

SZAKDOLGOZAT

Szakképesítés neve: Szoftverfejlesztő és -tesztelő technikus

OKJ száma: 5-0613-12-03

Ytravel

Kiss Csongor Dominik

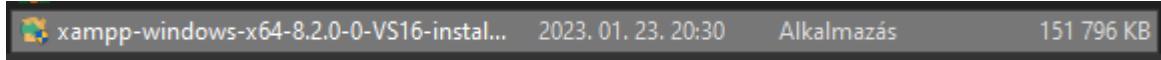
Rusai Márk, Vecsernyés András

Xampp telepítése

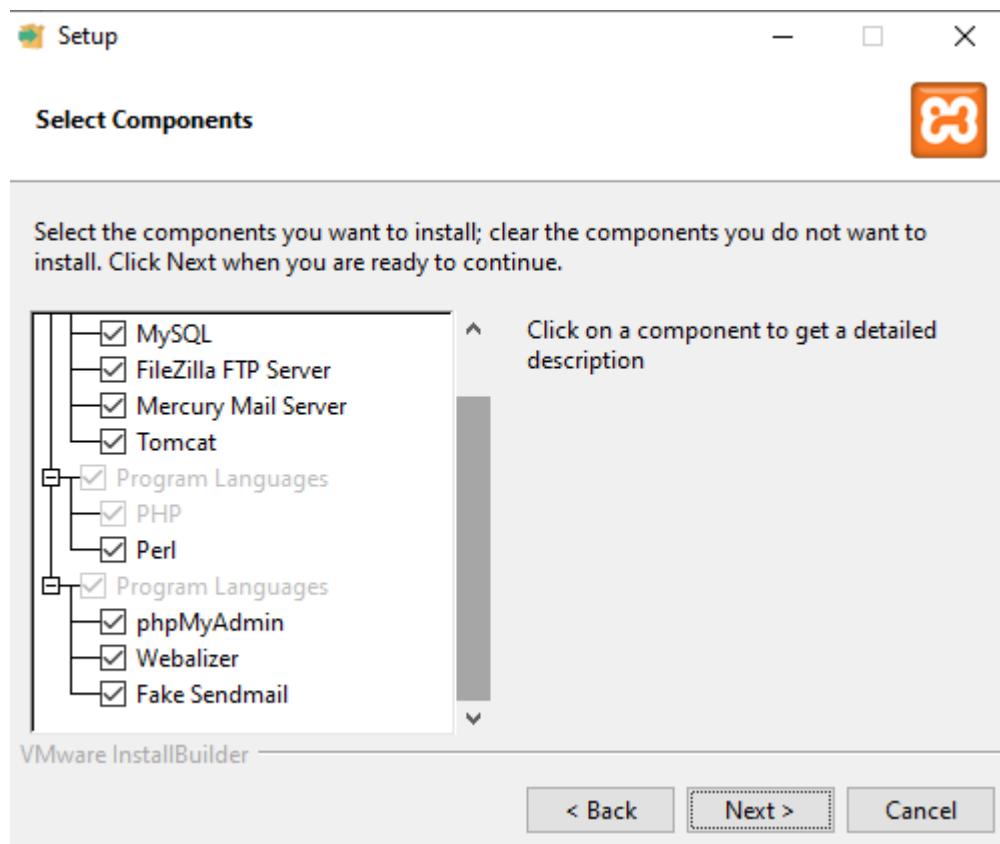
A weboldal megnyitásához szükségünk lesz a localhost létrehozására, mi axampp-ot használtuk a localhost létrehozásához. Első lépésként ha esetleg nincs feltelepítve axampp akkor a következő linkről lehet letölteni: <https://www.apachefriends.org/hu/index.html>. A link lekattintása után ez az oldal jelenik meg előttünk:

The screenshot shows the official XAMPP download page. At the top, there's a large orange XAMPP logo with the text "XAMPP Apache + MariaDB + PHP + Perl". Below it, a section titled "Mi a XAMPP?" provides a brief overview. To the right is a large orange square containing the XAMPP logo and the word "XAMPP" below it. At the bottom left, a green arrow-shaped button says "Letöltés" (Download) with the sub-instruction "Más verziókhöz kattintson ide". To the right are three download links: "XAMPP Windows rendszerre 8.2.4 (PHP 8.2.4)", "XAMPP Linux rendszerre 8.2.4 (PHP 8.2.4)", and "XAMPP OS X rendszerre 8.2.4 (PHP 8.2.4)".

Miután axampp telepítője sikeresen letöltött, nyissuk meg a telepítőt.

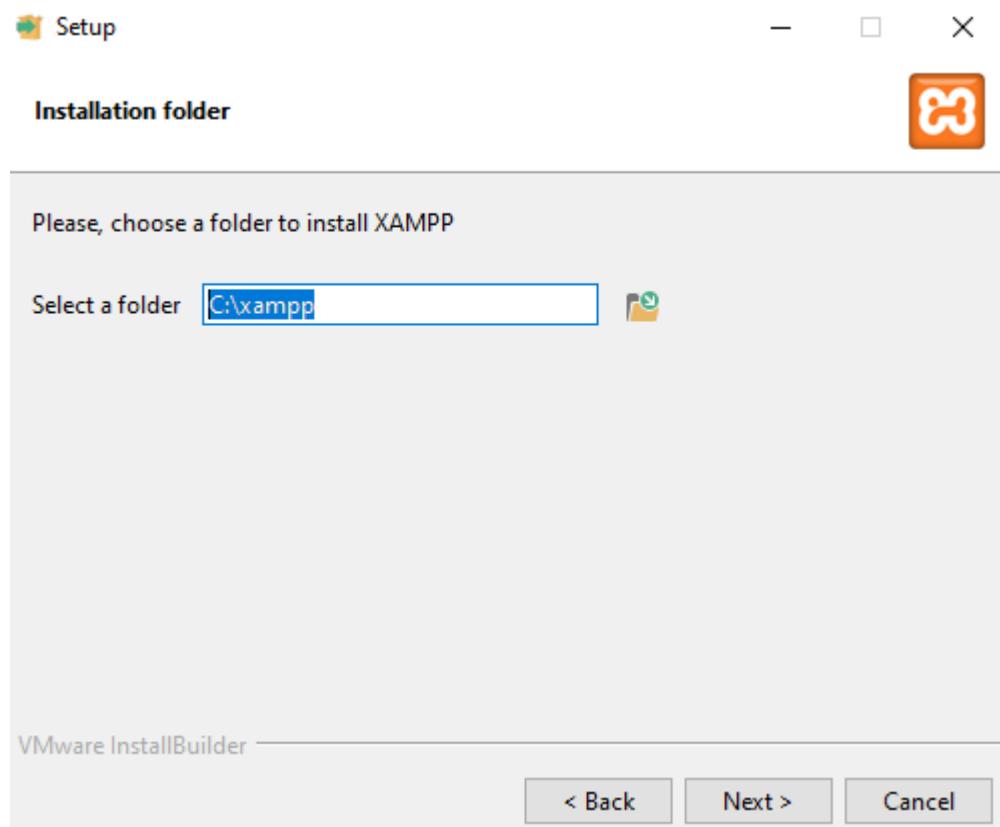


A telepítőn keresztül kiválaszthatjuk, mely szolgáltatásokat szeretnénk telepíteni az apache-al együtt (az apache nem választható szolgáltatás, hiszen erre mindenki által szükség van).



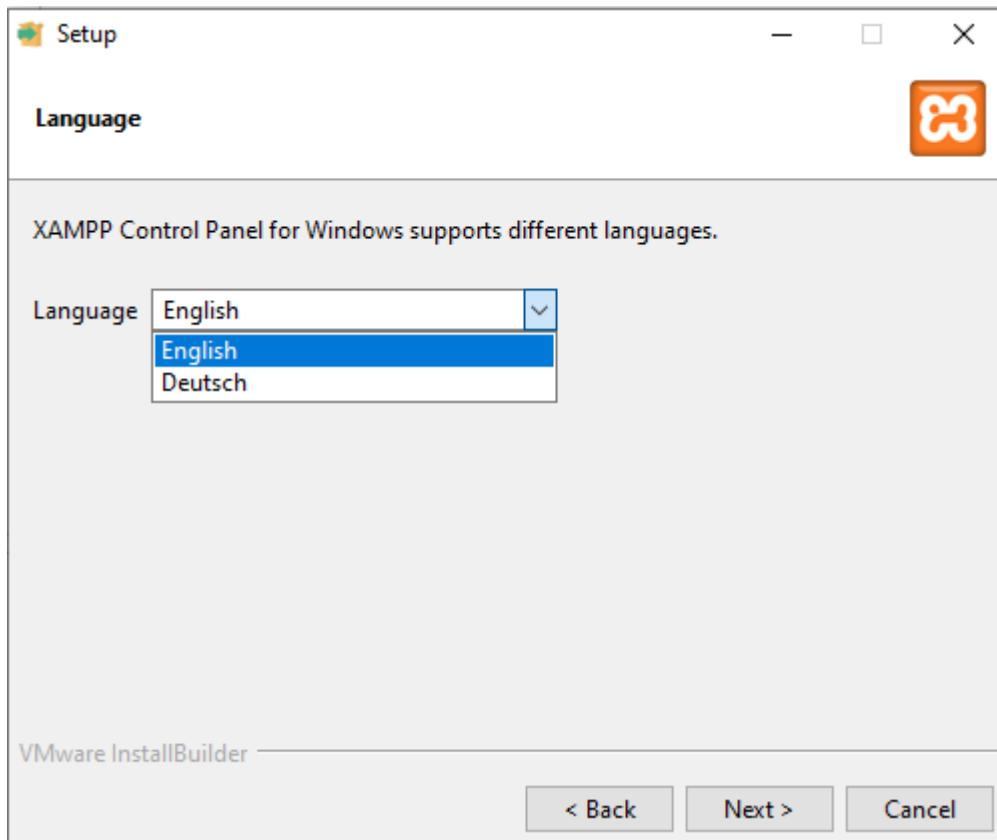
Javasolt az összes opciót kiválasztani, hiszen nem tudhatjuk, hogy a jövőben melyik szolgáltatásokat fogjuk igénybe venni.

Ezt követően a “next” gombra kattintva kiválaszthatjuk hova szeretnénk telepíteni a xampp kezelőfelületét.

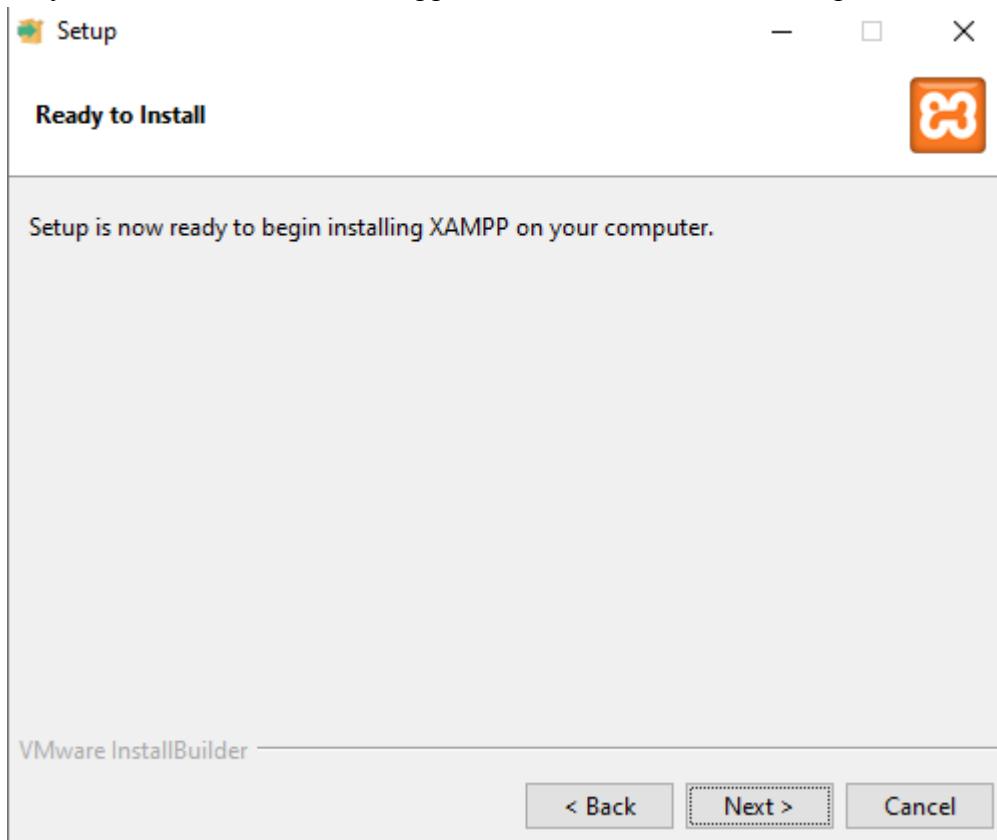


Ezután a kezelőfelület

nyelvét választhatjuk ki.



