'yellow'
```

```
    });
```

```
});
```

```
layer.on("mouseout", function(e) {
```

```
    layer.setStyle({
```

```
        color: '#72968b'
```

```
    });
```

```
});
```

```
},
```

```
});
```

```
var indclusters = L.markerClusterGroup({
```

```
    showCoverageOnHover: false,
```

```
});
```

```
tgIndustry.on('data:loaded', function() {
```

```
    indclusters.addLayer(tgIndustry);
```

```
    indclusters.addTo(map);
```

```
});
```

Katonai létesítmény

```
var tgMilitary = new L.GeoJSON.AJAX("{% url 'bombsites' %}", {
  pointToLayer: function(feature, geom) {
```

Lekérdezés a **BombSites** modellből, ahol a **targetEN** = **Military**:

```
    switch (feature.properties.targetEn) {
      case 'Military':
        return L.circleMarker(geom, {
          fillColor: '#b5ac7d',
          color: '#787046',
          weight: 1,
          fillOpacity: 0.5,
        })
      }
    },
    onEachFeature: function(feature, layer) {
      layer.bindPopup("<h2>" + feature.properties.place.toString() + "</h2>" +
        "<p>" + feature.properties.opDate.toString() + "</p>" +
        "<table><tr><td><b>Osztály:</b></td><td>" + feature.pro-
        perties.groupName.toString() + "</td></tr>" +
        "<tr><td><b>Célpont:</b></td><td>" + feature.properties.tar-
        get.toString() + "</td></tr>" +
        "<tr><td><a target='_blank' href='" + feature.properties.opOrder +
        "'><b>Bevetési parancs</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opPhoto +
"><b>Bevetésen készült fénykép</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opReport +
"><b>Bevétesi jelentés</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.targetPhoto +
"><b>Fénykép a bombázásról</b></a></td></tr></table>");
```

```
layer.on("mouseover", function(e) {
```

```
    layer.setStyle({
```

```
        color: 'yellow'
```

```
    });
```

```
});
```

```
layer.on("mouseout", function(e) {
```

```
    layer.setStyle({
```

```
        color: '#787046'
```

```
    });
```

```
});
```

```
},
```

```
});
```

```
var milclusters = L.markerClusterGroup({
```

```
    showCoverageOnHover: false,
```

```
});
```

```
tgMilitary.on('data:loaded', function() {
```

```
    milclusters.addLayer(tgMilitary);
```

```
    milclusters.addTo(map);
```

```
});
```

Bánya:

```
var tgMining = new L.GeoJSON.AJAX("{% url 'bombsites' %}", {
    pointToLayer: function(feature, geom) {
```

Lekérdezés a **BombSites** modellből, ahol a **targetEN** = **Mining**:

```
    switch (feature.properties.targetEn) {
        case 'Mining':
            return L.circleMarker(geom, {
                fillColor: '#9e7b5a',
                color: '#856f5b',
                weight: 1,
                fillOpacity: 0.5,
            })
        }
    },
    onEachFeature: function(feature, layer) {
        layer.bindPopup("<h2>" + feature.properties.place.toString() + "</h2>" +
            "<p>" + feature.properties.opDate.toString() + "</p>" +
            "<table><tr><td><b>Osztály:</b></td><td>" + feature.pro-
            perties.groupName.toString() + "</td></tr>" +
            "<tr><td><b>Célpont:</b></td><td>" + feature.properties.tar-
            get.toString() + "</td></tr>" +
            "<tr><td><a target='_blank' href='" + feature.properties.opOrder +
            "'><b>Bevetési parancs</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opPhoto +
"><b>Bevetésen készült fénykép</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opReport +
"><b>Bevetési jelentés</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.targetPhoto +
"><b>Fénykép a bombázásról</b></a></td></tr></table>");
```

```
layer.on("mouseover", function(e) {
```

```
    layer.setStyle({
```

```
        color: 'yellow'
```

```
    });
```

```
});
```

```
layer.on("mouseout", function(e) {
```

```
    layer.setStyle({
```

```
        color: '#856f5b'
```

```
    });
```

```
});
```

```
},
```

```
});
```

```
var mineclusters = L.markerClusterGroup({
```

```
    showCoverageOnHover: false,
```

```
});
```

```
tgMining.on('data:loaded', function() {
```

```
    mineclusters.addLayer(tgMining);
```

```
    mineclusters.addTo(map);
```

```
});
```

Vasúti infrastruktúra:

```
var tgRailway = new L.GeoJSON.AJAX("{% url 'bombsites' %}", {
  pointToLayer: function(feature, geom) {
    switch (feature.properties.targetEn) {
```

Lekérdezés a **BombSites** modellből, ahol a **targetEN = Railway**:

```
    case 'Railway':
      return L.circleMarker(geom, {
        fillColor: '#a6ab52',
        color: '#7e8238',
        weight: 1,
        fillOpacity: 0.5,
      })
    }
  },
  onEachFeature: function(feature, layer) {
    layer.bindPopup("<h2>" + feature.properties.place.toString() + "</h2>" +
      "<p>" + feature.properties.opDate.toString() + "</p>" +
      "<table><tr><td><b>Osztály:</b></td><td>" + feature.pro-
      perties.groupName.toString() + "</td></tr>" +
      "<tr><td><b>Célpont:</b></td><td>" + feature.properties.tar-
      get.toString() + "</td></tr>" +
      "<tr><td><a target='_blank' href='" + feature.properties.opOrder +
      "'><b>Bevetési parancs</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opPhoto +
"><b>Bevetésen készült fénykép</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.opReport +
"><b>Bevetési jelentés</b></a></td></tr>" +
```

```
"<tr><td><a target='_blank' href='" + feature.properties.targetPhoto +
"><b>Fénykép a bombázásról</b></a></td></tr></table>");
```

```
layer.on("mouseover", function(e) {
```

```
    layer.setStyle({
```

```
        color: 'yellow'
```

```
    });
```

```
});
```

```
layer.on("mouseout", function(e) {
```

```
    layer.setStyle({
```

```
        color: '#7e8238'
```

```
    });
```

```
});
```

```
},
```

```
});
```

```
var rwclusters = L.markerClusterGroup({
```

```
    showCoverageOnHover: false,
```

```
});
```

```
tgRailway.on('data:loaded', function() {
```

```
    rwclusters.addLayer(tgRailway);
```

```
    rwclusters.addTo(map);
```

```
});
```

A pont rétegek előállítását követően elkészítettem a két polygon réteget.

Budapest:

```
var bpDs = new L.GeoJSON.AJAX("{% url 'budapest' %}", {
  style: function(feature) {
    return {
      fillColor: '#06b300',
      fillOpacity: 0,
      color: '#000000',
      weight: 3,
      dashArray: '10 5 2 5 10 5',
    }
  }
});
```

Országhatár:

```
var huDs = new L.GeoJSON.AJAX("{% url 'hungary'%}", {
  style: function(feature) {
    return {
      fillColor: '#000000',
      fillOpacity: 0,
      color: '#000000',
      weight: 3,
      dashArray: '20 10 5 10 20 5',
    }
  }
});
```

```

    }
  }
});

```

Készítettem egy Leírást és egy Jelmagyarázatot is:

```

let desc = L.control({
  position: "bottomleft"
});

desc.onAdd = function() {
  let div = L.DomUtil.create("div", "desc");
  div.innerHTML =
    '<h2>Magyarország bombázása 1944-ben</h2><hr>' +
    'A térkép az <b>1944. április és 1944. december</b> közötti időszakban,'
+
    'az <b>Amerikai Egyesült államok 15. Légierőjének</b> bombatámadá-
sait ábrázolja. ' +
    'A térképen megjelenik <b>Magyarország</b> határa és <b>Budapest</b>
belterületének határa a 2. Világháború idején.<hr>' +
    'A térképen látható ikonokra <b>kattinva</b>, egy felugró szövegdozban
megjelennek az elemhez tartozó információk.';

  return div;
};

let legend = L.control({
  position: "topright"
});

```

```

legend.onAdd = function() {

    let div = L.DomUtil.create("div", "legend");

    div.innerHTML =

        '<b>Jelmagyarázat</b><br>' +

        '15. Légierő<br>' +

        '<div style="background-color: #008080"></div>Légibázis<br><hr>' +

        'Bombatámadások<br>' +

        '<div style="background-color: #02a0e3"></div><div>Repülő-  
tér</div><br>' +

        '<div style="background-color: #519681"></div><div>Ipar</div><br>'

    +

        '<div style="background-color: #b5ac7d"></div><div>Lakta-  
nya</div><br>' +

        '<div style="background-color: #9e7b5a"></div><div>Bá-  
nya</div><br>' +

        '<div style="background-color: #a6ab52"></div><div>Vasút</div><br>';

    return div;

};

```

Majd hozzáadtam két alaptérképet, egy Open Street Map-et és egy ERSI Imagery térképet:

```

var osm = L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

    'attribution': '&copy; <a href="http://www.openstreet-  
map.org/copyright">OpenStreetMap</a>',

    'maxZoom': 19

```

```

    }).addTo(map)

    var imagery = L.tileLayer('http://services.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {
        'attribution': 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community',
        'maxZoom': 19
    }).addTo(map)

```

Végül elkészítettem a rétegvezérlőt és minden réteget hozzáadtam a térképhez.

Alaptérképek a rétegkapcsolóhoz:

```

var baseMaps = {
    "Satellite": imagery,
    "OpenStreetMap": osm,
};

```

Vektor rétegek:

```

var overlayMaps = {
    "A 15. Légierő bázisai": afclusters,
    "Repülőtér": tgafclusters,
    "Ipari infrastruktúra": indclusters,
    "Katonai létesítmény": milclusters,
    "Bánya": mineclusters,
    "Vasúti infrastruktúra": rwclusters,
};

```

A rétegvezérlő hozzáadása a tréképhez:

```
var layerControl = L.control.layers(baseMaps, overlayMaps).addTo(map)
```

A határ rétegek, a leírás és a jelmagyarázat hozzáadás a térképhez.

```
huDs.addTo(map);
```

```
bpDs.addTo(map);
```

```
desc.addTo(map);
```

```
legend.addTo(map);
```

Ezzel a rétegek függvénye elkészült, hozzá lehet adni a kezdőoldalhoz.

Ezt a `<body>` tárolóban a következő paranccsal lehet megtenni:

```
{% leaflet_map "gis" callback="window.our_layers" %}
```

Itt a Leaflet **gis** nevű térképe egy hívással visszakapja az **our_layers** nevű függvényt az ablakba. A `<head>` tárolóban beállítottam a *gis* stílusát, amivel meghatároztam, az ablak méreteit:

```
<style type="text/css">
```

```
#gis {
```

```
width: auto;
```

```
height: calc(100vh - 17px);
```

```
}
```

```
</style>
```

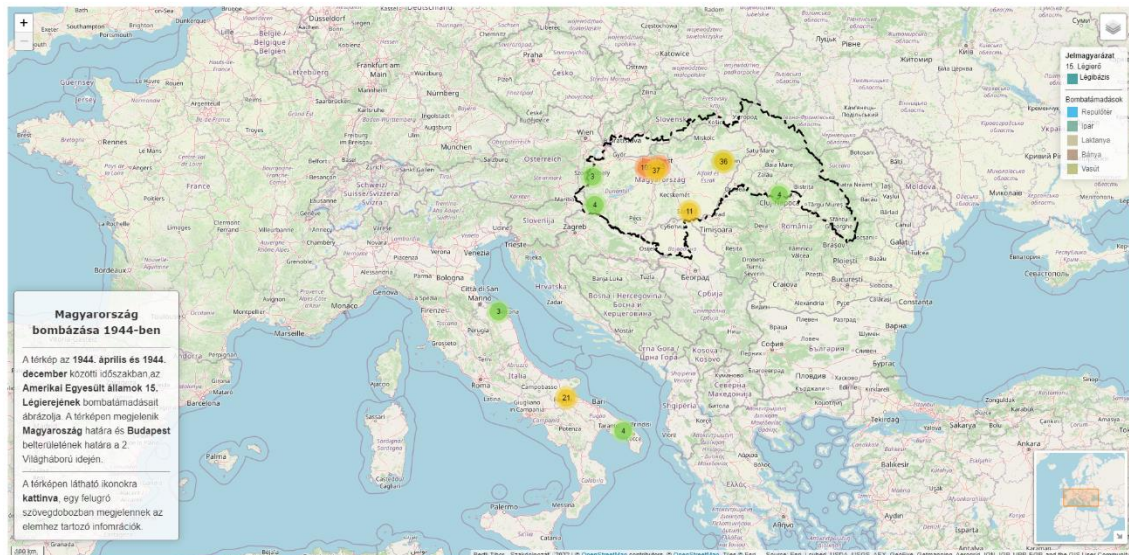
A fejlesztés során leginkább az ismerethiányból fakadó problémákkal szembesültem. Ezekre szerencsére online lehetett segítséget találni.

Az interneten sok olyan oldal áll rendelkezésre, melyek tovább vezetnek az elakadáson.

A tanulásban és a projekt megvalósításában sokat segített a „Geodjango Tutorial 1” [38] című videó oktató sorozat, a Ben Gurin University of the Negev „Introduction to Web

Mapping” [39] kurzusa, a „Django Leaflet Documentation” [40] című dokumentum, valamint a Stack Overflow [41] és Gis Stack Exchange [42] közösségi oldalakon is sok felemerülő problémára kaptam választ.

4.3 Az alkalmazás működése



16. ábra A Magyarország bombázása 1944-ben webalkalmazás kezdőlapja

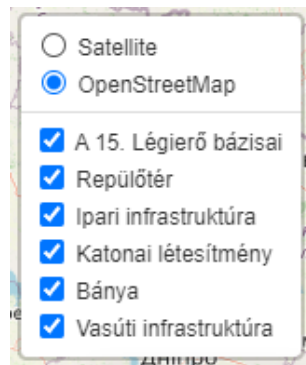
Az elkészült alkalmazás betöltése után a 16. ábrán látható nézetet láthatja a felhasználó.

A térkép középpontjának koordinátáit a Django projektben lévő *settings.py* fájlban beállított középponti koordináta adja. Itt lett beállítva a minimum és maximum zoom érték is.

A minimum zoomot a kezdőkép kiterjedésére határoztam meg, a maximum zoomot az ESRI World Imagery maximum zoom értékére adtam meg, mivel egy bizonyos érték felett ez a térkép nem szolgáltat adatokat, így az utolsó látható zoom tartományt adtam meg.

A nagyítás és kicsinyítés az egér görgővel és a jobb felső sarokban található nagyítás (+) és kicsinyítés (-) gombokkal is lehet, valamint a pontok csoportjára kattintva is a pontok szintjéig nagyít a térkép.

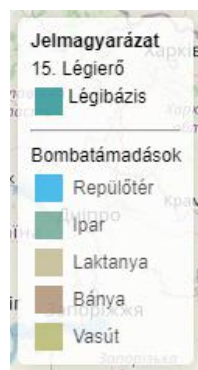
Az alkalmazás jobb felső sarkában van egy rétegkezelő ikon, ha ráhúzzuk az egeret, aktív lesz. Itt jelennek meg az alaptérképi rétegek rádió gombbal kapcsolhatók. (17. ábra)



17. ábra Rétegkapcsoló

A lekérdezések tárolt eljárásként vannak jelen a térképen. Ezek a rétegkezelőben ki- bekapcsolhatók, a réteg neve előtti jelölőnégyzet segítségével.

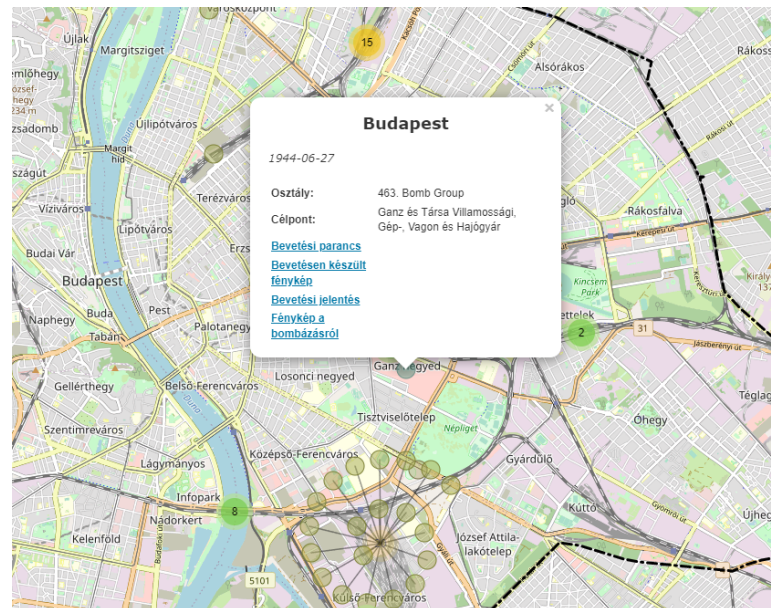
A rétegkezelő alatt egy statikus Jelmagyarázat látható. (18. ábra)



18. ábra Jelmagyarázat

A térkép nagyítása során a pontokra közelítve dinamikusan jelennek meg a pontok, amikre ráhúzva az egeret megváltozik a pont körvonalának színe, segítve a felhasználót a pontos kiválasztásban.

A pontra kattintva szövegdoboz jelenik meg, ami az elemhez tartozó információt jeleníti meg. Ezen felül aktív adatokat is tartalmaznak, amikre kattintva új oldalon, az elemhez kapcsolódó egyéb információk lekérdezhetők, például fényképek, szöveges dokumentumok, weboldalak. (19. ábra)



19. ábra Felugró ablak és a kibontott pontok csoportja

Egy csoportosított pontra kattintva, szétbomlik a csoport, így megjelennek azok a pontok, amik az adott helyen átfedik egymást, így minden egyes attribútum lekérdezhető. (19. ábra)

A képernyő bal alsó sarkában egy információs panelt helyeztem el, amin a térképpel kapcsolatos általános információk olvashatók. (20. ábra)



20. ábra Információs panel

A jobb alsó sarokban pedig egy átnézeti térkép látható.

4.4 Tapasztalatok és a továbbfejlesztés lehetőségei

Az alkalmazás ebben az állapotban megvalósítja a célját. Sikerült létrehozni egy működő, weboldalt, amivel szemléltethető a Geoinformatika nyújtotta lehetőségek használatát egy történettudományi munka megalapozásához.

Elkészült egy részleges forrásgyűjtemény, amely egy történelmi eseményen keresztül mutatja be a 2. világháború egy szegmensét.

A dolgozat készítése során, az oldal működésének leírása közben, további fejlesztési lehetőségek jutottak eszembe, a felhasználói élmény és interaktivitás bővítése érdekében.

Amit legfontosabbnak tartok, az a lekérdezési lehetőségek bővítése. Többféle lekérdezés megvalósításán gondolkodom:

- Időcsúszka: Mivel a *BombSites* modell dátum értéket is tartalmaz, egy időcsúszkával konkrét dátumra, időintervallumra is lehet szűrni, ez a lekérdezés lehetőséget ad animáció megjelenítésre is.
- Légibázis: Egy olyan lekérdezés generálása, amely segítségével ábrázolni lehet egy légibázisról indított támadás összes lehetséges célpontját, az időcsúszkával kombinálva pedig megvizsgálható, hogy egy adott napon a bázisról hova repültek.
- Bombázó és Vadász osztály: A fenti lekérdezés elvén működhet, az Osztályok felől vizsgálva az eseményeket.
- Bombázások célpontjai: Az előbb tárgyalt lekérdezések a célpontok felől vizsgálva.

További adatokkal is érdemes kibővíteni a térképet, hiszen a magyar védelmi rendszer megjelenítésére nem tértem ki a dolgozat elkészítése során, illetve a bombázások szórását sem vizsgáltam, azaz a találatok mennyire voltak pontosak, milyen károk keletkeztek a civil létesítményekben. Mindez további kutatást igényel, ami ennek a projektnek nem volt célja.

A lekérdezések mellett a tárolt adatok hozzáférhetőségét is biztosítani lehet, a lekérdezések eredményének szövegfájlba, vagy táblázatba mentésével.

Mindezen fejlesztések megvalósításával, további eddig feldolgozatlan forrás hozzáadásával, elkészülhet egy a kutatók és a téma iránt érdeklődők számára hasznos interaktív forrásgyűjtemény, amely a trébeli vizsgálat lehetőségét is biztosítja, kibővítve a történelem segédtudományainak tárházát egy modern eszközzel.

5. ÖSSZEFOGLALÓ

Dolgozatomban az 1944-es évben történt Magyarország elleni bombatámadásokon keresztül igyekeztem bemutatni a Geoinformatika által biztosított eszközök használhatóságát a történettudományi kutatómunkában.

A bevezetőben kitekintek a magyarországi történészképzésre, igyekszem megfejtetni, hogy milyen lehetőséget nyújt a képzés a hallgatók számára topográfiai, akár térinformatikai ismeretek megszerzése céljából. Sajnos elkeserítő eredményre jutottam, bár néhány helyen van topográfiai képzés, Geoinformatikával még ott sem ismerkednek, ahol erre lehetőség lenne.

A fentiektől függetlenül mégis vannak előremutató jelenségek a történettudomány és a térinformatika kapcsolatában, amit a második fejezetben részletesen kifejtek.

A következő fejezetek egy webalkalmazás elkészítését mutatják be, amin az Amerikai Egyesült Államok 15. Légierőjének hadműveleteit ábrázolom, a megvalósításhoz kifejezetten Open Source megoldásokat használva.

Először a PostgreSQL adatbázis tervezését és kivitelezését mutatom be, majd rátérek a Django - LeafletJS alkalmazásfejlesztésre. Bemutatom hogyan lehet kapcsolódni az adatbázishoz a modell és a nézetek segítségével, majd ebből a kapcsolatból hogyan lesz felhasználói felület. Végül ábrákkal szemléltetve bemutatom, hogyan működik az elkészült webalkalmazás.

A fejlesztés során merültek föl bennem olyan ötletek, amit ebben a munkában nem valósítottam meg, de egy továbbfejlesztés során beépíthetők lennének a jelenlegi alkalmazásba. Egyrészt a felhasználói élmény javítása további funkciókkal, időcsúszka beépítésével és olyan összetett lekérdezések megvalósításával, ahol az egyes repülőterekről mely célpontok ellen indítottak támadást, illetve ugyanez a lekérdezés Bombázó és Vadász Osztályokra elkészítve. Lehetővé tenni az adatok letöltését táblázatban.

Az oldal publikálás után lehetőséget biztosítani a regisztrált felhasználók (kutatók és a téma iránt érdeklődők) számára, az oldal tartalmának szerkesztésére és validáció után az adatokat megjelenítésére. Az adatokat letölthetővé tenni táblázat formájában.

6. SUMMARY

In my thesis, I tried to demonstrate the usability of the tools provided by Geoinformatics in historical research through the bomb attacks against Hungary in 1944.

