



**„Schola Europa Akadémia” Technikum,**

**Gimnázium**

**és Alapfokú Művészeti Iskola a**

**Magyarországi Metodista Egyház**

**fenntartásában**

**SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ**

**5 0613 12 03**

**Vizsgaremek dokumentáció**

*2023*

**Paulicsek Orsolya, Pércsi Róbert, Tóth Attila**

# Változó Jövő

Bevezetés.....	5
Koncepció.....	5
Mi található weblapunkon? .....	6
Milyen hibák lehetnek regisztrációnál? .....	7
Milyen hibák lehetnek bejelentkezésnél? .....	8
Adományozás feltételei .....	9
Milyen információkat várunk el a felhasználótól adományozás esetén?.....	9
Milyen hibákba ütközhetünk adományozás közben? .....	9
Táborozás/jelentkezés feltételei .....	11
Milyen információkat várunk el a felhasználótól táborozás esetén? .....	11
Milyen hibákba ütközhetünk jelentkezés közben? .....	11
Felhasználói adatok változtatása .....	13
Milyen hibák fordulhatnak elő? .....	13
Fejlesztői felület .....	14
Általunk használt programozási nyelvek.....	14
HTML/CSS/BOOTSTRAP .....	15
SQL .....	15
Kapcsolataink az adatbázisok között.....	16
Állat- és növénytáblák.....	17
Join Tábla .....	17
Donations Tábla .....	18
User Tábla .....	19
PHP/Javascript .....	20
Back to top/Oldal tetejére gomb.....	20
Regisztráció/Bejelentkezés PHP/JS.....	20
Adományozás PHP/JS.....	21
Jelentkezés/Táborozás PHP.....	22
Felhasználói felület PHP .....	22
Automatikus kijelentkezés PHP/JS .....	23

Területek és táblázatai PHP/JS.....	23
Süti kezelés/Cookie PHP/JS.....	24
Tesztelések .....	25
Back to top/Oldal tetejére teszt.....	25
Táblázatok teszt.....	27
Inaktivitás mérő.....	28
Fejlesztési lehetőségek .....	30

## Bevezetés

Az ember mindig is arra gondolt hogyan is segíthetne Földünk megvédésében, ápolásában és ezt vajon hogyan oldhatná meg. De ez már a múlté, mivel kis csapatunk kiötletelte ezt az oldalt, amellyel most te is segíthetsz könnyedén a természet növény és állatvilágának megvédéséért.

## Koncepció

A projektünk alapötlete egy természetvédelmi oldal, amely nyilvántartja hazánk összes természetvédelmi területét, esélyt biztosít arra, hogy egyfelől alkalom nyíljon az általunk listázott környezetvédelmi területeink átfogóbb megismeréséhez, másfelől pedig további lehetőséget nyújtson táborozásra azok számára, akik szeretnék kicsit közelebb kerülni a természethez. Csakugyan van opció minket támogatni is, akinek esetleg módjában áll, vagy szeretne hozzájárulni küldetésünkhöz.

A választásunk pedig azért esett a környezetvédelemre, mivel napjainkban kiemelkedően fontos az élővilágunk megvédése és ápolása a szennyezéstől és kiirtásától.

### Mi található weblapunkon?

A látogatóink oldalunkat csak is és kizárólag böngészőn keresztül tudják elérni a <https://www.valtozojovo.hu> vagy [valtozojovo.hu](https://www.valtozojovo.hu) webcímen.

Weblapunk tartalmaz temérdeknyi információt és tartalmat hazánk természetvédelmi területeiről állat és növényvilágáról.

Ezek a területek pedig:

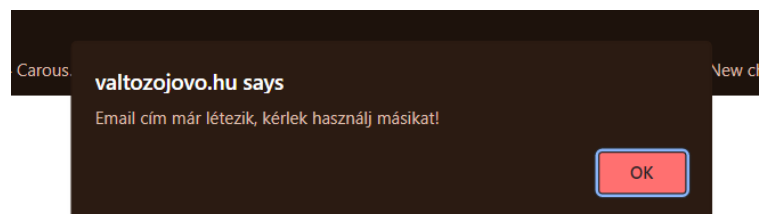
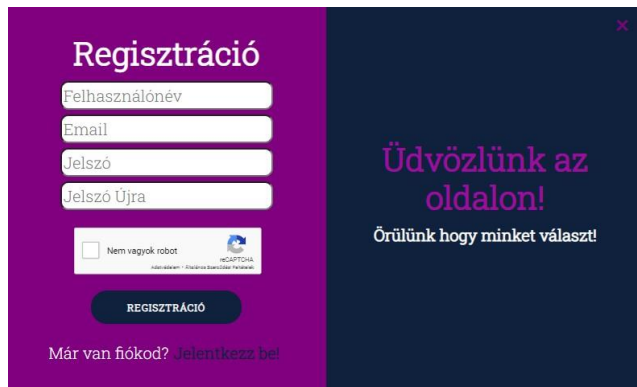
- Aggtelekei Nemzeti Park
- Balatonfelvidéki Nemzeti park
- Bükk Nemzeti Park
- Duna-Dráva Nemzeti Park
- Duna-Ipoly Nemzeti Park
- Fertő-Hanság Nemzeti Park
- Hortobágyi Nemzeti Park
- Kiskunsági Nemzeti Park
- Kőrös-Maros Nemzeti Park
- Őrségi Nemzeti Park

