

# A 2023/24-es tanév első félévének Webprogramozás II. tárgyához készített dokumentáció

## A feladat leírása

- A feladat szerint projektmunkában, csapatokat alkotva kellett létrehozni egy weboldalt a megadott adatbázisok alapján. A feladat elkészítése során a következő programokat használtuk:
- PHP Storm
- XAMPP v3.2.4
- Postman
- Filezilla

A kiírásban a következő szempontoknak kellett megfelelni.

- A weboldal alkalmazza az objektum-orientált PHP elveit.
- Többszintes menürendszer megvalósítása.
- Regisztrációs és bejelentkező felület létrehozása. Különböző szerepkörökkel ellátott felhasználók meghatározása.
- Restful webszolgáltatás létrehozása az adatbázis egyik táblájához. (GET, POST,PUT,DELETE funkciók, Postman API platformon való tesztelés)
- Ingyenes Restful API, amely tudja tesztelni a GET, POST, PUT, DELETE funkciókat egy oldalon.
- PDF készítő szolgáltatás.
- Github verziókövető rendszer használata..
- Az elkészült weblap feltöltése egy webszerverre.

## A feladat megvalósítása

**1.feladat:** Az első oldalon mutassa be a céget egy reszponzív weboldalon.

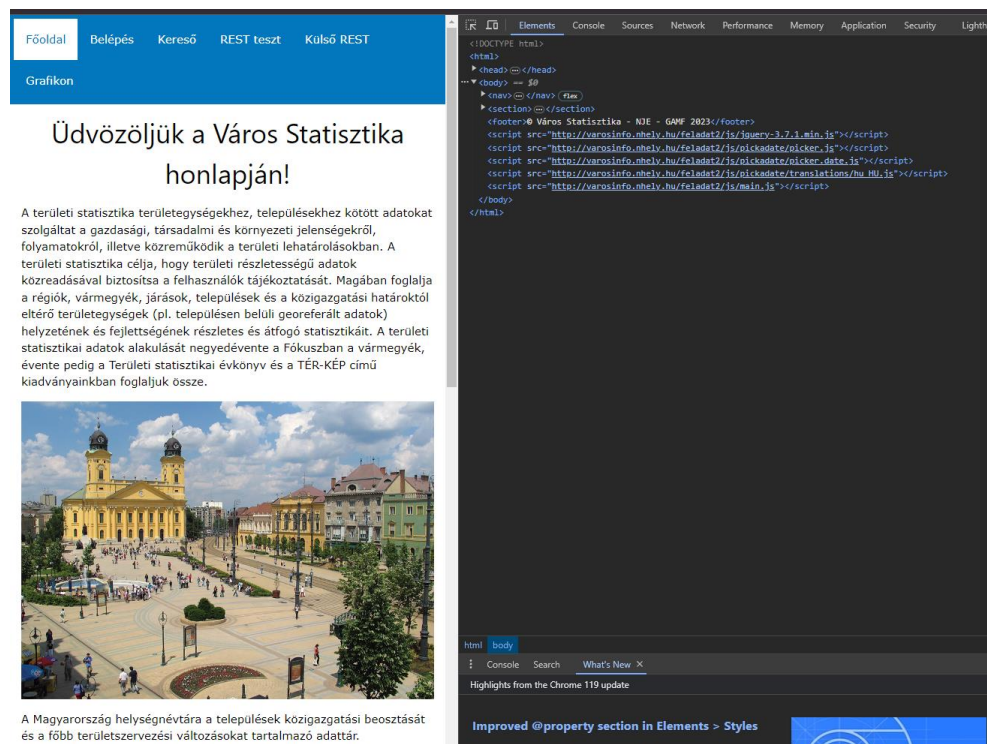
- a, alkalmazza az objektum-orientált PHP elveit
- b, a menüpontok neveit és a menüpontokhoz tartozó oldalak azonosítóját az adatbázisban tárolja,
- c, többszintes menürendszert megvalósít

d, Regisztráció, Bejelentkezés: legalább „látogató”, „regisztrált látogató” és „admin” szerepkör el van különítve