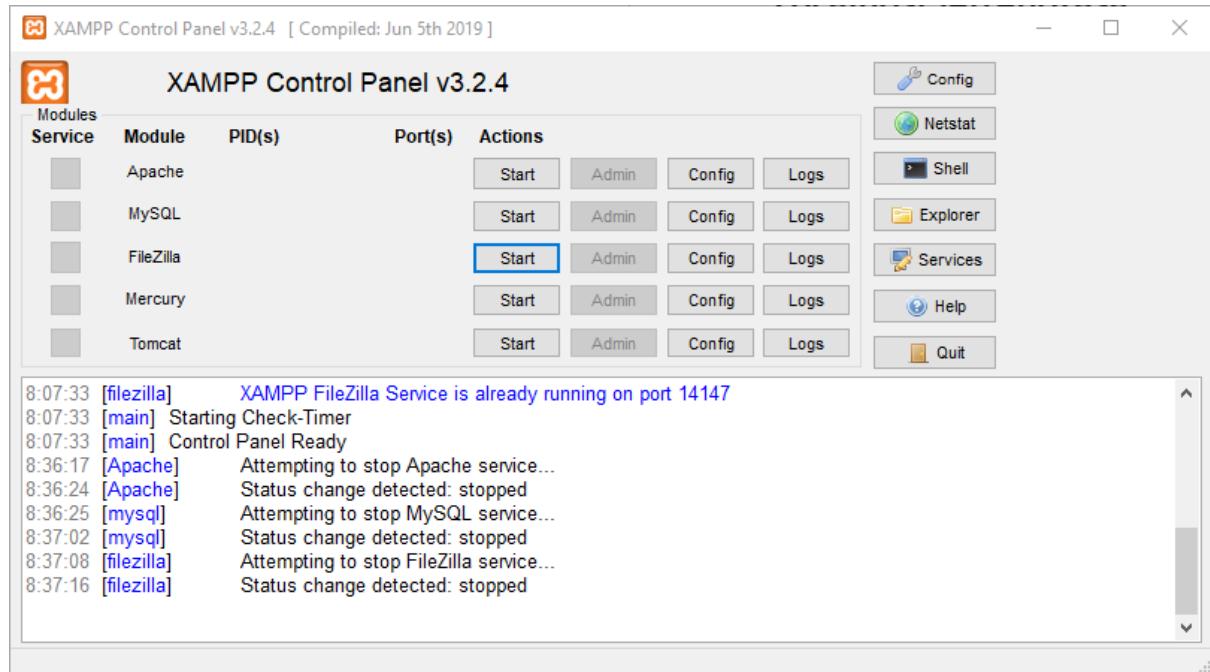
A nyelv kiválasztása után a xampp kezelőfelülete már kész a telepítésre.



A “next” gomb lekattintása után indul a telepítés. Utána lehet létrehozni a localhost-okat.

localhost létrehozása

A telepítés után megjelenik axampp kezelőfelülete.



A localhost-ot két módon is létre lehet hozni. Az első, hogy a repository-t a htdocs könyvtárba klónozzuk, illetve a másik lehetőség egy axampp-tól független könyvtárba klónozzuk a projektet. Ahhoz hogy a htdocs könyvtárat elérjük, axampp kezelőfelületén található “explorer” gombra kell kattintanunk. A gomb megnyitja a fájlkezelőt, azon belül is

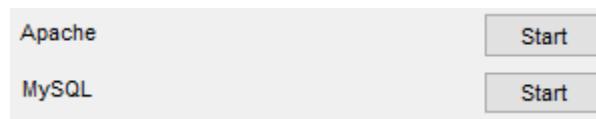
a xampp fájljait láthatjuk.

Név	Módosítás dátuma	Típus	Méret
anonymous	2020. 10. 28. 11:14	Fájlmappa	
apache	2020. 10. 28. 11:14	Fájlmappa	
cgi-bin	2020. 10. 28. 11:18	Fájlmappa	
contrib	2020. 10. 28. 11:14	Fájlmappa	
FileZillaFTP	2020. 10. 28. 11:18	Fájlmappa	
htdocs	2020. 10. 28. 11:14	Fájlmappa	
img	2020. 10. 28. 11:13	Fájlmappa	
install	2020. 10. 28. 11:18	Fájlmappa	
licenses	2020. 10. 28. 11:13	Fájlmappa	
locale	2020. 10. 28. 11:13	Fájlmappa	
mailoutput	2020. 10. 28. 11:13	Fájlmappa	
mailtodisk	2020. 10. 28. 11:14	Fájlmappa	
MercuryMail	2022. 09. 26. 8:12	Fájlmappa	
mysql	2020. 10. 28. 11:14	Fájlmappa	
perl	2020. 10. 28. 11:15	Fájlmappa	
php	2023. 04. 03. 13:42	Fájlmappa	
phpMyAdmin	2022. 09. 07. 13:00	Fájlmappa	
sendmail	2020. 10. 28. 11:18	Fájlmappa	
src	2020. 10. 28. 11:13	Fájlmappa	
tmp	2023. 04. 12. 8:37	Fájlmappa	
tomcat	2020. 10. 28. 11:14	Fájlmappa	
webalizer	2020. 10. 28. 11:18	Fájlmappa	
webdav	2020. 10. 28. 11:14	Fájlmappa	
apache_start	2013. 06. 07. 13:15	Windows kötegfájl	1 KB
apache_stop	2020. 10. 28. 11:19	Windows kötegfájl	1 KB
catalina_service	2013. 03. 30. 13:29	Windows kötegfájl	10 KB
catalina_start	2019. 10. 22. 15:36	Windows kötegfájl	5 KB
catalina_stop	2019. 10. 22. 15:35	Windows kötegfájl	5 KB
ctlscrip	2020. 10. 28. 11:13	Windows kötegfájl	3 KB
filezilla_setup	2013. 03. 30. 13:29	Windows kötegfájl	1 KB
filezilla_start	2013. 06. 07. 13:15	Windows kötegfájl	1 KB
filezilla_stop	2013. 06. 07. 13:15	Windows kötegfájl	1 KB
killprocess	2019. 08. 27. 16:01	Windows kötegfájl	1 KB
mercury_start	2013. 06. 07. 13:15	Windows kötegfájl	1 KB
mercury_stop	2013. 06. 07. 13:15	Windows kötegfájl	1 KB

Ezen belül ha megnyitjuk a htdocs könyvtárat, akkor a benne lévő fájlokat töröljük, majd csak ez után klónozzuk a projektet. Ezeket a fájlokat kell törölnünk:

Helyi lemez (C:) > xampp > htdocs			
Név	Módosítás dátuma	Típus	Méret
dashboard	2020. 10. 28. 11:14	Fájlmappa	
img	2020. 10. 28. 11:14	Fájlmappa	
webalizer	2020. 10. 28. 11:13	Fájlmappa	
xampp	2020. 10. 28. 11:14	Fájlmappa	
applications	2019. 08. 27. 16:02	Firefox HTML Doc...	4 KB
bitnami	2019. 08. 27. 16:02	CSS forrásfájl	1 KB
favicon	2015. 07. 16. 17:32	Ikon	31 KB
index	2015. 07. 16. 17:32	PHP forrásfájl	1 KB

Ha minden lépés sikeresen megtörtént, akkor el kell indítanunk a kezelőfelületen az apache és mysql szolgáltatásokat a “start” gombra kattintva.



Miután hiba nélkül elindultak a szolgáltatások, a böngésző URL sávjában egyszerűen csak beírjuk azt, hogy “localhost”, vagy az “admin” gombra kattintunk az “apache” szolgáltatás sorában.

A screenshot of the XAMPP Admin interface. It's a table with columns: Module, PID(S), Port(S), and Actions. The Apache row has a PID of 4188, port 80, 443, and buttons for Stop and Admin. The MySQL row has a PID of 964, port 3306, and buttons for Stop and Admin. The Admin button for Apache is highlighted with a blue border. Below the table is a browser address bar with "localhost" typed into it.

Ha minden jól ment, akkor ezt a felületet kell látnunk:

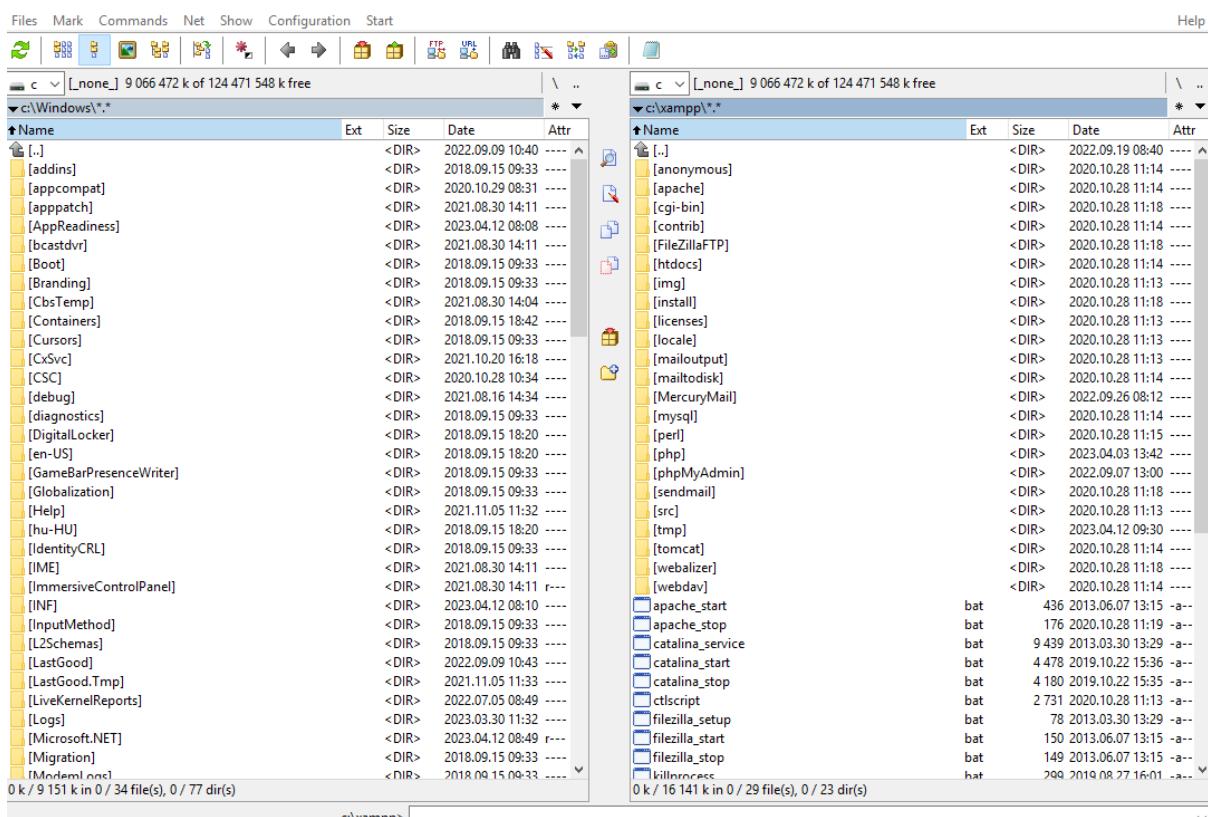


Index of /

Name	Last modified	Size	Description
evVegiProjekt/	2023-03-01 19:25	-	

Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0 Server at localhost Port 80

A localhost létrehozása egy különálló könyvtárban már egy fokkal bonyolultabb. A Total Commander használata javasolt az átláthatóság végett. Total Commander-ben meg kell nyitnunk a "xampp", illetve "windows" könyvtárakat.