Emellett felfedezheti eddigi túráink és táborainkról készült fényképeket és idézeteket, melyekről boldogan gondolunk vissza. Ezen felül még lehetőség van tájékozódni a jövőbeli programjainkról eseményeinkről. Ha ez nem lenne elég akkor lehetőség van jelentkezni táborainkba, amelyről szintén lehet információt gyűjteni vagy esetleg, ha igény van rá lehet ügyfélszolgálatunknál érdeklődni vasárnap kivételével a hét minden napján. Természetesen ez az oldal nem is létezne a kedves adományozó személyek nélkül. Ezért, ha van rá mód akkor mi lehetőséget nyújtunk arra is, hogy minket és csodaszép vidékeinket egyaránt támogassuk

Természetesen nem minden funkció elérhető egy átlagos felhasználó számára ezért a táborozás lehetőséget regisztráció mögé rejtettük így csak akkor van lehetőség jelentkezni táborainkra, ha már hivatalosan is felhasználó oldalunkon. Szerencsére ügyeltünk minden eshetőségre így felhasználóbaráttá tettük egyaránt a bejelentkezést is mint a regisztrációt is.

Ha a felhasználó sikeresen regisztrál akkor az oldal által mi ki küldünk egy email-t, amely igazolja a sikeres regisztrációt. Ezután már sikeresen be lehet jelentkezni oldalunkra és minden funkció elérhetővé válik.

Bejelentkezés után szabad utat kap a felhasználó táborainkba jelentkezéshez. Mindemellet saját fiókunkat akár személyre is szabhadtjuk vagy adatainkon változtatást végezhetünk. Természetesen a személyes adatok végett biztonsági intézkedéseket végeztünk, hogy az ön adatai biztonságban maradjon.



### Milyen hibák lehetnek regisztrációnál?

Az első és legfontosabb dolog az, hogy minden adatot jegyezzünk meg a későbbi bejelentkezés esetén.

Regisztrációhoz szükség van egy érvényes email címre, egy felhasználónévre és egy jól megjegyezhető jelszóra.

Esetleges hibák melyek előfordulhatnak a regisztráció folyamán:

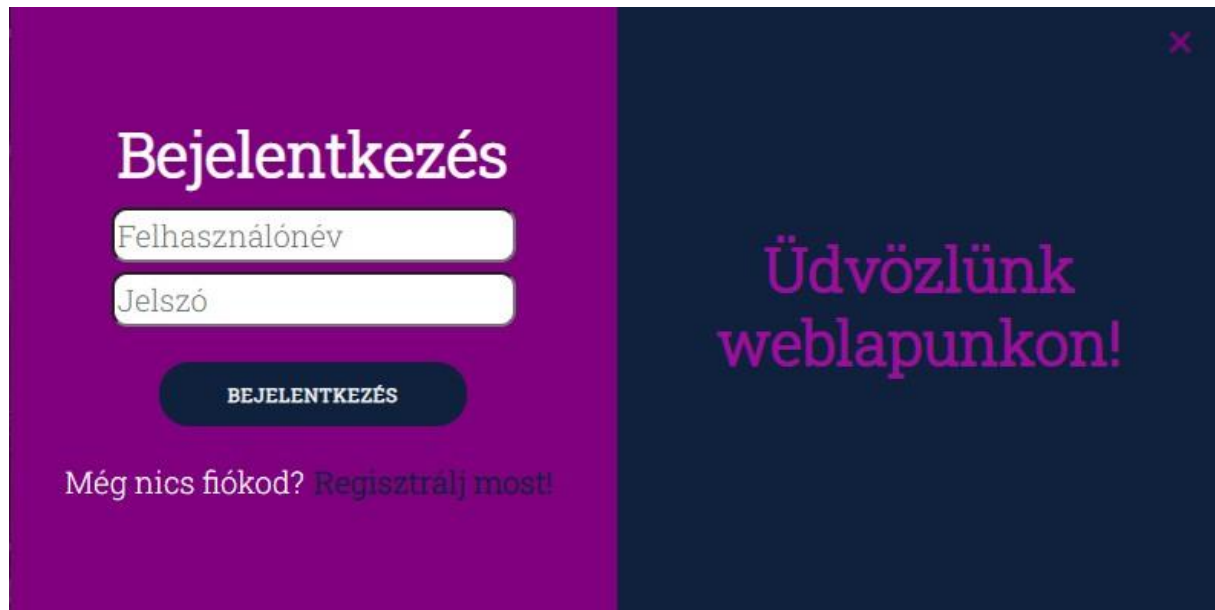
- A felhasználónév már foglalt – Ebben az esetben mi gondoskodunk a felhasználónévről mellől lehet dönteni, hogy elfogadjuk vagy sem
- Email cím már regisztrálva van – Szeretnénk, ha minden felhasználó egyszer regisztrálhasson oldalunkra, hogy mindenkinek legyen lehetősége regisztrálni oldalunkra
- Email cím nem megfelelő a formátumnak – Csak bizonyos email címeket fogadunk el (Gmail, Yahoo, Outlook)
- Jelszó formátuma nem egyezik meg – A nagyobb biztonság érdekében szeretnénk, ha a felhasználók olyan jelszót használnának melyet nehezebb feltörni. Így ezért a jelszónak minimum 8 karakternek kell lennie és tartalmaznia kell legalább 1 nagy és 1 kis betűt
- A két jelszó mező nem egyezik meg – Annak érdekében, hogy ne lehessen viccből regisztrálni így a jelszónak és a jelszó ismétlésének meg kell egyeznie.