1.ábra



2.ábra

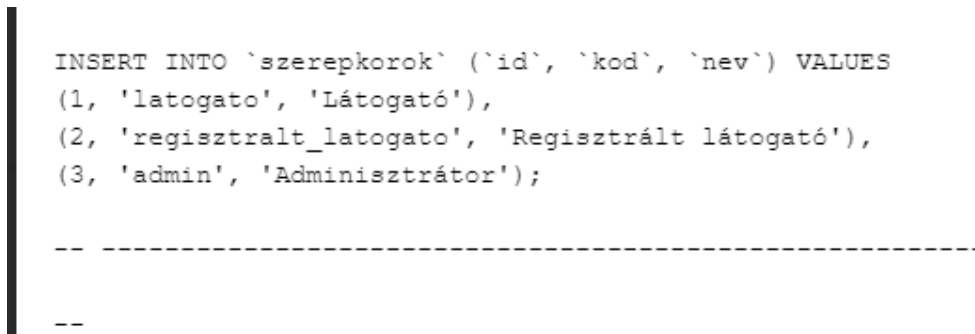


A nyitólapon látható informatív szöveget az adatbázis alapján egy területi statisztikai adatokkal foglalkozó honlapról másoltuk néhány kép kíséretében. A teljes weboldal reszponzív módon készült, íme néhány screenshot a megvalósításról.

### 3.ábra



### 4.ábra



Az első feladat megoldása az 1-4. ábrán látható képeken szerepel. A menüpontok neveit és az oldalak azonosítóját az adatbázisban tároltuk el. A többszintes menürendszer megvalósítását a 3. ábrán láthatjuk. A 4. ábrán láthatóak a szerepkörök melyeket szintén egy adatbázisban tárolunk el, amennyiben a beregisztrált felhasználó jogkörét szeretnénk változtatni, ezt az adatbázisban kell megtegyük.

**3.feladat:** „Alkalmazzon a weboldalon egy Ajax megvalósítást az adatbázis adatai alapján. A felhasználó 3 szöveges beviteli mezőben, vagy lenyíló listában megad adatokat és a rendszer az adatbázisból olvasva megjeleníti az adatokat Ajax megvalósítással. Olyan feladatot valósítson meg, amelyben az adatbázis mind a három táblájára szükség van.

5.ábra

Összesen	Férfi	Nő
----------	-------	----

Megye:  
Válasszon ...

Város:  
Válasszon ...

Év:  
Válasszon ...

PDF Export

6.ábra

Összesen	Férfi	Nő
----------	-------	----

Megye:  
Válasszon ...

- Válasszon ...
- Bács-Kiskun
- Baranya
- Békés
- Borsod-Abaúj-Zemplén
- Budapest
- Csongrád
- Fejér
- Győr-Moson-Sopron
- Hajdú-Bihar
- Jász-Nagykun-Szolnok**
- Komárom-Esztergom
- Nógrád
- Pest
- Somogy
- Szabolcs-Szatmár-Bereg
- Tolna
- Vas
- Veszprém

7.ábra

Összesen	Férfi	Nő
3684	1923	1761

Megye:  
Baranya

Város:  
Harkány

Év:  
2004

PDF Export

A 5-6-7. ábrákon látható, hogyan működik az Ajax megvalósítás a feladathoz megadott adatbázisok alapján. Az 5. ábrán maga a felület a 3 beviteli mezővel, melyek lenyíló listában jelentik meg az adatbázisban szereplő adatok. A 6. ábrán látható maga a lenyíló lista, a

7.ábrán pedig már a lefuttatott lekérdezés eredményét láthatjuk. A három beviteli mező függ egymástól, „Várost” akkor fogunk látni a listában ha választottunk „Megyét”, ugyan ez igaz az „Év”-re is. A megadott adatbázis mindhárom tábláját felhasználtuk a feladat megvalósításához.

#### 4.feladat:

Készítsen egy Restful webszolgáltatást az adatbázisának egyik táblájához. Valósítsa meg a GET, POST, PUT, DELETE funkciókat. Készítsen a webszolgáltatáshoz egy klienst, amivel tudja tesztelni a szolgáltatást az egyik oldalon.