Telepítéstől függően változhatnak az útvonalak, de ezek az általános útvonalak a szükséges fájlokhoz:

- HTTPD: xampp > apache > conf > httpd
- v-hosts: xampp > apache > conf > extra > vhosts

- hosts windows > system32 > drivers > etc > hosts

Fontos, hogy minden fájlt jegyzettömben nyissunk meg!

Első lépésünk az az engedély megadása a független könyvtárnak a HTTPD fájlban, az én esetben ez a “www” könyvtár lesz. A fájlon belül nagyjából a 279. sor körül kell létrehoznunk az engedély megadást a következő képpen:

```
<Directory "c:/www">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>
```

A Directory tagjébe megy a projekt könyvtárának elérési útvonala, ha ezzel megvagyunk, akkor a v-hosts fájlban kell megadnunk a virtual hostot.

```
<VirtualHost *:80>
    ServerName tura.loc
    DocumentRoot "C:/www/evVegiProjekt"
</VirtualHost>
```

Itt VirtualHost taggel kell dolgoznunk, a tag-en belül a 80-as portot használjuk, de ez lehet akár 8080, vagy esetleg a 443-as port is. A “ServerName” sorhoz adjuk meg a szerver nevét, illetve az alatta lévő sorban pedig a projektünk elérési útvonala található. Ezek után a hosts fájlból majd hozzáfűzni az IP címet a szerverünkhöz a következő módon:

```
127.0.0.1      aruhaz.loc
127.0.0.1      forum.loc
127.0.0.1      fbfeladat.loc
127.0.0.1      email.loc
127.0.0.1      face.loc
127.0.0.1      tura.loc
127.0.0.1      kereso.loc
127.0.0.1      restapi.loc
```

Ha mentettük a fájlok tartalmát, akkor a böngésző URL sávjában meg adjuk a szerver nevét, ami átvezet a weboldalra.

<http://tura.loc>

Adatbázis létrehozása

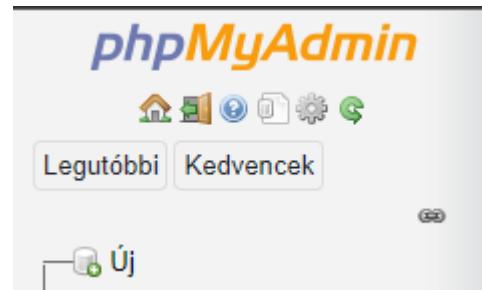
Az adatbázis létrehozásához futnia kell a korábban már elindított szolgáltatásoknak(apache, mysql), a phpmyadmin oldalát az “admin” gombbal érhetjük el a mysql sorában .

Apache	4188	80, 443	Stop	Admin
MySQL	964	3306	Stop	Admin

A gomb lenyomása után a phpmyadmin oldala jelenik meg.

The screenshot shows the phpMyAdmin login interface. On the left, there's a sidebar with a tree view of databases: 'gutóbbi' (Recent), 'Adatbázisok' (Databases), 'SQL', 'Állapot' (Status), 'Felhasználói fiókok' (User Accounts), 'Exportálás' (Export), 'Importálás' (Import), 'Beállítások' (Settings), 'Tobbszörözés' (Multiple), 'Változók' (Variables), 'Karakterkészlet' (Character Set), 'Motorok' (Engines), and 'Bővítmények' (Extensions). The main area has tabs for 'Általános beállítások' (General settings) and 'Megjelenési beállítások' (Display settings). The right side contains sections for 'Adatbázis-kiszolgáló' (Database server), 'Webkiszolgáló' (Web server), and 'phpMyAdmin' (phpMyAdmin version information). At the bottom, there's a message about a missing configuration file and a note about the current version.

Egy adatbázist kétféle módon lehet létrehozni, grafikusan és kód alapján. A grafikus

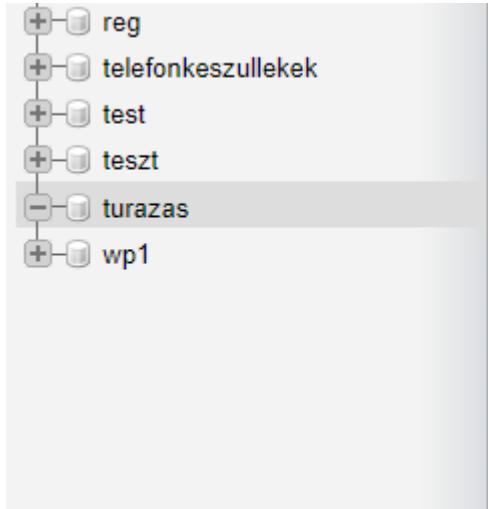


megoldást fogom megmutatni.

Az "új" gombra kattintva a következő jelenik meg:

The screenshot shows the 'Adatbázis létrehozása' (Create Database) form. It includes fields for 'Turazas' (Name) set to 'utf8_hungarian_ci', a dropdown for 'Létrehozás' (Create), and a 'Szűrők' (Filters) section with a search input. Below is a 'Tartalmazza ezt a szót:' (Contains this word:) input field. The main table lists databases with their character sets and collations, including 'adat' (utf32_hungarian_ci) and 'adatbazis_loremipsum' (utf8mb4_general_ci).

Itt megadjuk az adatbázis nevét (Turazas) és kódolását (utf8_hungarian_ci), majd a "létrehozás" gombbal létrehozzuk az adatbázist.

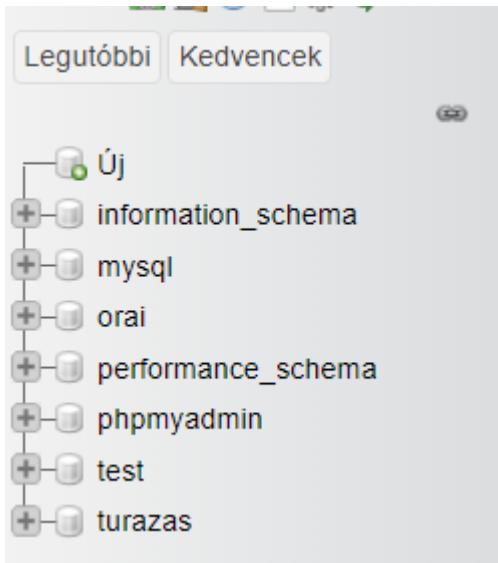


A táblák és rekordok létrehozásához a könyvtárban található turazas.sql fájlt kell importálni.

📁	.git	2023. 04. 11. 20:38	Fájlmappa
📁	components	2023. 04. 11. 20:37	Fájlmappa
📁	frontend	2023. 04. 11. 20:37	Fájlmappa
📁	kepek	2023. 03. 19. 20:19	Fájlmappa
📁	kepektura	2023. 04. 11. 20:38	Fájlmappa
📁	server	2023. 04. 11. 20:38	Fájlmappa
📁	src	2023. 04. 11. 20:38	Fájlmappa
📁	tests	2023. 04. 11. 20:38	Fájlmappa
📄	.gitignore	2023. 04. 11. 20:37	Szöveges dokume...
📄	composer	2023. 04. 11. 20:37	JSON forrásfájl
📄	composer.lock	2023. 04. 11. 20:37	LOCK fájl
📄	index	2023. 04. 11. 20:37	PHP forrásfájl
📄	style	2023. 04. 11. 20:38	CSS fájl
📄	turazas	2023. 04. 11. 20:38	SQL forrásfájl

Ezt a phpmyadmin felületén az importálás fülön lehet elvégezni a következő lépésekben:

- Kattintsa le az újonnan létrehozott “turazas” adatbázist



- Majd az oldal felső navigációs menüjében válassza ki az “importálás” fület:



- A menüponton belül kattintson a “Fájl kiválasztása” gombra:
Importálás a(z) "turazas" adatbázisba

The screenshot shows the 'Importálás a(z) "turazas" adatbázisba' dialog window. It includes the following fields and instructions:

- Az importálandó fájl:** (File to import)
- A fájl lehet tömörített (gzip, bzip2) vagy tömörítetlen.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**
- Számítógép tallózása: (Max: 40MB)
File upload location: (Max: 40MB)
- Fájl kiválasztása (Select file) and Nincs fájl kiválasztva (No file selected).
- Továbbá behúzhat és beeíthet egy fájlt bármely oldalra.
You can also drag and drop a file onto any page.
- A fájl karakterkészlete: (Character set of the file)
Character set: utf-8

- Keresse ki a turazas.sql fájlt, majd kattintson a “megnyitás” gombra, ez után ezt kell látnia:

Importálás a(z) "turazas" adatbázisba

Az importálandó fájl:

A fájl lehet tömörített (gzip, bzip2) vagy tömörítetlen.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Számítógép tallózása: (Max: 40MB)

Fájl kiválasztása turazas.sql

Továbbá behúzhat és beejthet egy fájlt bármely oldalra.

A fájl karakterkészlete:

utf-8

- Végül az oldal alján található “importálás” gombra kattintva a rekordok felkerülnek az adatbázisba:

Formátum

SQL

Formátum specifikus opciók:

SQL kompatibilitási mód:

NONE

Ne használja az AUTO_INCREMENT-et nulla értékekhez

Importálás

Ha az importálás sikeres volt akkor ezt az üzenetet kell látnia:

 Az importálás sikeresen befejeződött, 16 lekérdezés került végrehajtásra. (turazas.sql)

Ezzel a rekordok létrejöttek az adatbázisban.

Az oldal funkciója és bemutatása

Az Ytravel egy túra kereső weboldal. Csináltunk egy tűrásáról szóló oldalt, ami segíti az érdeklődőket eligazodni Magyarország vármegyéi között. Egy egyszerűen kezelhető felhasználó barát weboldalt írtunk, ami a modern weboldalakra hasonlít de mégis a természetből lett inspirálva az oldal. Az oldalon a felhasználó regisztrálhat és ott kiválaszthatja a 4 db alap képből azt a képet ami neki tetszik. Belépés után a felhasználó elmentheti a túrákat a saját profiljába.

Az oldal kinézete

Az Ytravel weboldal háttere egy mesterséges intelligencia segítségével lett megcsinálva.

The screenshot shows the homepage of the Ytravel website. At the top, there's a navigation bar with the Ytravel logo, search bar, and links for 'Főoldal', 'Túrák', 'Információ', 'Profil', 'Üdv Mark!', 'Kijelentkezés', and a user profile icon. Below the header, a large banner features the text 'Találd meg kedvenc túra útvonalad.' and a 'Felfedezés' button. Three travel destination cards are displayed below the banner:

- Bács-Kiskun vármegye**: A landscape image of a field under a cloudy sky. Description: Bács-Kiskun vármegye Magyarország déli részén található, és gazdag természeti kincsekben, változatos tájban, sűrűn telepített és dombokban gazdag terület.
- Baranya vármegye**: A landscape image of hills covered in green vegetation. Description: Baranya vármegye Magyarország déli-délnyugati részén található, a Dél-Dunántúli régióban. A terület gazdag természeti kincseiben, változatos tájban, hegyekben, völgyekben, erdőkben és folyókban.
- Békés vármegye**: An aerial view of a town. Description: Békés vármegye Magyarország délkeleti részén található, a Dél-Alföldön. A területen számos érdekes természeti látványt található.

At the bottom of the page, there's a footer with links for 'Kapcsolatok', phone number '+11 111 111 1111', email 'Ytravel@gmail.com', a message saying 'Köszönjük, hogy a mi weboldalunkat választottál!', the year 'Ytravel © 2023-2028', and social media icons for Facebook, Instagram, and Twitter.

A logó egyszerű modern stílusban lett elkészítve.

Bejelentkezés, regisztráció

Regisztráció felhasználói útmutató

A regisztrációs oldal a felhasználótól három darab bemenetet kér (név, jelszó, felhasználói kép), ezeknek kitöltése kötelező a sikeres regisztrációhoz.

Regisztráció

Név:

Jelszó:

Jelszó újra:

Felhasználói kép:

1 ▾

© 2023–2028

Regisztráció

[Vissza a belépéshoz!](#)



Sikeres Regisztráció!