Milyen hibák lehetnek bejelentkezésnél?

Bejelentkezésnél a felhasználótól csak saját felhasználónevét és jelszavát kérjük tőle.

Hogy sikeres legyen a bejelentkezés ezért ezekre figyeltünk oda:

- Felhasználónév nincs regisztrálva vagy hibás
- Jelszó nem megfelelő



The image shows a login interface with a purple background on the left and a dark blue background on the right. On the purple side, the title 'Bejelentkezés' is at the top. Below it are two white input fields: the first is labeled 'Felhasználónév' and the second is labeled 'Jelszó'. A dark blue button with the text 'BEJELENTKEZÉS' is positioned below the fields. At the bottom of the purple section, it says 'Még nics fiókod? Regisztrálj most!'. On the dark blue side, there is a small red 'X' icon in the top right corner and a large, stylized red text message that reads 'Üdvözlünk weblapunkon!'.



### Adományozás feltételei

Az adományozás oldalrész bármilyen jogosultással elérhető a „Támogasd a természetet!” fülön. Regisztrált és átlagos felhasználó között ezen az oldalon nagy különbség nem fedezhető fel. A regisztrált fiókok abban az előnyben részesülhetnek, hogy email címüket már előre kitöltve jelenítjük meg.

Ha esetleg ez az email cím nem az, amelyet szeretnénk használni, akkor is van lehetőség ennek megváltoztatására.

### Milyen információkat várunk el a felhasználótól adományozás esetén?

Adományozóinknak kettő lehetőséget ajánlunk fel mely lehet kártyás fizetés az oldalon keresztül vagy átutalás alapján, ahol az adatok leadása után email-ben küldjük számlázási adatokat.

Ahhoz, hogy sikeres adományt végezzünk el, ahhoz a felhasználónak szüksége lesz ezekre az adatokra:

- Teljes név
- Email cím
- Egy felhasználó által megadott értékre, amely forintban (Ft) értetődő.

Ha a felhasználó úgy dönt kártyás fizetéssel szeretne minket támogatni akkor további adatokra lesz szükségünk, hogy a tranzakció sikeresen végbe mehessen.

Ezekre a további adatokra van szükség:

- Kártya szám, mely csakis kizárólag Visa vagy Mastercard lehet
- Kártya tulajdonos neve
- Lejárat dátum, mely ily módon van feltüntetve „hónap / év”
- Biztonsági kód (CVC). A kártya hátlapján található 3-4 számjegy

Ha minden adat sikeresen kitöltve és az „Adományozás” gombra kattintva megkapjuk a „sikeres adományozás” üzenetet akkor szintén egy email formájú értesítést küldünk az adott felhasználónak az alapján, hogy milyen fizetési módszert választott.

### Milyen hibákba ütközhetünk adományozás közben?

Mivel hibázni emberi dolog így egy ilyen adatlapnál is van rá esély, hogy hibázunk.

Ennek érdekében ajánljuk a figyelmes kitöltést és az utána történő ellenőrzést is.

Milyen hibák lehetnek a kitöltés folyamán:

- Email cím nem felel meg a formátumnak – Ahogy a regisztrációnál így itt is ugyan azok az email-ek használhatók
- Adatok üresek – Minden adatnak kitöltve kell lennie
- Csak Visa és Mastercard fogadható el
- A kártya lejárat dátuma csak 2023 és 2026 között lehet érvényes – Azért, hogy ne hamis lejárt adatokat kezeljünk.

Teljes név:

Név

Email:

atus200105@gmail.com

Hova szeretnél adományozni?

Bármelyik

▼

Mennyit szeretnél adományozni?

Összeg

Ft

Fizetési mód:

☐ Utalással ☒ Kártyával

Kártya adatai:

Kártyaszám

0123-4567-8910-1112

Kártyatulajdonos neve:

Kártyatulajdonos neve

Lejárat dátum:

Hónap

/

Év

CVC:

CVC

Adományozás

10

### Táborozás/jelentkezés feltételei

Táborozáshoz amint már említve lett csak is regisztrált majd bejelentkezett felhasználónak van lehetősége.