Tesztelje a webszolgáltatást a Postman API Platform-al is: <https://www.postman.com/>

A következő ábrákon (8-9) a feladathoz tartozó megvalósítást láthatjuk, melyben szerepel egy lenyíló lista amiből ki tudjuk választani az adott „Város”-t, ami ezután megjeleníti az adatbázisban szereplő többi adatot is. Ehhez tudunk POST, DELETE, PUT funkciókkal hozzáadni avagy eltávolítani új adatokat. A 10.ábrán látható a POST funkció eredménye, amely hozzáadta az új értékeket a már meglévő elemekhez a táblázatban. A 11, 12. ábra a PUT és DELETE lefuttatott funkciók eredményeit mutatják be.

8.ábra

Főoldal	Minta	Kereső	REST teszt	Külső REST	Grafikon	Kilépés	Bejelentkező: koss.janos (janoskoss)
Város kiválasztása							
Város kiválasztása							
Abony							
Küldés							
Év	Nő	Férfi	Összesen				
2001	8007	7517	15524				
2002	8032	7531	15563				
2003	8052	7579	15631				
2004	8096	7574	15670				
2005	8110	7614	15724				
2006	8082	7612	15694				
2007	8054	7656	15710				
2008	8003	7628	15631				
2009	7976	7576	15552				
2010	7977	7565	15542				
2011	7879	7492	15371				
2012	7835	7446	15281				
2013	7833	7414	15247				
2014	7837	7410	15247				
2015	7797	7374	15171				
2016	7756	7339	15095				
2017	7740	7316	15056				
2018	7712	7256	14968				

9.ábra

POST

Év  
2023

Mű  
0

Összesen  
0

Küldés

PUT

Év  
2023

Mű  
0

Összesen  
0

Küldés

DELETE

Év  
2023

Küldés

10.ábra

2016	7756	7339	15095
2017	7740	7316	15056
2018	7712	7256	14968
2019	7685	7256	14941
2023	341	1959	2300

POST

Év  
2023

Mű  
341

Összesen  
2300

11.ábra

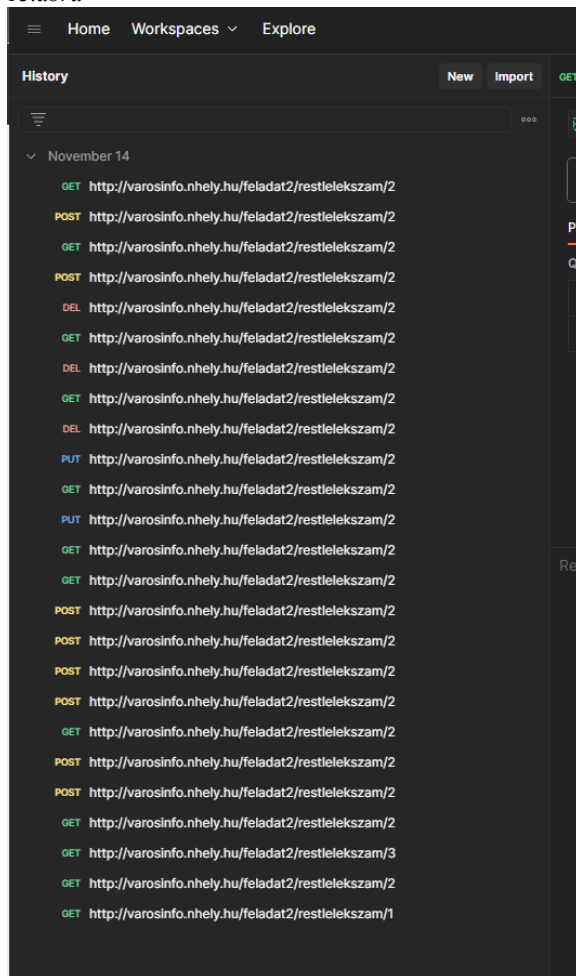
2018	7712	7256	14968
2019	7685	7256	14941
2023	341	4159	4500

12.ábra

2009	7976	7576	15552
2010	7977	7565	15542
2011	7879	7492	15371
2012	7835	7446	15281
2013	7833	7414	15247
2014	7837	7410	15247
2015	7797	7374	15171
2016	7756	7339	15095
2017	7740	7316	15056
2018	7712	7256	14968
2019	7685	7256	14941

A feladat tesztelés során használtuk a Postman platformot is, az eredményeket sikeresen visszakaptuk az adatbázisban. A következő képen a lefuttatott funkciók számát szeretnénk bemutatni, minden lekérdezésünk sikeres volt.