In the introduction, I look at the university level history education in Hungary, I explore the education opportunities that are available to students when they are learning about topography and GIS. Unfortunately, I came to a disappointing conclusion. Although topographic studies are available in some places, Geoinformatics is not part of the history curriculum not even at universities where Geoinformatics is represented as a separate department at university.

Meanwhile there are advances being made at the Hungarian Academy of Sciences. I will explain this in the second chapter.

In the following chapters I present how I created a web application that dynamically maps the operations of the US 15th Air Force, specifically using Open Source solutions for implementation.

First, I will present the design and implementation of the PostgreSQL database, and then I will move on to the explanation of the Django - LeafletJS application development. I will show you how to connect to the database using the model and the views, and then how this connection becomes a user interface. Finally, I will show you how the completed web application works, using images.

During the development, I came up with ideas that I did not implement in this project, but they could be incorporated into the current application during further development. For example: improving the user experience with additional functions, incorporating a time slider and implementing complex queries where the attacks were launched from to which targets, and the same query should be made possible for the Bomber and Fighter Groups. Spreadsheet export could be made possible as well.

After launching the application, registered users (researchers and other people interested in the topic) could be provided with opportunities to edit the map data and improve the service (but only after their edits were professionally validated).

IRODALOMJEGYZÉK

- [1] S. Rózsa, Térinformatika és történettudomány, %1. kötet, összesen: %2Hagyományos források, új megközelítések: A digitalizáció kínálta lehetőségek a történeti kutatásokban, Eger: Eszterházy Károly Egyetem Líceum Kiadó, 2019, p. 103 – 133.
- [2] „PPKE - BTK Régészeti Térinformatikai Laboratórium,” [Online]. Available: <https://btk.ppke.hu/tudomanyos-kozelet/kutatasok/regeszeti-terinformatikai-laboratorium>. [Hozzáférés dátuma: 10 12 2022].
- [3] „Országgyűlési Könyvtár,” Országgyűlés Hivatala, [Online]. Available: <https://www.ogyk.hu/uploads/ogyk/tartalom/mo-neprajzi-terkepe-hu.jpg>. [Hozzáférés dátuma: 10 12 2022].
- [4] D. Gábor, „<http://real.mtak.hu/>,” 01 01 2019. [Online]. Available: <http://real.mtak.hu/89067/1/GISStoriaDG.pdf>. [Hozzáférés dátuma: 10 12 2022].
- [5] „GISTa Hungarorum,” MTA BTK Történettudományi Intézete, Pécsi Egyetem, Debrecen Egyetem, Eötvös Loránd Tudományegyetem, 31 12 2017. [Online]. Available: <https://www.gistory.hu/g/hu/gistory/index>. [Hozzáférés dátuma: 10 12 2022].
- [6] M. Béla, Térbeli döntések támogatása, %1. kötetTérinformatikai ismeretek 5, Székesfehérvár: Nyugat Magyarországi Egyetem, Geoinformatikai kar, 2010.
- [7] Nullaév, „Magyarország elleni bombatámadások 1944-1945,” Nemzeti Emlékezet Bizottsága, 1 10 2014. [Online]. Available: http://marvinn.carto.com/viz/f44798e8-183e-11e5-8537-0e018d66dc29/embed_map#zoom_in. [Hozzáférés dátuma: 10 12 2022].
- [8] „CARTO,” CARTO, 2012. [Online]. Available: <https://carto.com/>. [Hozzáférés dátuma: 10 12 2022].

- [9] „GISTa Hungarorum letöltés,” 01 01 2017. [Online]. Available: https://www.gistory.hu/docs/1_MO-HOR_Shp/1_MO-HOR_Shp_EPSG3857/. [Hozzáférés dátuma: 10 12 2022].
- [10] „Hungaricana,” Petőfi Irodalmi Múzeum, Arcanum Adatbázis Kft., [Online]. Available: <https://www.hungaricana.hu/hu/>. [Hozzáférés dátuma: 10 12 2022].