### Milyen információkat várunk el a felhasználótól táborozás esetén?

Ezekre az információkra van szükségünk a jelentkezéshez:

- Gyermek keresztnév és vezetéknév
- Szülő/gondviselő keresztnév és vezetéknév – Ha a felhasználó elmúlt 18 akkor magának is kitöltheti
- Gyermek születési éve – Maximum 23 éves minimum 5 éves lehet a jelentkező
- Helyszín, ahova a tábor szól
- Ország - Amely alapértelmezetten Magyarország és nem lehet változtatni
- Megye
- Irányítószám
- Város
- Utca, házszám
- Emelet ajtó – Nem kötelező
- Telefonszám
- Email cím – Szintén formátumnak megfelelően

### Milyen hibákba ütközhetünk jelentkezés közben?

Egy nagyobb adatlapon több az esély hibázni ezért ezen részen kimondottan figyelni kell adatinkra

Hibák melyek lehetségesek lehetnek:

- Hiányzó adatok
- Telefonszám formátuma – A telefonszám csak is magyar lehet, melyet 06 vagy +36 formájában fogadunk el

- Email cím – A feltételek nem változnak

<b>Gyermek neve*</b>	
Vezetéknév:	
<input type="text" value="Gyermek Vezetéknéve"/>	
Keresztnév:	
<input type="text" value="Gyermek Keresztneve"/>	
Születési év:	
<input type="text" value="mm/dd/yyyy"/>	
Helyszín:	
<input type="text" value="Szalamandra Erdeti Iskola - Aggteleki Nemzeti Park"/>	
<b>Szülő/Gondviselő neve *</b>	
Vezetéknév:	
<input type="text" value="Szülő Vezetéknéve"/>	
Keresztnév:	
<input type="text" value="Szülő Keresztneve"/>	
<b>Lakcím *</b>	
Ország:	
<input type="text" value="Magyarország"/>	
Megye:	
<input type="text" value="Baranya megye"/>	
Irányítószám:	
<input type="text" value="Irányítószám"/>	

Város:
<input type="text" value="Város"/>
Utca, Házzám:
<input type="text" value="Utca, Házzám"/>
Lépcsőház, Emelet, Ajtó*
<input type="text" value="Emelet, ajtó (Nem kötelező)"/>
<b>Elérhetőség*</b>
Telefonszám
<input type="text" value="+36123456789 vagy 06123456789"/>
Csak magyar telefonszám fogadható el! (+36 / 06)
Email:
<input type="text" value="atus200105@gmail.com"/>
<input type="button" value="Jelentkezés"/>

### Felhasználói adatok változtatása

Ahhoz, hogy ezt a funkciót megtehessek, előbb regisztrálnunk és bejelentkeznünk is kell.

Majd ezek után hozzáférünk a felhasználói felülethez, ahol személyes adatainkat találhatjuk, mint például a felhasználónevet, az email-t és a jelszót.

Ezek közül a felhasználónevet és jelszót van lehetőségünk változtatni. Ezt a két adatot külön és egybe is cserélhetjük, ha úgy tartja kedvünk.

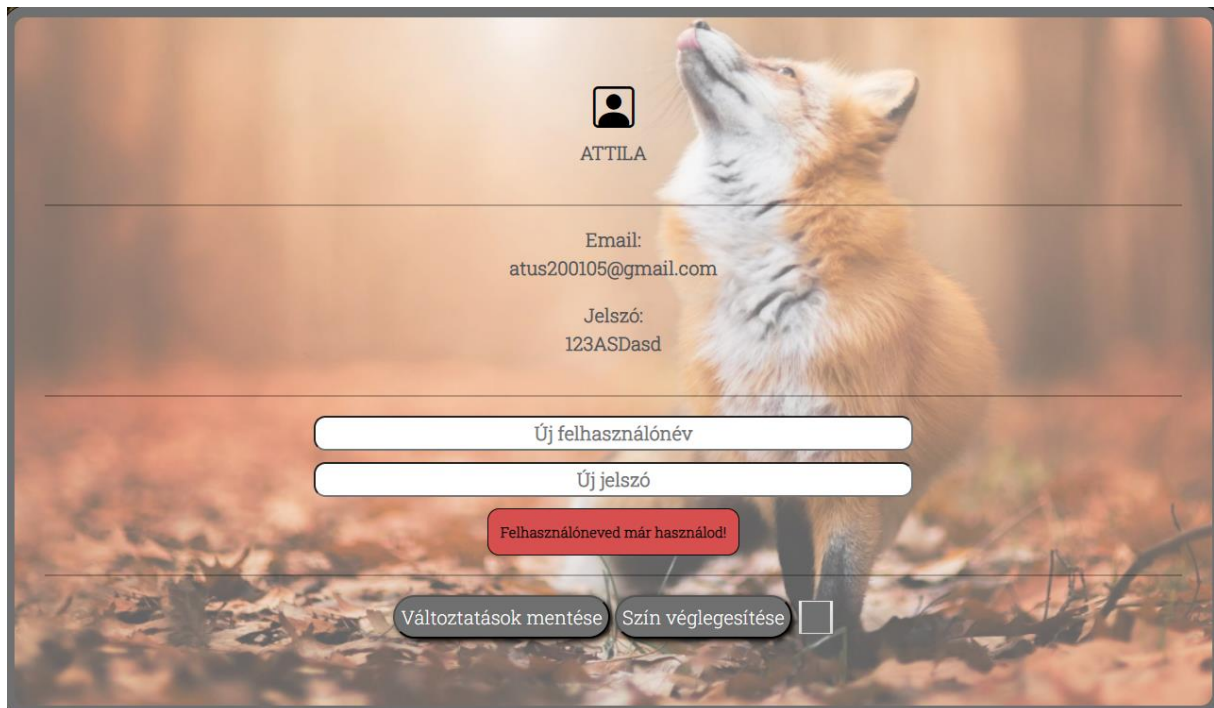
Persze ha úgy döntünk akkor felhasználói felületünket személyre is szabhatjuk színválasztónkkal.