Jelentkezzen be!

OK

A sikeres regisztráció után a felhasználó egy feljövő ablakból tudja hogy az adat amit beírt az megfelelő.



Már van ilyen felhasználó!

Regisztráljon másik névvel!

OK

Ha az adatbázisban létezik a név akkor nem engedi regisztrálni a felhasználót.

Regisztráció fejlesztői dokumentáció

A regisztráció a regisztracio.php-ról történik meg. A név és a jelszó beírása input mezővel történik meg. A kép kiválasztása egy select legördülő lista ahol ki lehet választani 1-4 ig a felhasználói képet.

```

$regiszt = new Regisztralas();
if(isset($_POST["regiszt2"]))
{
    if(isset($_POST["nev2"])&&isset($_POST["jelszocska"])&&isset($_POST["jelszocska2"]))
    {
        echo '<script type="text/javascript">

$(document).ready(function(){

    Swal.fire(
        "Sikeres Regisztráció!",
        "Jelentkezzen be!",
        "success"
    )
})
</script>
';
$regiszt ->Regist($_POST["nev2"],$_POST["jelszocska"],$_POST["jelszocska2"],$_POST["pfp2"]);
    }
}

```

A regisztráció gombra kattintva egy if(isset()) -el megnézzük hogy be írt a felhasználó minden adatot. Ha igen akkor a felhasználó kap egy sweatalertet hogy sikeres a regisztráció. Itt meghívjuk a Regisztrálás osztályt és azon belül a Regist functiont.

```

$nevcheck = $this->csatlakozas->query("SELECT * from felhasznalok where nev = '" . $nev . "'");
if ($adat = $nevcheck->fetch_assoc())
{
    echo '<script type="text/javascript">

$(document).ready(function(){

    Swal.fire(
        "Már van ilyen felhasználó!",
        "Regisztráljon másik névvel!",
        "error"
    )
})
</script>
';
}

```

Egy egyszerű sql lekérdezéssel megnézzük hogy van-e ilyen felhasználó név. Ha van akkor nem engedi a regisztrálást.

```

else {
    if ($jelszo == $jelszo2)
    {
        $regiszralas = $this->csatlakozas->query("INSERT INTO felhasznalok (nev,jelszo, user_kep_id)
values('" .nev. "','" .jelszo. "','" .pfp.')");
    } else
    {
        echo '<script type="text/javascript">

$(document).ready(function(){

    Swal.fire(
        "Nem egyezik meg a két jelszó!!",
        "Próbálja újra!",
        "error"
    )
})</script>
';
    }
}

```

Utána megnézzük hogy a két jelszó egyezik-e ,ha igen akkor a felhasználok táblába feltöljük a felhasználó adatait. Az else ágba akkor lépünk be ha a két jelszó nem egyezik és ilyenkor egy sweetalertal közöljük ezt a felhasználóval.

Bejelentkezés felhasználói útmutató

A bejelenkező oldal a felhasználótól kettő darab bemenetet kér (nevet , jelszót),ezeknek kitöltése kötelező a sikeres bejelentkezéshez.

Belépés

Név:

Jelszó:

Belépés

Ytravel © 2023-2028

Regisztráció

[Vissza a főoldalra](#)

Ha a felhasználó már létező név és jelszó kombinációt ad meg akkor a bejelentkezés sikeres volt, és a tovább lesz irányítva a főoldalra.

Ha a felhasználó nem létező kombinációt ad meg, akkor az oldal értesíti erről.



Nem megfelelő adatokkal próbál belépni!

Próbálja újra!

OK

Bejelentkezés fejlesztői dokumentáció

A bejelentkezés oldal a felhasználótól három darab bemenetet kér (név, jelszó, felhasználói kép), ezeknek kitöltése kötelező a sikeres belépéshez.

A bejelentkezés a belepes.php-ról történik meg. A név és a jelszó beírása input mezővel történik meg. A belépés gombra kattintva a form post változókat elküldi.

Ha rá kattint a felhasználó a regisztráció gombra akkor átküldi a regisztracio.php-ra.

```
if(isset($_POST["belep"]))
{
    $belepes = new Belepes();
    if(isset($_POST["nev"])&&isset($_POST["jelszo"]))
    {
        $belepes ->Login($_POST["nev"],$_POST["jelszo"]);
    }
    else{
        echo '<script type="text/javascript">
$(document).ready(function(){

    Swal.fire(
        "Nem megfelelő adatokkal próbál belépni!",
        "Próbálja újra!",
        "error"
    )
})</script>
';
    }
}
```

Egy If(isset)-el megnézzük hogy a felhasználó rendesen kitöltötte a belépés felületet , ha nem akkor értesítjük a felhasználót egy sweetalerttel. Ha a név és a jelszó ki van töltve akkor meghívjuk a Belépés osztályt és azon belül a Login functiont.

```

function Login($nev, $jelszo)
{
    $belepes = $this->csatlakozas->query("SELECT * from felhasznalok
    where nev = '" . $nev . "' and jelszo = '" . $jelszo . "' ");
    if ($adat = $belepes->fetch_assoc()) {
        $_SESSION["nev"] = $adat['nev'];

        $_SESSION["id"] = $adat["id"];
        $_SESSION["profilkep"] = $adat["user_kep_id"];
        header("Location: ./");
    } else {
        echo '<script type="text/javascript">

            $(document).ready(function(){

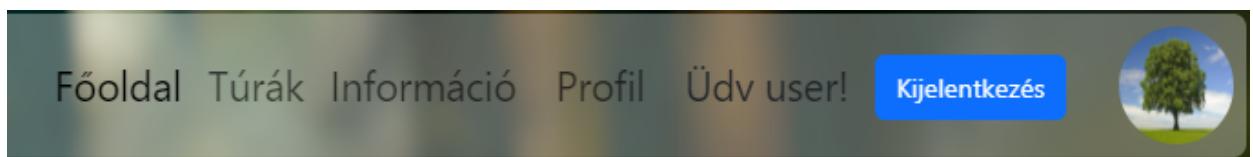
                Swal.fire(
                    "Nem megfelelő adatokkal próbál belépní!",
                    "Próbálja újra!",
                    "error"
                )
            })
        </script>
    ';
}
}

```

Egy sql lelkendezéssel megnézzük hogy létezik-e ezekkel az adatokkal felhasználó, ha igen akkor a SESSION[id]-t , SESSION[nev]-et és a SESSION[profilkep]-et a felhasználónak az idjével, nevével és profilképével tesszük egyelővé. Ezután átirányítjuk a felhasználót a főoldalra.

Az else ágba akkor lép be ha nem megfelelő az adatok. Ezt a felhasználóval közöljük egy sweetalertel.

Saját profil felhasználói útmutató



A felhasználó belépés után megtekintheti a saját profilját. Ezt úgy tudja ha rákattol a profil menü pontra vagy a profilképére.

The screenshot shows two overlapping windows. The left window is titled 'Profil szerkesztése' (Profile editing) and contains fields for 'Név:' (Name) with 'user' typed in, 'Jelszó:' (Password) with '123' typed in, and 'Profilkép:' (Profile picture) with a dropdown menu showing '1'. A blue 'Szerkesztés' (Edit) button is at the bottom. The right window is titled 'Mentett túrák' (Saved trips) and lists four items, each with a red 'x' button to delete:

- Sombokor-kör Kékes hegység
- Pusztatemplom romjai
- Hegyestű Geológiai bemutatóhely
- Tisza-parti Tanösvény

Miután belépett a profil felületre a felhasználó át tudja írni a nevét, jelszavát és profilképét.

A mentet túrákat is látja a felhasználó és ezeket ki lehet törölni is.



Sikeres törlés!

OK

Túra törlése után a felhasználót az oldal értesíti elről.

Saját profil fejlesztői dokumentáció

Ha a felhasználó rá kattint a profil gombra vagy a saját profilképére akkor az oldal átirányítja a profil.php-ra.

```
$adminlekeres->ProfilKiiras();
```

Az oldal betöltésekor meghívjuk az FelhasznaloiProfil osztálynak a ProfilKiiras funkcióját.

```
$felhasznalokiiras=$this->csatlakozas->query("SELECT * from felhasznalok  
where nev = '". $_SESSION["nev"]. "'");  
if($adat = $felhasznalokiiras->fetch_assoc())
```

Utána egy sql lekérdezéssel megkeressük azokat az adatokat ahol a név egyenlő a SESSION[nev] -el. Majd ezután ki íratjuk a felhasználónak adatait és annak a szerkesztési lehetőségeit input mezőkkel.

```
if(isset($_POST["profilszerkeszes"]))  
{  
$adminlekeres->ProfilUpdate($_POST["nev3"],$_POST["jelszo3"],$_POST["kepek4"]);
```

Ha a felhasználó rákattint a Szerkesztés gombra akkor meghívjuk az FelhasznaloiProfil osztálynak a ProfilUpdate funkcióját.

```
$felhasznalokiiras=$this->csatlakozas->query("SELECT * from felhasznalok  
where nev = '". $nev."' and id != '". $_SESSION["id"]." ' ");  
if($adat = $felhasznalokiiras->fetch_assoc())
```

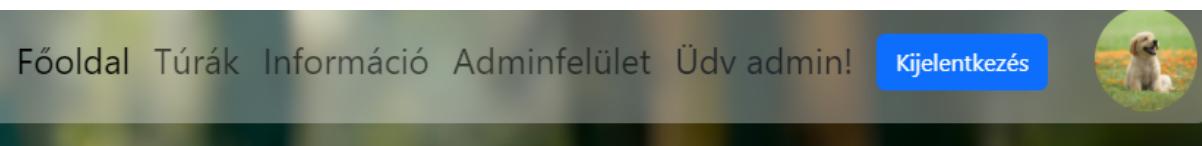
Egy sql lelkendezéssel megnézzük hogy van e ilyen név az adatbázisban, ha létezik akkor egy sweetalert köözöljük ezt a felhasználóval és nem engedjük átírni az adatokat.

```
$felhasnaloJavitás = $this->csatlakozas->query("UPDATE felhasznalok SET  
nev = '". $nev."', jelszo = '". $jelszo."', user_kep_id = '". $kepid."' WHERE id = '". $_SESSION["id"]." '");
```

Viszont ha nincs ilyen felhasználói név akkor updateljük a felhasználónak a nevet, jelszavát és profilképét a beírt adatokkal.

Admin felület felhasználói dokumentáció

Az admin felület "admin, admin" név és jelszó párossal érhető el.



Ekkor megjelenik egy külön menüpont, amin belül tudjuk törlni, szerkeszteni vagy hozzáadni felhasználókat és túrákat.



Üdvözöllek az admin felületen!

Válassz a menüpontok között az adatok változtatásához!

Az admin felületről vissza tudunk menni a főoldalra, amennyiben böngészni kívánunk tovább. Más esetben a felhasználók és a túrák menüpontra kattintva érhetjük el ezek szerkesztését.

Felhasználók szerkesztése

Felhasználók szerkesztése.