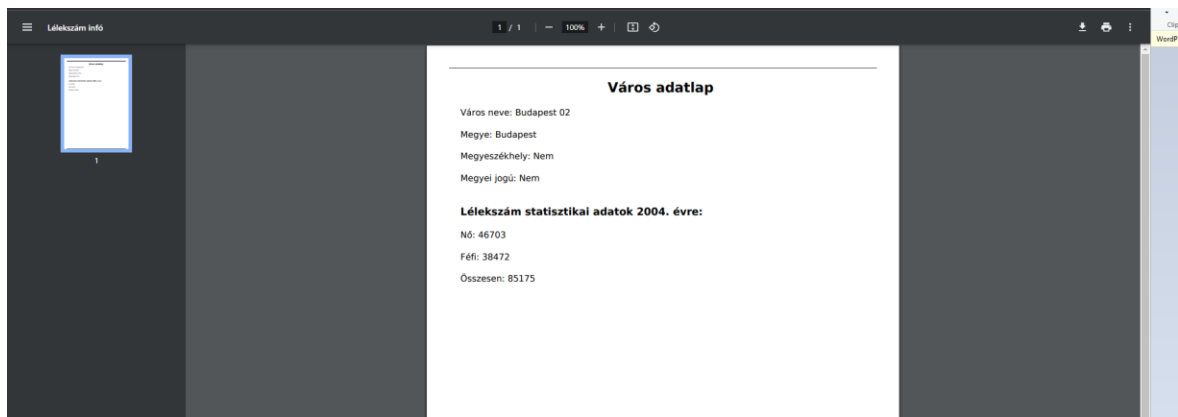
13.ábra



**5.feladat:** Az egyik oldalon készítsen egy PDF készítő szolgáltatást a **TCPDF** segítségével. A felhasználó 3 szöveges beviteli mezőben, vagy lenyíló listában megad adatokat és a rendszer az adatbázisból olvasva készít egy letölthető PDF fájlt. Olyan feladatot valósítson meg, amelyben az adatbázis mind a három táblájára szükség van.

A már korábban bemutatott Ajax megvalósítás mellé használtuk fel a TCPDF segítségével a PDF készítő szolgáltatást. A korábban bemutatott keresés eredményét (ld. 6-7.ábra) tudjuk exportálni egy PDF kiterjesztésű állományba, amely létrehoz egy “Város adatlapot” a városhoz tartozó adatokkal.