### Milyen hibák fordulhatnak elő?

A felhasználói adatok frissítésében is hasonló hibakezelés megy végbe, mint a regisztrációnál

Ezek pedig:

- A felhasználónév már létezik
- A felhasználónév ugyanaz, mint amire változtatni szeretnénk
- A jelszó nem felel meg a formátumnak (8 karakter, 1 nagy és kis betű)
- A jelszó ugyanaz, mint a jelenlegi
- A kiválasztott szín már használatban van



The screenshot shows a user profile editing interface. At the top, there is a profile icon and the name "ATTILA". Below this, the email address "atus200105@gmail.com" and the password "123ASDasd" are displayed. There are two input fields for "Új felhasználónév" (New username) and "Új jelszó" (New password). A red error message "Felhasználóneved már használatban!" (Your username is already in use!) is shown below the username field. At the bottom, there are two buttons: "Változtatások mentése" (Save changes) and "Szín véglegesítése" (Finalize color), followed by a small square checkbox.

## Fejlesztői felület

Az oldal fejlesztése közben törekedtünk az adatok biztonságos tárolásában és tartásában emellett a kinézet sem volt utolsó szempont

## Általunk használt programozási nyelvek

A projekt elkészítéséhez, valamint fejlesztéséhez az alábbi nyelveket használtuk:

- HTML5
- CSS
- JavaScript
- PHP
- SQL
- Bootstrap 5

Az oldal tesztelését több eszközön és böngészőn is megtettük a biztos működés érdekében

Ezeket a böngészőket és eszközöket teszteltük:

- Böngészők:
  - Google Chrome
  - Opera
  - Microsoft Edge 10, 11
  - Safari (Telefon)
- Eszközök:
  - Mobil
  - Tablet
  - Asztali gép

Az oldal létrehozásához a Visual Studio Code, Notepad++, Adobe Photosop és Illustrator programokat használtuk. Míg az Adobe programokat a képekre és személyes logónkra használtuk addig a Visual Studio Code és a Notepad++ a programozás részében segített.

A munka nagyrészt laptopon végeztük el melynek specifikációi ezek:

- Típus: Asus X571
- Processzor: i5-9300H
- Videókártya: GTX 1650
- Ram: 16Gb

## HTML/CSS/BOOTSTRAP

A weboldal felépítése Bootstrap5 CSS, és JavaScript keretrendszer, HTML5 nyelven íródott, utf8 kódolással ellátva, és CSS3 formázással és Javascript animációk segítségével készült.

### **Weboldal felépítése:**

- Főoldal
- Táborozz Velünk!
- Támogasd a természetet!
- Területek
- Eddigi események
- ÁSZF
- Felhasználói felület
- Regisztrációs/Bejelentkezési adatlap

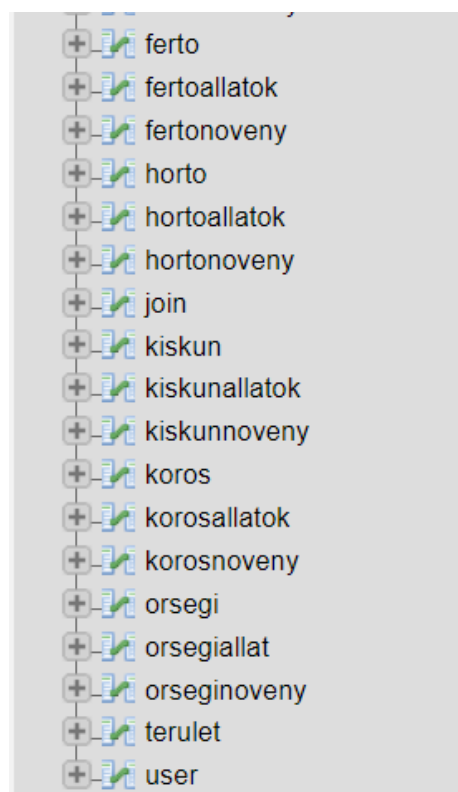
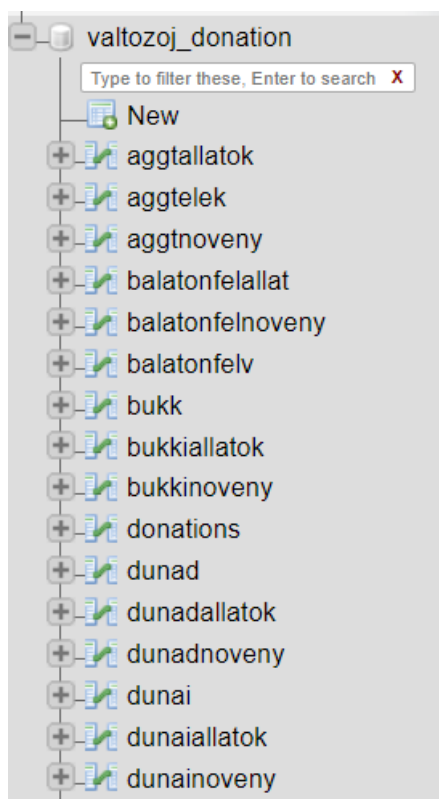
## SQL

Természetvédelmi projektünk során az általánosan használt nyílt forráskódú MYSQL relációs adatbázist használtunk. Táblákat és a hozzájuk tartozó adatokat a Notepad++ illetve Visual Studio Code alkalmazással hoztunk létre. Az adatbázis utf8mb4\_general\_ci karakterkódolással van ellátva, amely egyszerűsített rendezési szabályokat használ, a sebesség javítása valamint nem kívánatos összehasonlítást eredményez bizonyos nyelvek, vagy karakterek használatakor.