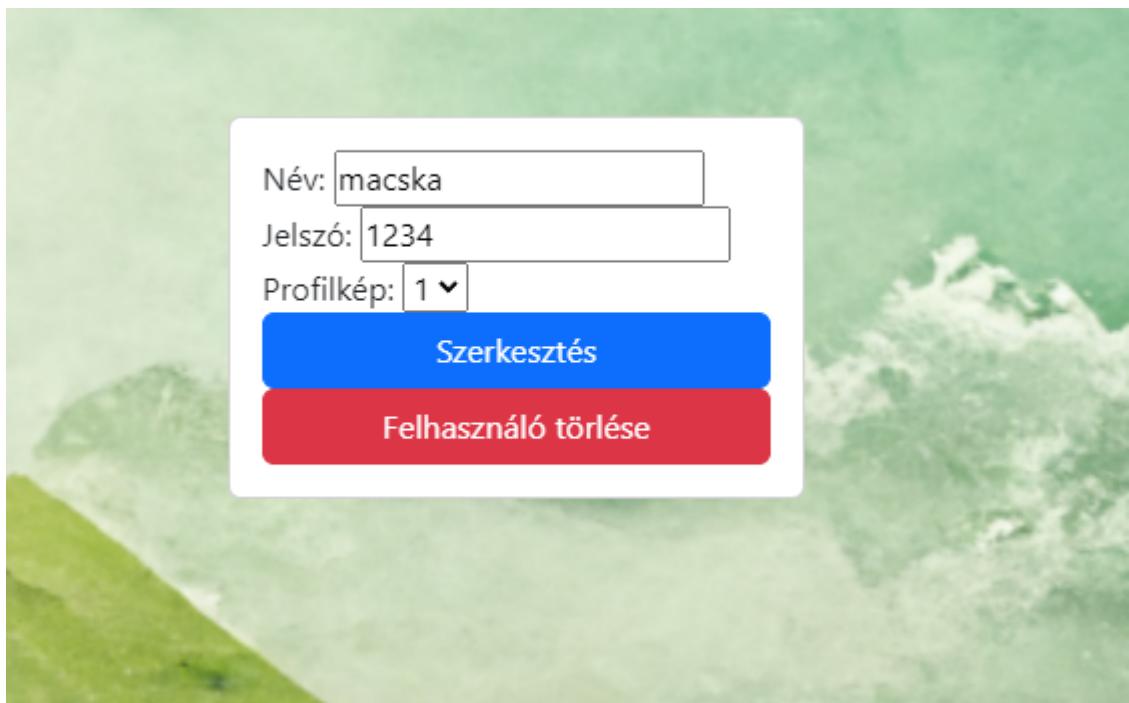
The screenshot shows a user management interface. At the top left, there is a blue button labeled "Felhasználó hozzáadása". Below it, there are two cards. The first card features a photo of a golden retriever puppy sitting in a field of orange flowers, with the name "macska" below it and a "Szerkesztés" button at the bottom. The second card features a photo of a brown goat standing on a wooden post, with the name "anti" below it and a "Szerkesztés" button at the bottom. The background of the interface is a green landscape with mountains.

Itt a “felhasználók hozzáadása” gombra nyomva elnavigál minket az oldalra, ahol adhatunk a nevet, jelszavat és profilképet.

Felhasználók szerkesztése.

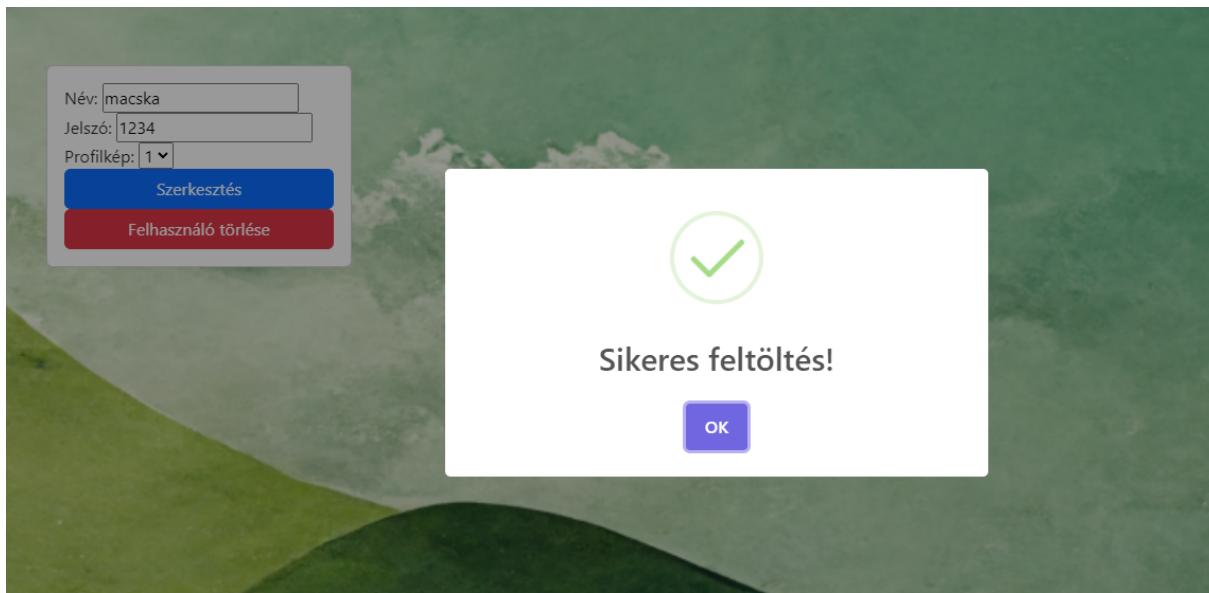
Név:
Jelszó:
Profilkép:

A meglévőket pedig ugyan ezen szisztema szerint változtathatjuk.(az oldalon történő változtatások megkövetelik az oldal frissítését vagy ki és újra bejelentkezést)

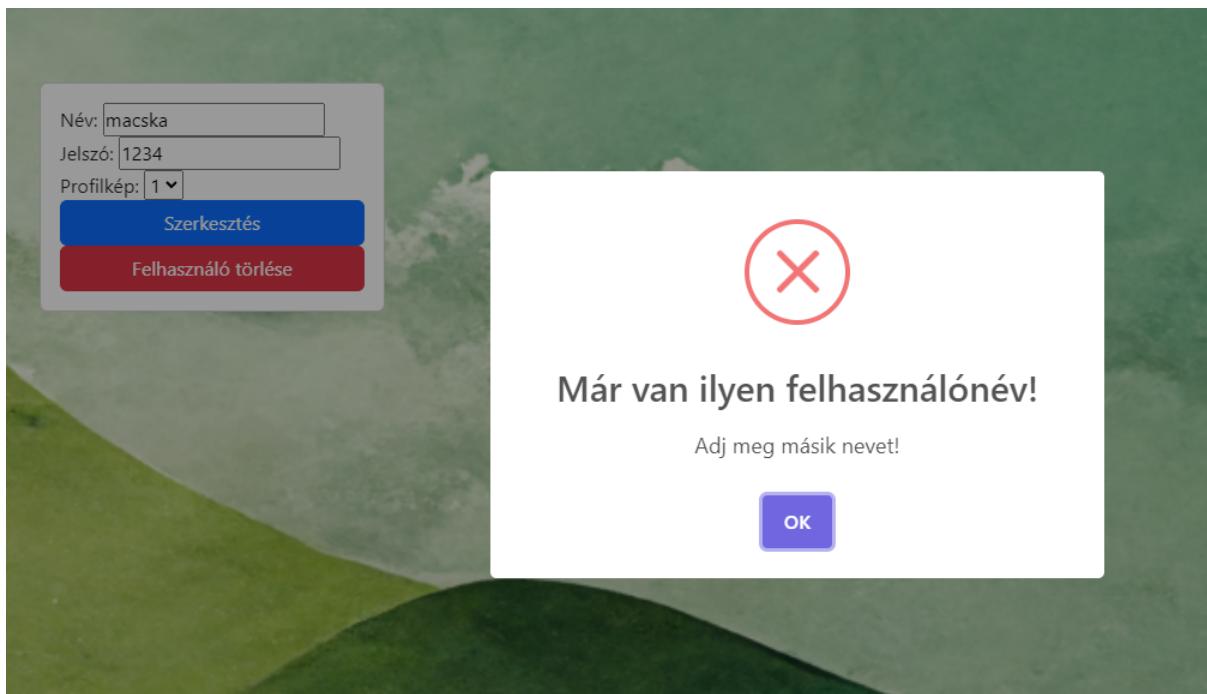


Természetesen ezen szerkesztéseket ellenőrizzük, tehát nem lehet már adatbázisban szereplő nevet megadni sem új, sem régi felhasználónak.

Amennyiben jó a szerkesztés:

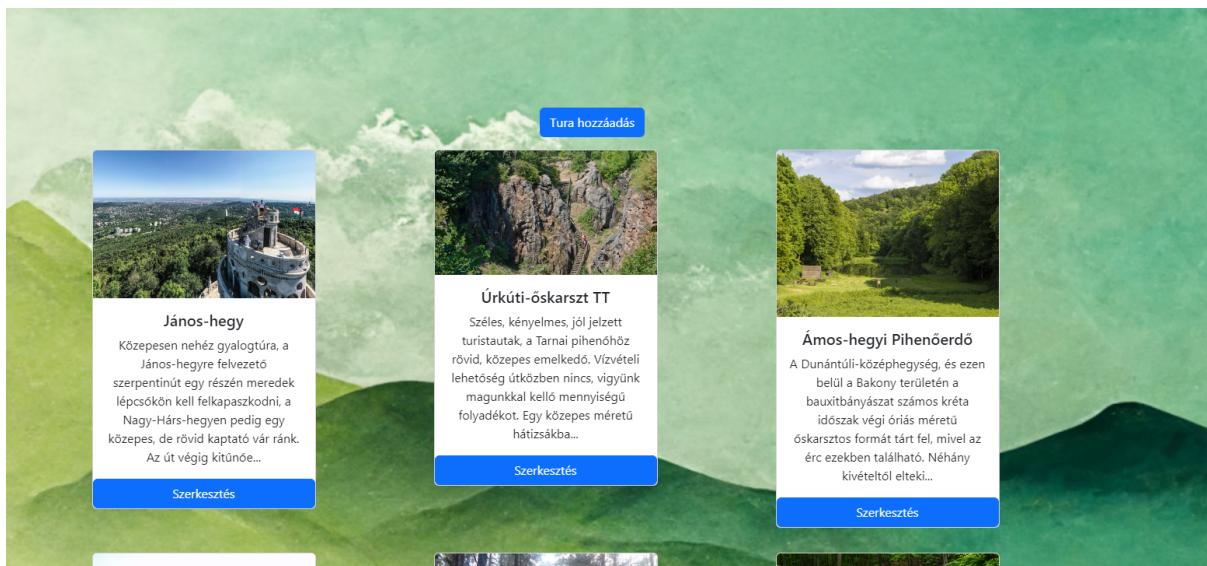


Amennyiben nem:



Túrák szerkesztése

Alap helyzetben megjelenik az összes túra és mindegyiknél egy szerkesztés gomb.



A "szerkesztés" gombra kattintva megjelenik minden adata az adott túrának amit lehet szerkeszteni vagy akár törölni az adott túrát.

Túra szerkesztés

Túra név szerkesztés:

János-hegy

Túra hossza:

11.1

Túra nehézsége:

2

Túra felkapottsága:

89

Megye id:

1

Szerkesztés

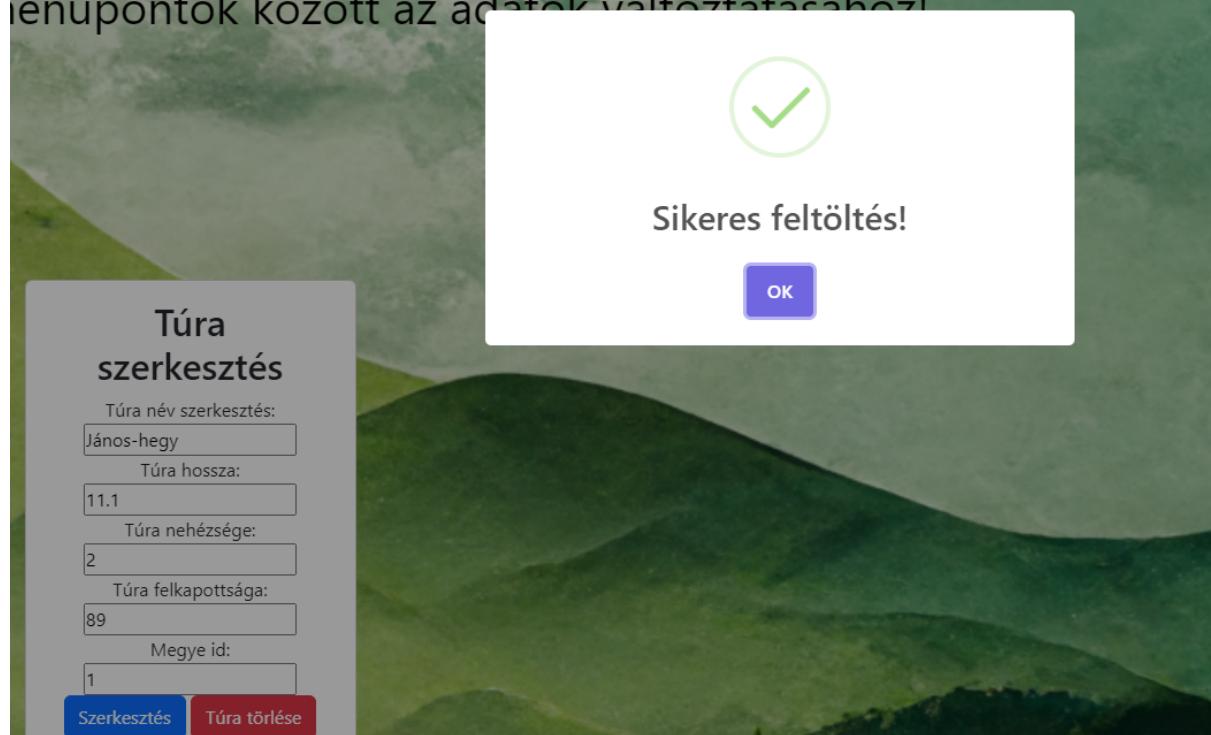
Túra törlése

Túra név esetében nem adhatunk meg ugyan olyan nevet, mint ami szerepel az adatbázisban, ha olyat adunk meg, nem tölti fel az adatbázisba a változtatásokat.