- [11] „Hungaricana térképtár,” Petőfi Irodalmi Múzeum, Arcanum Adatbázis Kft., [Online]. Available: <https://maps.hungaricana.hu/hu/HTTITerkeptar/33455/view/?bbox=-7134%2C-12257%2C21828%2C1013>. [Hozzáférés dátuma: 10 12 2022].
- [12] „Arcanum,” Arcanum Adatbázis Kiadó Magyarország, [Online]. Available: <https://www.arcanum.com/hu/>. [Hozzáférés dátuma: 10 12 2022].
- [13] „Budapest (1945) • Budapest közigazgatási térképsorozata,” Arcanum Adatbázis Kiadó Magyarország, [Online]. Available: <https://maps.arcanum.com/hu/map/budapest-1945/?layers=103&bbox=2083569.7410770797%2C6011564.768357%2C2156299.573490425%2C6038814.5689406665>. [Hozzáférés dátuma: 10 12 2022].
- [14] „The Fifteenth Air Force,” Glantzberg, Hughes, [Online]. Available: <https://15thaf.org/index.htm>. [Hozzáférés dátuma: 10 12 2022].
- [15] „Fortepan,” Szepessy Ákos, Tamási Miklós, 2010. [Online]. Available: <https://fortepan.hu/hu/photos/?q=bomb%C3%A1z%C3%A1s>. [Hozzáférés dátuma: 10 12 2022].
- [16] B. Zoltán, „Computerworld,” 04 09 2020. [Online]. Available: <https://computerworld.hu/tech/minden-amit-a-nyilt-forraskodu-technologirol-tudni-erdemes-283907.html>. [Hozzáférés dátuma: 10 12 2022].
- [17] „GISTa Hungarorum letöltés,” MTA BTK Történettudományi Intézete, Pécsi Egyetem, Debrecen Egyetem, Eötvös Lóránt Tudományegyetem, 01 01 2017.

- [Online]. Available: https://www.gistory.hu/docs/1_MO-HOR_Shp/1_MO-HOR_Shp_EPSG3857/. [Hozzáférés dátuma: 10 12 2022].
- [18] „PostgreSQL,” The PostgreSQL Global Development Group, 1996. [Online]. Available: <https://www.postgresql.org/>. [Hozzáférés dátuma: 10 12 2022].
- [19] „pgAdmin,” PostgreSQL Community Association of Canada, [Online]. Available: <https://www.pgadmin.org/>. [Hozzáférés dátuma: 10 12 2022].
- [20] „Visual Studio Code,” Microsoft, [Online]. Available: <https://code.visualstudio.com/>. [Hozzáférés dátuma: 10 12 2022].
- [21] „Python,” Python Software Foundation, 1989. [Online]. Available: <https://www.python.org/>. [Hozzáférés dátuma: 10 12 2022].
- [22] „Django,” Django Software Foundation, 2003. [Online]. Available: <https://www.djangoproject.com/>. [Hozzáférés dátuma: 10 12 2022].
- [23] „Leaflet Js,” Volodymyr Agafonkin, [Online]. Available: <https://leafletjs.com/>. [Hozzáférés dátuma: 10 12 2022].
- [24] „GeoJSON,” 2022. [Online]. Available: <https://geojson.org/>. [Hozzáférés dátuma: 10 12 2022].
- [25] O. A. Pál, „Újkor,” 03 04 2019. [Online]. Available: <https://ujkor.hu/content/budapest-bombazasa-ii-vilaghaboruban>. [Hozzáférés dátuma: 10 12 2022].
- [26] „Second Bombardment Association,” [Online]. Available: <http://www.2ndbombgroup.org/15thAirForce.htm>. [Hozzáférés dátuma: 10 12 2022].

- [27] „Foggia Airfield Complex,” [Online]. Available: https://www.wikiwand.com/en/Foggia_Airfield_Complex. [Hozzáférés dátuma: 10 12 2022].
- [28] R. L. S. G. Pataky Iván, „Légi háború Magyarország felett,” 1992. [Online]. Available: <https://www.arcanum.com/hu/online-kiadvanyok/2vhSzakkonyv-magyarok-a-ii-vilaghaboruban-2/legi-haboru-magyarorszag-felett-6E95/>. [Hozzáférés dátuma: 10 12 2022].
- [29] A. Gábor, „7. gyakorlat,” 29 03 2022. [Online]. Available: <https://okt.inf.szte.hu/python-programozas/gyakorlat/gyak7/>. [Hozzáférés dátuma: 10 12 2022].
- [30] „Adataalapú webhelyek létrehozása a Django Python-keretrendszerrel,” Microsoft, 2022. [Online]. Available: <https://learn.microsoft.com/hu-hu/training/modules/django-get-started/5-exercise-first-project>. [Hozzáférés dátuma: 10 12 2022].
- [31] „Adataalapú webhelyek létrehozása a Django Python-keretrendszerrel, Gyakorlat – Az első projekt létrehozása,” 2022. [Online]. Available: <https://learn.microsoft.com/hu-hu/training/modules/django-get-started/5-exercise-first-project>. [Hozzáférés dátuma: 10 12 2022].