14.ábra



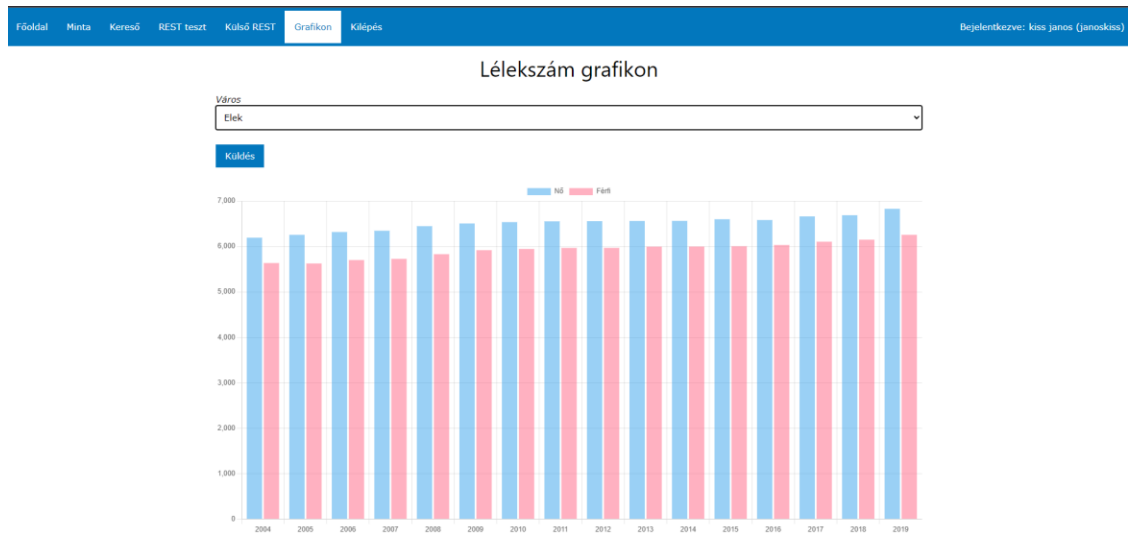
**6.feladat:** Alkalmazzon egy oldalon egy objektum orientált JavaScript megvalósítást, osztálydeklarációval (class)

Az összes olyan oldal ahol szükséges objektum-orientált JavaScript megvalósítással készült el amely tartalmaz osztálydeklarációt.

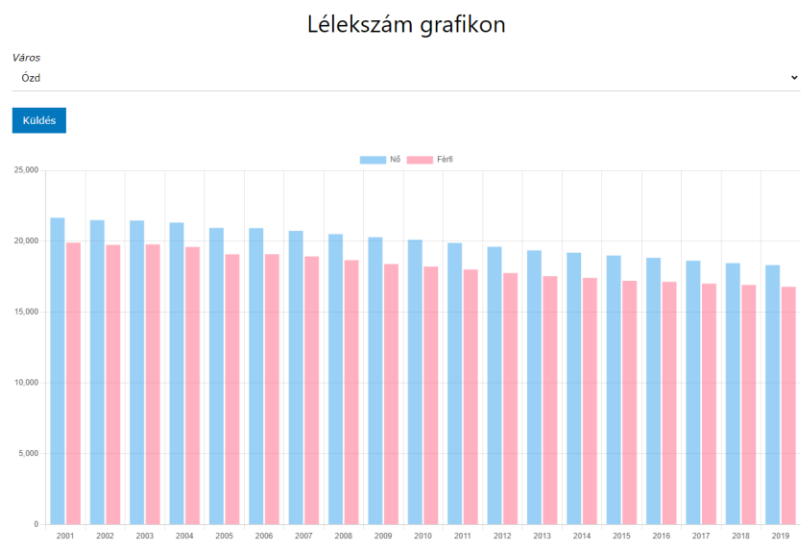


**7.feladat:** Az egyik oldalon készítsen egy grafikont az adatbázis adatai alapján.

15.ábra



16.ábra



A 15-16.ábrán látható a megvalósított grafikon melyhez a [chartjs.org](https://chartjs.org) weboldalt használtuk. A grafikon az adatbázis adatai alapján megmutatja nekünk a kiválasztott Város lélekszámának eloszlását Férfi/Női arányban. A várost szintén lenyíló listából tudjuk kiválasztani, a két ábrán két külön város adatai szerepelnek.

**7. feladat:** Alkalmazását tölts fel és valósítsa meg Internetes tárhelyen is.

Az elkészült feladatot a nethely.hu internetes tárhelyre töltöttük fel <http://varosinfo.nhely.hu/feladat2/nyitolap> néven.

**8.feladat:** Használják a GitHub (github.com) verziókövető rendszert

A feladathoz tartozó repository neve:

[https://github.com/pagerdani/dkkg9\\_at3le6\\_beadando\\_masodik\\_feladat](https://github.com/pagerdani/dkkg9_at3le6_beadando_masodik_feladat)

**9. feladat:** Készítsen egy legalább 10 oldalas dokumentációt képernyőképekkel

(Kötelező elem!),

A feladatot megvalósította:

- Páger Dániel (AT3LE6)
- Szauter András Gábor (DKKGH9)