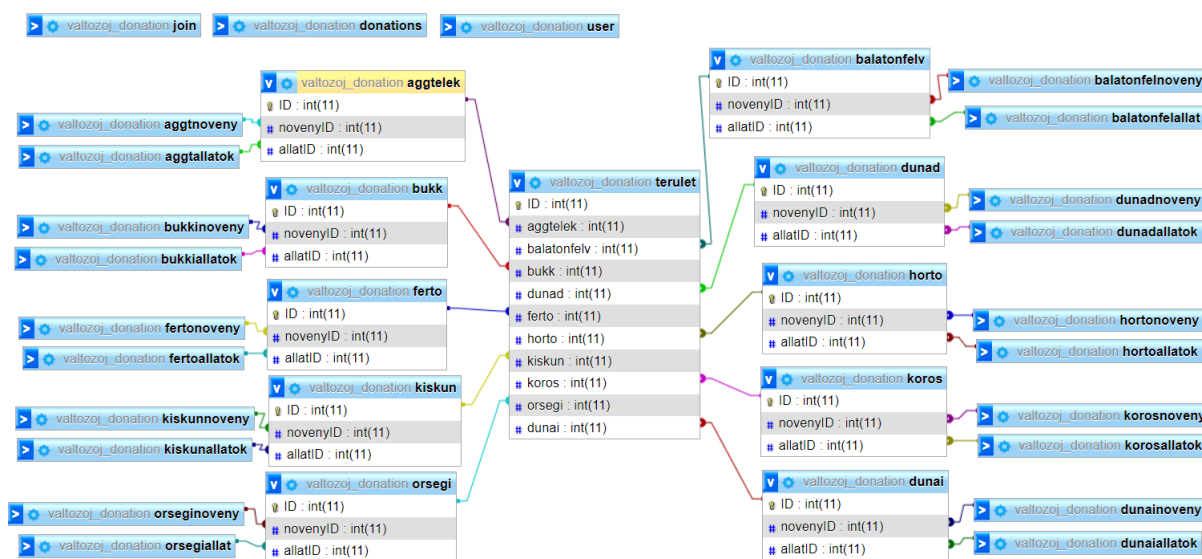
Ci jelentése: A “case-insensitive” kifejezés azt jelenti, hogy a kis- és nagybetűk megkülönböztetése nélkül történik valami.

Az adatbázisunk 34 táblát tartalmaz. Ezek közül 30 táblát két csoportra tudjuk osztani. Az egyik a növények a másik pedig az állatok. Ezek külön területre vannak bontva. Mindegyik növény és állat tábla kap 2-2 saját id-t melyeket összecsatolunk. Összesen 10 természetvédelmi terület

van ezért minden területhez kerül egy-egy állat és növény tábla a saját adataival és 2 állat és növény id.



## Kapcsolataink az adatbázisok között





## Állat- és növénytáblák

Az állat- és növénytáblák felépítése a következő:

Az állatok táblája integer és varchar értékeket tud tárolni.

**Integer:** Az elsődleges kulcsunk és a létszám.

**Varchar:** A faj, a név, a helyszín és a veszélyeztetettség. Növénytáblánk hasonló, de egy különbséggel. Az id, ami a fő kulcsunk is, számértékű, míg a többi mező varchar értékeket tartalmaz, mint például a faj, a név, a helyszín, a veszélyeztetettség és a virágzás.

A társított tábláinkat a PHPMyAdmin segítségével csatoltuk a szerverhez, mivel az állat- és növény adatokat a weboldalunkon jelenítjük meg. Ha a látogató kiválaszt egy természetvédelmi területet, megtekintheti az adatokat a php kód által legenerált táblázatokban. Ezeket az adatokat a PHPMyAdmin-ban tároljuk el és onnan is hívjuk meg.

## Join Tábla

A join táblát a táborozásnál használjuk, mivel sok táborozást szervezünk, különösen gyerekeknek és fiatal felnőtteknek. Ezáltal sok információra van szükségünk, ahhoz, hogy a táborozáshoz való jelentkezés véghez tudjon menni. Amennyiben minden adat megfelelően lett megadva, úgy a táborozásra való jelentkezés adatait a szerverünkön tároljuk el.

A join tábla két integer-t, hét longtext mezőt, hat varchar mezőt és egy dátum típust tartalmaz.

**Integer:** Az elsődleges kulcsunk (id) és az irányítószám. Az irányítószám legfeljebb négy számjegyű lehet, fontos, hogy csak szám alapú lehet.

**Longtext:** A táborozni vágyó gyermek és a gondviselő vezetékneve, keresztnéve, valamint az ország, ami alapértelmezetten Magyarország, a kiválasztott megye és a hozzátartozó város, amelyek a lakcímhez kapcsolódnak.