- [32] „OSGeo4W,” OSGeo, 2022. [Online]. Available: <https://trac.osgeo.org/osgeo4w/>. [Hozzáférés dátuma: 10 12 2022].
- [33] „Adataalapú webhelyek létrehozása a Django Python-keretrendszerrel - Modellek a Django-ban,” 2022. [Online]. Available: <https://learn.microsoft.com/hu-hu/training/modules/django-models-data/4-models>. [Hozzáférés dátuma: 10 12 2022].
- [34] „Adataalapú webhelyek létrehozása a Django Python-keretrendszerrel, A felügyeleti webhely engedélyei,” 2022. [Online]. Available: <https://learn.microsoft.com/hu-hu/training/modules/django-models-data/4-models>.

- hu/training/modules/django-admin-site/3-admin-permissions. [Hozzáférés dátuma: 10 12 2022].
- [35] „Adataalapú webhelyek létrehozása a Django Python-keretrendszerrel, Ismerkedés a nézetekkel,” 2022. [Online]. Available: <https://learn.microsoft.com/hu-hu/training/modules/django-views-templates/3-views>. [Hozzáférés dátuma: 10 12 2022].
- [36] „Github, calvinmetcalf/leaflet-ajax,” 2015. [Online]. Available: <https://github.com/calvinmetcalf/leaflet-ajax>. [Hozzáférés dátuma: 10 12 2022].
- [37] „GitHub, Leaflet/Markercluster,” 2017. [Online]. Available: <https://github.com/Leaflet/Leaflet.markercluster#customising-the-clustered-markers>. [Hozzáférés dátuma: 10 12 2022].
- [38] W. Kibui, „Youtube, Geodjango Series 1,” 08 01 2017. [Online]. Available: https://www.youtube.com/playlist?list=PL7amXK4vKqATa_KrfQ3_tEF_ywAgAqWeJ. [Hozzáférés dátuma: 10 12 2022].
- [39] M. Dorman, „Introduction to Web Mapping,” 2022. [Online]. Available: <http://132.72.155.230:3838/js/index.html>. [Hozzáférés dátuma: 10 12 2022].
- [40] M. Corpus, „Django Leaflet Documentation,” 24 10 2022. [Online]. Available: <https://readthedocs.org/projects/django-leaflet/downloads/pdf/latest/>. [Hozzáférés dátuma: 10 12 2022].
- [41] „Stack Overflow,” Stack Overflow Ltd., 2022. [Online]. Available: <https://stackoverflow.com/>. [Hozzáférés dátuma: 10 12 2022].
- [42] „GIS Stack Exchange,” 2022. [Online]. Available: <https://gis.stackexchange.com/>. [Hozzáférés dátuma: 10 12 2022].
- [43] „Szoftverfejlesztés Tanszék,” SZTE TTIK Informatikai Intézet, 25 09 2022. [Online]. Available: <https://okt.inf.szte.hu/>. [Hozzáférés dátuma: 10 12 2022].

- [44] A. Gábor, „Python programozás a gyakorlatban,” SZTE TTIK Informatikai Intézet, Szoftverfejlesztés Tanszék, 11 02 2021. [Online]. Available: <https://okt.inf.szte.hu/python-programozas/>. [Hozzáférés dátuma: 10 12 2022].
- [45] „GitHub,” Microsoft, 2022. [Online]. Available: <https://github.com/>. [Hozzáférés dátuma: 10 12 2022].

7. ÁBRAJEGYZÉK

1. ábra Térinformatikai/topográfiai/történeti földrajzi tárgyak történelem szakon	1
2. ábra Magyarország néprajzi térképe népsűrűség alapján [https://www.ogyk.hu/uploads/ogyk/tartalom/mo-neprajzi-terkepe-hu.jpg]	5
3. ábra A Nemzeti Emlékezet Bizottsága térképének nyitólapja [https://marvinn.carto.com/viz/f44798e8-183e-11e5-8537-0e018d66dc29/embed_map#zoom_in]	7
4. ábra A CARTO oldalon a térkép adatbázisa [https://marvinn.carto.com/tables/bombatamadasok_uj/public]	8
5. ábra Adatbázis exportálás a CARTO felületéről [https://marvinn.carto.com/tables/bombatamadasok_uj/public]	8
6. ábra Az airforce és a group táblák	16
7. ábra Az airforcegroup tábla	17
8. ábra airfields tábla.....	17
9. ábra Az airfields és az airforcegroup táblák kapcsolata	17
10. ábra Az airfieldsag tábla	18
11. ábra Az operations és groupoperations táblák kapcsolata	19
12. ábra A bombsites tábla.....	20
13. ábra Az elkészült adatbázis UML diagramja	22
14. ábra Az alkalmazás terve QGIS-ben.....	24
15. ábra A Django adminisztrációs felületén az airfields modell	37
16. ábra A Magyarország bombázása 1944-ben webalkalmazás kezdőlapja	59
17. ábra Rétegh kapcsoló	60
18. ábra Jelmagyarázat.....	60
19. ábra Felugró ablak és a kibontott pontok csoportja	61
20. ábra Információs panel.....	61