Ha sikeres volt:

Párek az admin felületen!

Mindenüppontok között az adatok változtatásához!



Túra hozzáadása

A "túra hozzáadás" gomb megnyomásakor megjelenik egy hasonló szerkesztő, csak üresen.

Túra neve:

Túra hossza:

Túra nehezsége:

Túra felkapottság:

Megye id:

Túra leírása:

Kép neve:

Kép fájl:

Tárhely... Nincs kijelölve fájl.

Túra hozzáadása

Itt fel lehet tölteni a túra minden adatát képpel együtt.

A “megye id” jelöli, hogy hanyadik megyébe kerüljön az adatbázisban a túra, melyet később ott jelenítünk meg.

Admin Felület fejlesztői dokumentáció

Amikor az belép a felhasználó a fiókjába akkor a navbar.php-ba megnézzük hogy a \$_SESSION[nev] egyenlő-e az adminnal.

Ha igen akkor a Navbarba belerakunk egy új gombot , ha rá kantinkunk akkor elvisz az adminfelulet.php-ra.

```
if(isset($_SESSION['nev'])){
    if($_SESSION["nev"] == "admin")
    {
        print("
<li class='nav-item'>
    <a class='nav-link' href='../frontend/adminFelulet.php' role='button' aria-expanded='false'>
        Adminfelület
    </a>
</li>
")
```

Felhasználó hozzáadása fejlesztői dokumentáció

Ha a Felhasználók gombra kattintott az admin akkor a GET[adminmenu] változott 1-re állítjuk és ezzel meghívjuk az Adminfelulet osztályból a Felhasznalok funkciót.

```
if(isset($_GET["adminmenu"]))
{
    $adminlekeres = new AdminFelulet();
    if($_GET["adminmenu"]==1)
    {
        $adminlekeres->Felhasznalok();
    }
    ?>
```

Itt ki iratunk egy formot amiben egy submit gomb van amivel felhasználót tudunk hozzáadni.
Ha rákattintunk akkor kiírunk egy formot ahol egy új felhasználót tudunk hozzáadni.

```

if(isset($_POST["felhasznalohozzaadas"]))
{
print
'

<div class="card text-center" style="width: 19rem; height: 11rem;">
<div class="card-body">
<form action="" method="post">
<label for="nevecske2">Név:</label>
<input type="text" name="nevecske2">
<label for="jelszocska2">Jelszó:</label>
<input type="text" name="jelszocska2">
<label for = "kepek2">Profilkép:</label>
<select name="kepek2">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
</select>
</div>
<button type="submit"  name = "hozzaadas" class = "btn btn-primary">Hozzáadás</button>
</form>
</div>
</div>
';
}

```

Ha rákattintunk a Hozzáadás gombra akkor egy if(!empty)-el ellenőrizzük hogy az admin minden mezőt kitöltött. Ha ki töltötte akkor az Adminfelulet osztályból meghívjuk a FelhasznaloHozaadas funkciót és átadjuk a felhasználónak az adatait.

```

if(!empty($_POST["nevecske2"]) && $_POST["jelszocska2"] && $_POST["kepek2"]){
$adminlekeres = new AdminFelulet();
$adminlekeres-> FelhasznaloHozzaadas($_POST["nevecske2"],$_POST["jelszocska2"],$_POST["kepek2"]);
$_POST = array();
}

```

A FelhasznaloHozzaadas funkcióba egy sql lekérdezéssel megnézik hogy van-e ilyen felhasználó az adatbázisban.Ha van akkor egy swearalertel közöljük az adminnak hogy már van ilyen felhasználó.

```

function FelhasznaloHozzaadas($nev,$jelszo,$kepid)
{
    $nevcheck=$this->csatlakozas->query("SELECT * from felhasznalok where nev = '". $nev . "'");
    if($adat = $nevcheck->fetch_assoc())
    {
        echo '<script type="text/javascript">

        $(document).ready(function(){

            Swal.fire(
                "Már van ilyen felhasználónév!",
                "Adjon meg másik nevet!",
                "error"
            )
        })
        </script>
    ';
    }
}

```

Ha nincs ilyen felhasználó akkor belépünk az else ágba és feltöljük az adatbázisba az új felhasználót egy sql parancssal, ezt az adminnak egy sweatalertel jelzünk vissza hogy sikeres volt a feltöltés.

```

else
{
    echo '<script type="text/javascript">

    $(document).ready(function(){

        Swal.fire(
            "Sikeres feltöltés!",
            "",
            "success"
        )
    })
    </script>
';
$felhasznaloJavitás = $this->csatlakozas->query("INSERT INTO felhasznalok(nev,jelszo,user_kep_id)
| Values('".$nev."','".$jelszo."','".$kepid."')");
}

```

Felhasználó törlése és szerkesztése fejlesztői dokumentáció

A felhasználók menüpontra kattintva megjelenik az összes felhasználónak a képe és neve egy Szerkesztés gombbal.

```

$felhasznalokiiras=$this->csatlakozas->query("SELECT * from felhasznalok where nev != 'admin'");
while($adat = $felhasznalokiiras->fetch_assoc())
{
    print(
        '
        <div class="feature col">
            <div class="card" style="width: 18rem;">
                <img src=../kepek/profilKeppek/'.$adat["user_kep_id"].'.jpg class = "card-img-top">
                <div class="card-body">
                    <p>'.$adat['nev'].'</p>
                </div>
                <a href="?userid='.$adat['id'].'" class = "btn btn-primary">Szerkesztés</a>
            </div>
        </div>
        '
    );
}

```

Az admin amikor megnyomja az egyik felhasználónál a Szerkesztés gombot , akkor a GET[userid] változonak az értékét egyenlővé tesszük a felhasználó id-vel.

Egy if(isset)-el ellenőrizzük hogy van-e userid GET elve. Ha igen akkor az Adminfelulet osztályból meghívjuk a FelhasznaloSzerkeszes funkciót.

```

if(isset($_GET["userid"]))
{
    $adminlekeres = new AdminFelulet();
    $adminlekeres->FelhasznaloSzerkeszes();
}

```

A FelhasznaloSzerkeszes funkcióba kiírunk egy formot ahol a felhasználónak az adatait át írhatjuk. A formon belül van két gomb az egyik a Szerkesztés a másik a Felhasználó törlése gomb.

```

function FelhasznaloSzerkeszes(){
    $tartalom = "";
    $felhasznalokiras=$this->csatlakozas->query("SELECT * from felhasznalok where id = '".$_GET['userid']."'");

    if($adat = $felhasznalokiras->fetch_assoc()){
        $tartalom .='
        <div class="col-sm-4 p-3">
            <div class="card" style="width: 18rem;">
                <div class="card-body">
                    <form action = "" method = "post">
                        <label for = "nevecske">Név:</label>
                        <input type="text" name="nevecske" value="'.$adat['nev'].'">
                        <label for = "jelszocska">Jelszó:</label>
                        <input type="text" name="jelszocska" value="'.$adat["jelszo"].'">
                        <label for = "kepek">Profilkép:</label>
                        <select name="kepek">
                            <option value="1">1</option>
                            <option value="2">2</option>
                        </select>
                        <br>
                        <button type="submit" name = "szerkeszes" class = "btn btn-primary w-100">Szerkesztés</button>
                        <button type="submit" name="felhasznaloTorles" class = "btn btn-danger w-100">Felhasználó törlése</button>
                    </form>
                </div>
            </div>
        ';
    }
}

```

Ha rá katolunk a Felhasználó törlés gombra akkor egy if(isset)-el meghívjuk az Adminfelület osztályból a felhasznaloTorles funkciót és át adjuk a felhasználónak az id-t egy változóba.

```

if(isset($_POST["felhasznaloTorles"])){
    $admin = new AdminFelulet();
    $admin->felhasznaloTorles($_GET["userid"]);
}

```

Ez a funkció nagyon rövid, csak egy sql parancsal ki töröljük az adatbázisból azt a felhasználót amit kapott változóként a funkció. Majd utána az admin viszsa viszik a felhasználó szerkesztésére egy header(location)-el.

```

function felhasznaloTorles($id){
    $this->csatlakozas->query("DELETE FROM felhasznalok WHERE id = ".$id."");
    header("Location: ./adminFelulet.php?adminmenu=1");
}

```

Ha viszont rá kattintunk a Szerkesztés gombra akkor meghívjuk a Adminfelulet osztályból a FelhasznaloUpdate funkciót és elküldjük POST változókban a felhasználónak az adatait.

```

if(isset($_POST["szerkeszes"]))
{
    $adminlekeres = new AdminFelulet();
    $adminlekeres->FelhasznaloUpdate($_POST["nevecske"], $_POST["jelszocska"], $_POST["kepek"]);
}

```

Ebben a funkcióba egy sql lekérdezést megnézünk hogy van-e ilyen felhasználó az adatbázisban. Ha van akkor egy swal alerttel közöljük az adminnak hogy már van ilyen felhasználói név.

```

function FelhasznaloUpdate($nev,$jelszo,$kepid)
{
    $felhasznalokiiras=$this->csatlakozas->query("SELECT * from felhasznalok
where nev = '".$nev."' and id != '".$GET["userid"].".'");
    if($adat = $felhasznalokiiras->fetch_assoc())
    {
        if(isset($_POST["szerkeszes"])){
            echo '<script type="text/javascript">
$(document).ready(function(){

    Swal.fire(
        "Már van ilyen felhasználónév!",
        "Adj meg másik nevet!",
        "error"
    )
})</script>
';
        }
    }
}

```

Ha nincs ilyen felhasználó akkor belépünk az else ágba és feltöljük az adatbázisba az felhasználónak az új adatait egy sql parancssal, ezt az adminnak egy swal alerttel jelzünk vissza hogy sikeres volt a felhasználónak a szerkesztés.

```

else
{
    if(isset($_POST["szerkesztes"])){
        echo '<script type="text/javascript">

$(document).ready(function(){

    Swal.fire(
        "Sikeress feltöltés!",
        "",
        "success"
    )
})</script>
';
    }

$felhasznaloJavitás = $this->csatlakozas->query("UPDATE felhasznalok SET nev =
'".$_nev."', jelszo = '".$_jelszo."', user_kep_id = '".$_kepid."' WHERE id = '".$_GET['userid']."'");

$_SESSION["nev"] = $nev;
}

```

Turák fejlesztői dokumentáció

Ha a Túrák gombra kattintott az admin akkor a GET[adminmenu] változott 2-re állítjuk és ezzel meghívjuk az Adminfelület osztályból a Turak funkciót.

```

if($_GET["adminmenu"]==2)
{
    ?>
    <div class = "container text-center">
        <div class = "row">
            <?php
                $adminlekeres->Turak();
            ?>
        </div>
    </div>
    <?php
}

```

A Turak funkcióban egy sql lekérdezésben kiiratjuk az összes turának a nevét , leírását és a képét. Emellett kiírunk két gombot az egyik tura hozzáadásra való a másik pedig túra szerkesztésére.

```
function Turak()
{
    $tartalom = '';
    $turakiiras = $this->csatlakozas->query("SELECT *, SUBSTRING(tura_leiras.tura_szoveg, 1, 200)
    | AS vagott FROM turak INNER JOIN tura_leiras ON turak.id=tura_leiras.id");
    print
    '<form action="" method="post" >
    <button type="submit" name="turahozaadas" class = "btn btn-primary">Tura hozzáadás</button>
    </form>';
    while($adat = $turakiiras->fetch_assoc())
    {
        $tartalom .= '
        <div class="col-sm-4 p-3">
            <div class="card" style="width: 18rem;">
                
                <div class="card-body">
                    <h5 class="card-title">'.$adat['tura_nev'].'
                    <p class="card-text">'.$adat['vagott'].'...
                </div>
                <a href="?turaid='.$adat["id"].'" class = "btn btn-primary">Szerkesztés</a>
            </div>
        </div>
        ';
    }
}
```

A Turak funkcióban egy adott tura szerkesztése a következőképpen zajlik, a “szerkesztés” gombra kattintva egy \$_GET[‘turaid’] változóban megkapjuk a szerkesztendő tura ID-t.