**Varchar:** A gyermek születési dátuma, a tábor helyszíne és megnevezése, amit ki tud választani a látogató. Utca, házszám, emelet, ajtó, telefonszám és e-mail címe. A telefonszám magyar telefonszám kell, hogy legyen, ami lehet 06 vagy +36 kezdésű. Az e-mail cím nem csak gmail alapú lehet.

Lehet például:

- Yahoo
- Outlook
- Hotmail

**Dátum:** A dátum automatikusan generált adat, amely létrejön ha a felhasználó elvégez egy jelentkezést. Ezáltal nyilván tudjuk tartani ki mikor jelentkezett táborainkba

#### Donations Tábla

Az általunk bemutatott természetvédelmi területeket lehetőség van támogatni, az ehhez a célra elküldött támogatásokat a szerveren tároljuk el.

A donations tábláját az adományokhoz használtuk, amely 13 elemet tartalmaz, három integer, tíz varchar mezőt és egy dátum típust.

**Integer:** Az elsődleges kulcsunk (donation-id), az év és a hónap, amelyek összeállítják a lejárat dátumot.

**Varchar:** Az adományozó neve és e-mail címe, az összeg, a fizetési mód, ami kétféleképpen valósulhat meg, utalással, vagy kártya által. A valuta pedig forint (Ft).

Ha a kártyás opciót választja, akkor meg kell adnia a kártya adatokat, ami kártyaszámból, a kártyatulajdonos nevéből, lejárat dátumából, és a hitelesítési kódjából (CVC) áll.

Csak Visa és MasterCard típusú kártya fogadható el. Amelyeknek mindenképpen érvényesnek kell lenniük.

**Dátum:** A dátum automatikusan generált adat, amely létrejön ha a felhasználó elvégez egy adományt. Ezáltal nyilván tudjuk tartani ki mikor küldött nekünk adományt.

## User Tábla

A user táblát a regisztrációhoz, és a bejelentkezéshez, valamint a felhasználói felülethez kötöttük. Ha a regisztráció sikeres volt, abban az esetben az adatokat phpMyAdmin segítségével a szerverünkön tároljuk el.

6 elemmel rendelkezik, melyből 1 int, 4 varchar, és 1 dátum típusú.

**Integer:** Ide az elsődleges kulcs (id) tartozik, hiszen minden egyes regisztrált felhasználó rendelkezik egy külön hozzárendelt id-val.

**Varchar:** A felhasználó neve, az általa létrehozott jelszó, az általa megadott email, és a későbbiekben általa kiválasztott felhasználói felület form keret, és a benne lévő gombok háttérszíne.

**Dátum:** A dátum automatikusan generált adat, amely létrejön ha a felhasználó regisztrál oldalunkon. Ezáltal nyilván tudjuk tartani ki mikor hozott létre fiókot.

## PHP/Javascript

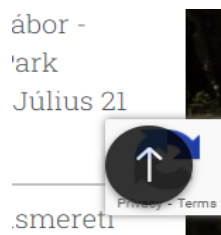
Számunkra a PHP volt ideális mint kódolási nyelv mivel egy weblapot készítettünk. Így a PHP segítségével eltudtuk tárolni az adatok vagy esetleg lehívni őket.

Az oldalon funkciókból nincs hiány ezért több mindent is be lehet mutatni

## Back to top/Oldal tetejére gomb

Mivel oldalaink tartalmilag elég hosszadalmasak ezért szükségét érezte eme gombnak

A dizájn és animáció részét css-ben végeztük el és a funkcióját pedig Javascript-ben.

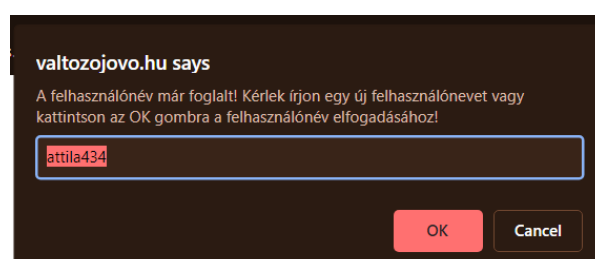
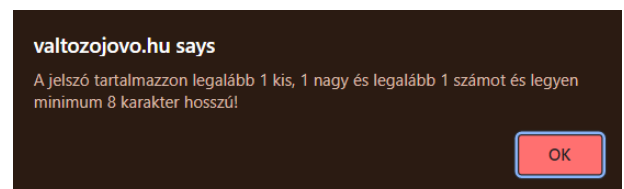
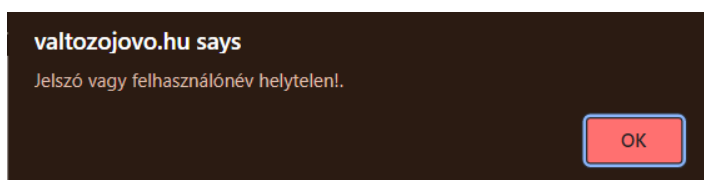
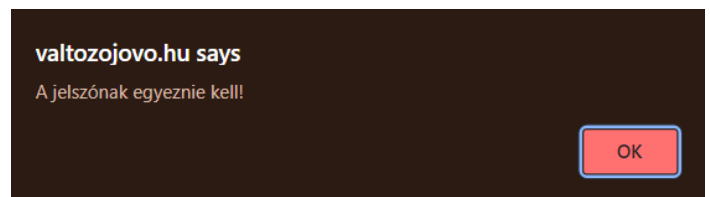
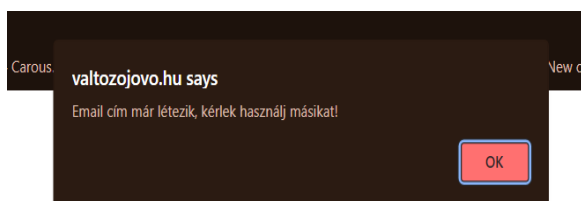