Turák törlése és szerkesztése fejlesztői dokumentáció

A túra adatai módosítása után, a “szerkesztés” gombra kattintva az adatok feltöltésre kerülnek az adatbázisba. Feltöltés előtt összevetjük a szerkesztett tura nevét az adatbázisban rögzített túrák neveivel, így elkerülve, hogy egy tura duplikálva legyen.

```
function TuraUpdate($turanev,$turahossz,$turanehez,$turafelkap,$megye_id)
{
    $felhasznalokiras=$this->csatlakozas->query("SELECT * from turak where tura_nev = '".$turanev."' and id != '".$_GET["turaid"]." '");
```

Egy if/else elágazásban vizsgáljuk az adatokat. Ha a lekérdezés során nem kaptunk azonos nevet a szerkesztett tura nevével akkor fog lefutni a feltöltés.

A tura törléséhez megadjuk a kiválasztott tura ID-t, majd ez alapján fogjuk törölni az adott tura adatait az adatbázisból. Törlés után az oldal frissül és ismét a kilistázott túrákat látjuk magunk előtt.

```
function turaTorles($id){
    $this->csatlakozas->query("DELETE FROM turak WHERE id = ".$id."");
    header("Location: /frontend/adminFelulet.php?adminmenu=2");
}
```

```

if($adat = $felhasznalokiras->fetch_assoc())
{
    echo '
<script type="text/javascript">
$(document).ready(function(){
    Swal.fire({
        icon: "error",
        title: "Már van ilyen túra név!",
        text: "Adj meg másikat!",
        footer: "<a></a>" Vecsernyés András, 2 weeks ago • admin felület design, fooldal megye leírás
    })
})
</script>;
';

else
{
    echo '<script type="text/javascript">
$(document).ready(function(){
    Swal.fire(
        "Sikeres feltöltés!",
        "",
        "success"
    ))
}
</script>;
';
$turaJavítás = $this->csatlakozas->query("UPDATE turak SET tura_nev = '".$turanev."', tura_hossza = '".$turahossz."', tura_néhezseg = '".$turanehez."',
tura_felkapotttag = '".$turafelkap."', megye_id = '".$megye_id."' WHERE id = '".$_GET['turaid']."'");

}

```

Turák hozzáadása fejlesztői dokumentáció

Amikor az admin rákattint a túra hozzáadás gombra egy if(isset)-el megvizsgáljuk hogy létezik-e a \$_POST[turahozaadas] változó. Ha igen akkor egy formot kiírunk amiben az adminnak ki kell töltenie a türának az adatait. Itt a tűrához képet is fel kell tölteni.

```

if(isset($_POST["turahozaadas"]))
{
    print '
<div class="col-sm-4 p-3">
    <div class="card" style="width: 18rem;">
        <div class="card-body">
            <form action = "" method = "post" enctype="multipart/form-data">
                <label>Túra neve:</label>
                <input type="text" name="turanev1">
                <label>Túra hossza:</label>
                <input type="number" step="any" name="turahossz1">
                <label>Túra nehezsége:</label>
                <input type="number" name="turanehez1">
                <label>Túra felkapottság:</label>
                <input type="number" name="turafel1">
                <label>Megye id:</label>
                <input type="number" name="megyeid1">
                <label>Tura leírása:</label>
                <input type="text" name="tura_szoveg"><br/>
                <label>Kép neve:</label>
                <input type="text" name="tura_kep"><br/>
                <label>Kép fájl:</label>
                <input type="file" name="kep">
                <button type="submit" name = "turaadd" class = "btn btn-primary">Túra hozzáadása</button>
            </form>
        </div>
    </div>
';
}

```

Amikor el küldjük a formot a Túra hozzáadás gombbal akkor meghívjuk az AdminFelulet osztályból a TuraHozaadas funkciót.

```

if(isset($_POST["turaadd"]))
{
    $adminlekeres = new AdminFelulet();
    $adminlekeres->TuraHozaadass($_POST["turanev1"], $_POST["turahossz1"],
    $_POST["turanehez1"], $_POST["turafel1"], $_POST["megyeid1"], $_POST["tura_kep"], $_POST["tura_szoveg"]);
}

```

A funkció elején egy sql lekérdezésel megnézik hogy van-e ilyen túra név ha van akkor az adminnal ezt közöljük.

```

function TuraHozaadass($turanev,$turahossz,$turanehez,$turafelkap,$megye_id,$kepnev,$turaszoveg)
{
    $turakiir=$this->csatlakozas->query("SELECT * from turak where tura_nev = '". $turanev . "'");
    if($adat = $turakiir->fetch_assoc())
    {
        print("Már van ilyen tura név!");
    }
}

```

Ha nincs ilyen túra név akkor az else ágba egy INSER INTO sql parancsal feltöltyük a túrának az adatait a turak táblába. Ezután a túra szöveget a tura_leiras táblába feltöltyük és a túra id-t. A képet egy move_uploaded_file() beépített funkció segítségével elmentjük a kepektura mappába a megadott kép névvel.

```
else
{
    $turahozaadas = $this->csatlakozas->query("INSERT INTO turak (tura_nev,tura_hossza,tura_nehezseg,
    tura_felkapottsg,megye_id,tura_kep_nev)
    values('" . $turanev . "','" . $turahossz . "','" . $turanehez . "','" . $turafelkap . "','" . $megye_id . "','" . $kepnev . "')");
    $turakiiras = $this->csatlakozas->query("SELECT * from turak");
    $db = 0;
    while($adat = $turakiiras->fetch_assoc())
    {
        $db = $adat["id"];
    }
    $leirashozaadas = $this->csatlakozas->query("INSERT INTO tura_leiras (tura_szoveg,tura_id)
    values('" . $turaszoveg . "','" . $db . "')");
    move_uploaded_file($_FILES["kep"]["tmp_name"], "../kepektura/" . $_FILES["kep"]["name"]);
}
}
```

Túra oldal felhasználói dokumentáció

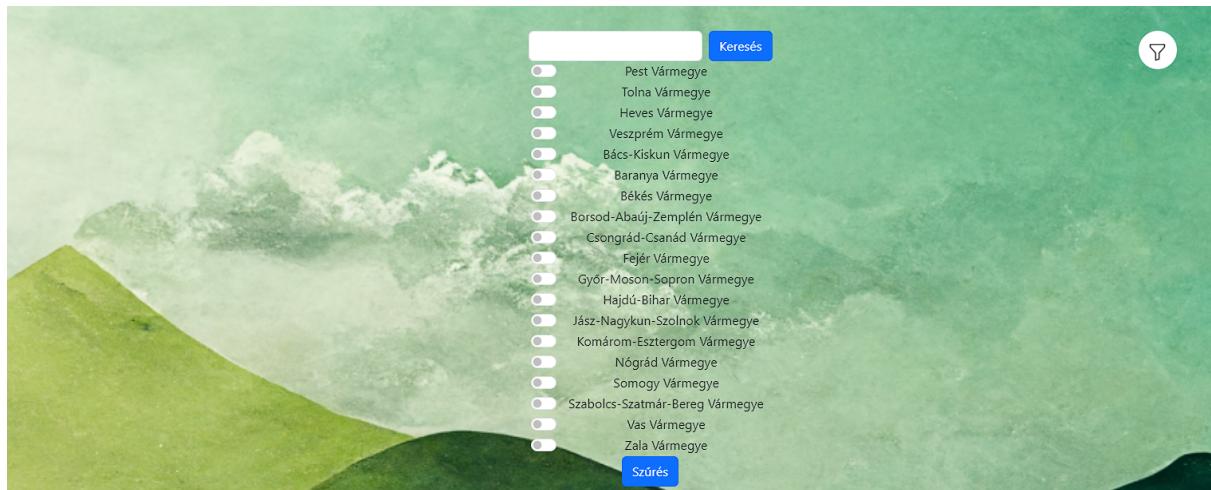
The screenshot shows a website layout for tourism information. At the top left is a logo with a tree icon and the word 'Ytravel'. At the top right are links for 'Főoldal', 'Túrák', 'Információ', 'Bejelentkezés' (with a user icon), and a search bar with a magnifying glass icon. Below the header is a green decorative background featuring stylized hills. In the center, there's a grid of three main destination cards and three smaller thumbnail images below them. Each main card has a large photo at the top, followed by the destination name, a brief description, and a 'Részletek' button. The first card is for 'Pest Vármegye', the second for 'Tolna Vármegye', and the third for 'Heves Vármegye'. The descriptions provide details about the region's natural beauty and cultural landmarks.

Pest Vármegye	Tolna Vármegye	Heves Vármegye
A térség a magyarországi turisztikai régiók közül a Budapest-Közép-Duna-vidék régióiba tartozik, fő vonzerejét a számtalan műemlék, a természeti élményterületek és azok kirándulóövezeteket, a hangulatos mackakőves utcás Szentendre műemlékei, a Dunakanyar és települései, a Bükkben, az erdővel borított hegyek és a már az Alföld részét képező síkságok és annak városai jelentik.	A Tolna vármegye Magyarország déli részén helyezkedik el, és számos gyönyörű túraútvonalat kínál a természet szerelmeseinek. Az alábbiakban néhány érdekes túralehetőséget mutatunk be.	Heves vármegye Magyarország északkeleti részén található, és a Mátra- és a Bükk-hegység között helyezkedik el. A terület gazdag természeti kincseiben, gyönyörű tájakban, erdőkben, völgyekben és hegyekben.
Részletek	Részletek	Részletek

A túrák oldalon először a vármegyék között lehet böngészni ahol többféle módon lehet szűrni ezeket.

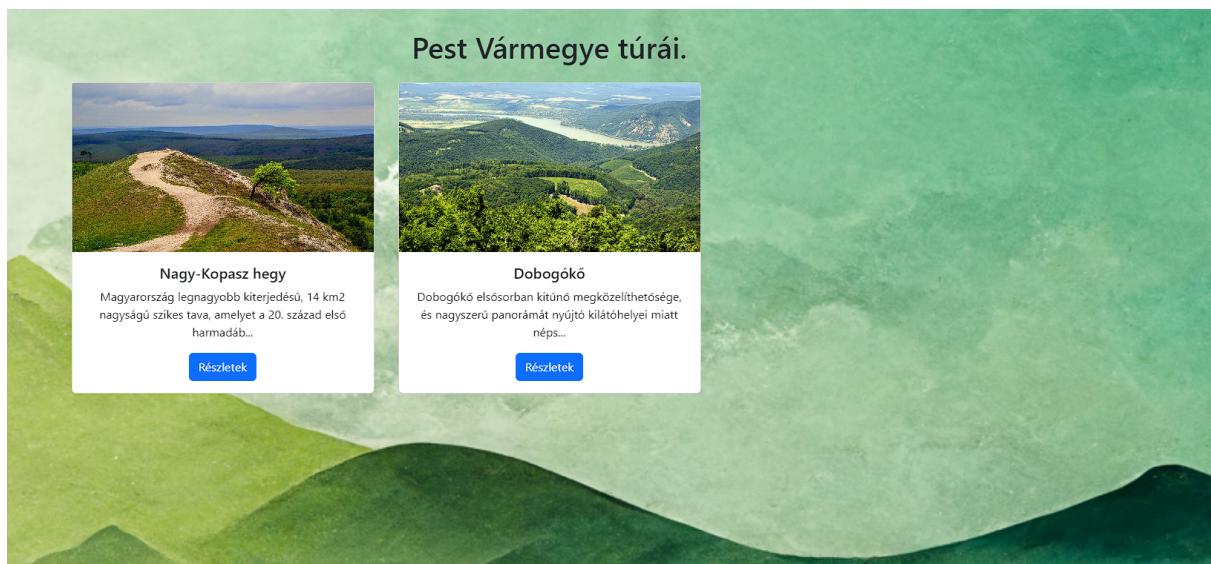


Ezen a szűrőn bármilyen keresőszó beirásával működik, akár csak 1 betűvel rá lehet keresni a vármegyékre, de ha elírásnál nem talál.



A másik szűrő a jobb felső gomb megnyomásával érhető el. Az egymás alatt megjelenő checkbox-okból bármelyiket és bármennyit ki lehet jelölni, a keresés sikeres lesz az alulsó “Szűrés” gomb megnyomásával.

A vármegyéknek láthatjuk a leírását a nevét és egy képet róluk, alattuk pedig egy “részletek” gombot, mellyel böngészhetünk az adott megye túráiban.



Például ha rányomtunk Pest vármegye “részletek” gombjára, láthatunk 2 túrát, névvel, leírással, képpel kiegészítve.

Természetesen mindenkorban, minden oldalon ugyan úgy kattinthatunk a menüpontok között.

The screenshot shows a green-themed landing page for 'Pest Vármegye túrái'. At the top right are links for 'Főoldal' (Main Page), 'Túrák' (Tours), 'Információ' (Information), and 'Bejelentkezés' (Login). The main content area features two tour cards. The first card for 'Nagy-Kopasz hegy' shows a photo of a dirt path on a hillside, with text describing it as the 'largest plateau in Hungary' (14 km²) and mentioning its history from the 20th century. It includes a blue 'Részletek' (Details) button. The second card for 'Dobogókő' shows a scenic view of a valley and a lake, with text about its panoramic views and popularity. It also has a blue 'Részletek' button. The background of the page is a large, stylized green landscape graphic.

Ha rányomunk az adott túra "részletek" gombjára a következő oldal tárul elénk:

The screenshot shows a detailed tour information page for 'Nagy-Kopasz hegy'. At the top right are links for 'Főoldal' (Main Page), 'Túrák' (Tours), 'Információ' (Information), and 'Bejelentkezés' (Login). The main content area is titled 'Túra információk' (Tour Information). It features a large photo of the hilltop. Below it, a section for 'Nagy-Kopasz hegy' provides specific details: 'Túra nehézsége: 45', 'Túra hossza: 10.5km', and 'Túra felkapottsága: 88%'. A progress bar visualizes the 88% climb. To the right, there is descriptive text about the hill's status as the largest plateau in Hungary, its history, and its location within the Kiskunság National Park. The background of the page is a large, stylized green landscape graphic.

Ezen az oldalon megtaláljuk a túra összes részletét, teljes leírással, a nevével, a hosszával, nehézségével, felkapottságával. **A felkapottság és a nehézség nem statisztikai adat, mivel ahhoz felhasználói értékelésekre lenne szükségünk, ezért adatbázisban előre tárolt adatokkal dolgoztunk.**

Amennyiben be vagyunk jelentkezve, láthatunk egy gombot, mellyel lementhetjük az adott túrát.

Túra információk



Nagy-Kopasz hegy

Túra nehézsége: 45

Túra hossza: 10.5km

Túra felkapottsága:

Magyarország legnagyobb kiterjedésű, 14 km² nagyságú szikes tava, amelyet a 20. század első harmadában halastrendszerré alakítottak. Növényzete, de különösen rendkívül gazdag, 280 fajt számláló madárvilágára miatt fokozottan védett terület, a Kiskunsági Nemzeti Park része. A táj változásait és a tó élővilágát bemutató Tisza-völgyi bemutatóház Szatymazon, az E75-ös út mellett található. A kiállítás megtekintése mellett érdemes végigjárni a Szalai tanösvényen, melynek egyik állomása a Beretzk kútja, amelyről a Felső-tó madárvilágát, kiemelten a Korom-sziget irányába lehet megfigyelni. A mintegy két kilométeres túrautvonal csak vezetével látogatható. Szintén a bemutatóháztól indulnak a darákok esti húzásának megfigyelésére induló szeneszett túrák az ószivonalusi időszakban.

Amint a fekete mentés gombra rákattint, megjelenik egy ablak a sikeres mentésről.

Túra információk



Nagy-Kopasz hegy



Sikeres Mentés!

OK

részze. A táj változásait és a tó élővilágát bemutató Tisza-völgyi bemutatóház Szatymazon, az E75-ös út mellett található. A kiállítás megtekintése mellett érdemes végigjárni a Szalai tanösvényen, melynek egyik állomása a Beretzk kútja, amelyről a Felső-tó madárvilágát, kiemelten a Korom-sziget irányába lehet megfigyelni. A mintegy két kilométeres túrautvonal csak vezetével látogatható. Szintén a bemutatóháztól indulnak a darákok esti húzásának megfigyelésére induló szeneszett túrák az ószivonalusi időszakban.

Amennyiben mégegyszer le akarná menteni ugyan azt, hibaüzenetet kap.

Túra információk



Nagy-Kopasz hegy



Már lementette egyszer!

Isten ments!

OK

völgyi bemutatóház Szatymazon, az E75-ös út mellett található. A kiállítás megtekintése mellett érdemes végigjárni a Szalai tanösvényen, melynek egyik állomása a Beretzk kútja, amelyről a Felső-tó madárvilágát, kiemelten a Korom-sziget irányába lehet megfigyelni. A mintegy két kilométeres túrautvonal csak vezetével látogatható. Szintén a bemutatóháztól indulnak a darákok esti húzásának megfigyelésére induló szeneszett túrák az ószivonalusi időszakban.

Túrák oldal fejlesztői dokumentáció

Amikor megnyitjuk a túrák oldalt láthatjuk a szűrőket és a vármegyéket. minden ami az oldalon megjelenik, php-val html-t küldünk vissza, illetve, ami az oldal saját content-je, az az adott html file-on belül megtalálható.

A szűrő megjelenése

```
<form action="" method="post" class = "d-flex gap-2 w-25 m-auto">
    <input type="text" name="kereses" id="kereses" autocomplete = "off" class = "form-control">
    <button type="submit" class = "btn btn-primary">Keresés</button>
</form>
```

Form-ot használtunk a “keresés” post elküldésére, melyre rákattintva megtörténik az érték alapján a keresés. Ezt az értéket továbbküldjük a “MegyeKiiras” function-nak, mely elvégzi a kiiratást.

```
<div class="row">
    <?php $megye->MegyeKiiras(isset($_POST["kereses"]) ? $_POST["kereses"] : null, isset($_POST["megyeNev"]) ? $_POST["megyeNev"] : null);?>
</div>
```

Már paraméterben megvizsgáljuk, hogy létezik-e vagy üres-e az “\$POST[“keresés”]”, mely tartalmazza a keresőszót és csak akkor küldjük tovább ha nem üres, illetve létezik.

```
function MegyeKiiras($keresesEredmeny, $megyeNev)
{
    $tartalom = "";
    $belepes = "";
    if(isset($megyeNev)){
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id WHERE megye.id
        IN ('.join(",",$megyeNev).')");
    }else if(isset($keresesEredmeny)){
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id WHERE megye_nev
        LIKE '%".$keresesEredmeny."%'");
    }else{
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id");
    }
}
You, last week via PR #31 • komponens alap, menu design, profil oldal, ikon, ...
```

A “MegyeKiiras” function feltétel alapján hoz létre lekérdezést, minden alkalommal, amikor paramétert kap. Az első feltétel akkor hoz létre lekérdezést, ha a checkbox-os szűrést alkalmaztuk, melynél több lehetőséget is tudunk egyszerre választani.

```
<div class="form-check form-switch">
    <input class="form-check-input" type="checkbox" role="switch" value = "'.$megyeNev["id"].'" name = "megyeNev[]">
    <label class="form-check-label" for="'.$megyeNev["id"].'>'.$megyeNev["megye_nev"].'</label>
</div>
```

Ez azért lehetséges, mert a name=“megyeNev[]” egy tömb, melybe több “megyeNev[“id”]” is bekerül és ezáltal a lekérdezésben csak azok kerülnek kiválasztásra, amik ebben a tömbben voltak.

A “MegyeKiiras” function alap helyzetben kiiratja az összes vármegyét, amik közül kerül szűrésre a késöbbiekbén.

Vármegyék leírásai, képek és “részletek gomb”



Pest Vármegye

A térség a magyarországi turisztikai régiók közül a Budapest–Közép-Duna-vidék régiókba tartozik, fő vonzerejét a számtalan műemlék, a természetvédelmi területek és azok kirányulóövezetei, a hangulatos macskaköves utcás Szentendre műemlékei, a Dunakanyar és települései, a Börzsöny, az erdővel borított hegyek és a már az Alföld részét képező síkságok és annak városai jelentik.

[Részletek](#)

A vármegye képe, leírása és a részletek gomb, mind a “MegyeKiiras” function által van megjelenive.

```
while ($adat = $belepes->fetch_assoc()) {  
    $tartalom.=''  
    <div class="col-md-4 p-3">  
        <div class="card">  
              
            <div class="card-body">  
                <h5 class="card-title">' . $adat['megye_név'] . '</h5>  
                <p class="card-text">' . $adat["megyeLeiras"] . '</p>  
                <a href="../frontend/turakInfo.php?idx=' . $adat['id'] . '" class="btn btn-primary">Részletek</a>  
            </div>  
        </div>  
    '>  
}  
print($tartalom);
```

a While ciklus az alábbi lekérdezések alapján fut le:

```

function MegyeKiiras($keresesEredmeny, $megyeNev)
{
    $tartalom = "";
    $belepes = "";
    if(isset($megyeNev)){
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id WHERE megye.id
        IN (".$join(",",$megyeNev).")");
    }else if(isset($keresesEredmeny)){
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id WHERE megye_nev
        LIKE '%".$keresesEredmeny."%'");
    }else{
        $belepes = $this->csatlakozas->query("SELECT *, SUBSTRING(megye_leiras.megye_szoveg,1,800)
        AS megyeLeiras FROM megye INNER JOIN megye_leiras ON megye.id = megye_leiras.id");
    }
}

```

You, last week via PR #31 • komponens alap, menu design, profil oldal, ikon, ...

Itt dől el hogy mit fog és hányat megjeleníteni a megyékből. Az \$belepes eredménye bekerül egy \$adat változóba, amit a fetch_assoc() tömbbé alakit. Ezáltal hozzáférünk a lekérdezés által a megye tábla minden oszlopához.

	id	megye_nev		turak_szama	megye_felkapottsg	megye_kep_nev
1	1	Pest Vármegye		1	1	1
2	2	Tolna Vármegye		1	1	1
3	3	Heves Vármegye		1	1	1
4	4	Veszprém Vármegye		1	1	1
5	5	Bács-Kiskun Vármegye		1	1	1

Részletek gomb

A részletek gomb arra szolgál, hogy tovább vigyen a következő oldalra az adott megyére vagy túrára kattintva.

Részletek

Megkapja az “id”-t, mely sorszámként szolgál a lekérdezésben az adott megyéhez.

```
<a href="../frontend/turakInfo.php?idx='".$adat['id']."' class="btn btn-primary">Részletek</a>
```

Az “adat[“id”]” az adatbázisban az elsődleges kulcs, mely alapján beazonosítjuk, melyik megyéről van szó.

id	megye_nev
1	Pest Vármegye
2	Tolna Vármegye
3	Heves Vármegye
4	Veszprém Vármegye
5	Bács-Kiskun Vármegye

Ezt pedig egy “a” taggel oldjuk meg, hogy olyan url jöjjön létre, mely alapján majd kiiratjuk a túrákat.

```
turazas.loc/frontend/turakInfo.php?idx=1
```