## Regisztráció/Bejelentkezés PHP/JS

A regisztrációt PHP segítségével oldottuk meg és azon adatokat PHPMyAdmin-ban tároltuk el.

A felhasználó regisztrációnál megadja adatait, majd teljesítve a captcha feltételeket utána lehetősége van regisztrálni.

Ha valamelyik feltétel hibás vagy nem teljesül akkor hibaüzenetet küldünk a felhasználó felületére



## Adományozás PHP/JS

Az adományozás résznél szintén PHP-t és PHPMyAdmin-t használtunk, de emellett JavaScriptet is.

Természetesen ha a felhasználó sikeresen adományoz akkor hasonló üzenetet kap, mint a regisztráció/bejelentkezés-nél.

A kártya és utalás radio button úgy lett megoldva, hogy ha a felhasználó rákattint az utalásra az adatok kitöltése után akkor PHP segítségével eltároljuk az adatokat majd kinyerjük az email-t egy változóba majd ezt felhasználva egy emailt generálunk a felhasználónak melyet a megadott email-re küldünk el.

Ha viszont a kártyás fizetést választjuk akkor a JavaScript segítségével az adatlap kibővül a kártya adatokkal melyet ilyenkor kötelező kitölteni és véglegesítésnél ez is az adatbázisba kerül.

Ezen kívül JavaScript-el még megszabjuk hogy csak szám karaktereket lehet használni és egyéb karaktereket nem jeleníti meg az input-ban. Az összeg, a CVC és a lejárat dátum használja ezt a funkciót.

A lejárat dátumnál használt JavaScript figyel arra hogy a dátum 23 és 26 között legyen az év és a hónap csak 1 és 12 között lehet hogy ne legyen hamis/téves információ

A kártya szám-nál szintén használtunk JavaScriptet ami megformázza és felismeri a kártyatípus. A kód szétválasztja egy kötőjellel minden negyedik karakter után és felismeri ha Visa vagy Mastercard a típusa

Ha nem egyezik meg valami akkor a jelentkezés gomb letiltása kerül és addig nem oldódik fel amíg minden adat nem jó

PHP segítségével az email input-nál meghatározom egy if else függvénnyel hogy melyik emailt töltöm fel. Ez azért van mert Ha a sütit elfogadjuk ami szintén JavaScript-el működik akkor az email eltárolódik a sütikben majd onnan kitölti az email inputot. Ezáltal két féle email-nek kell lennie

Egy a süti vagy kézi email és a bejelentkezés után legenerált email amellyel kitöltjük szintén az inputot

## Jelentkezés/Táborozás PHP

Ahogy az adományozásnál vagy a regisztrációnál, itt szintén a programozási nyelvek ugyanazok.

A táborozásnál is hasonló funkciókat használunk, mint az adományozásnál.

A csak szám funkciót egyedül az irányítószámnál használunk. Itt is hasonlóan működik az oldal. Ha minden adat sikeres akkor egy email-t küldünk ki a felhasználónak PHP-n keresztül.

## Felhasználói felület PHP

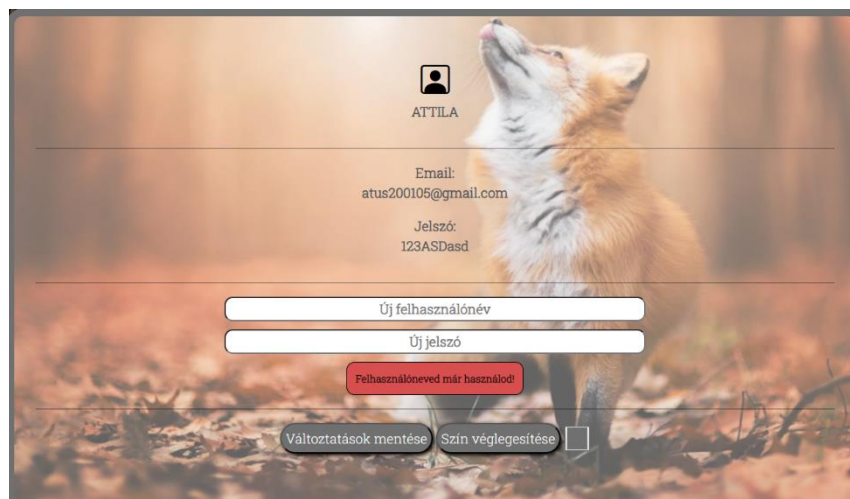
A felhasználói felületet akkor érjük el ha a bejelentkezésnél a PHP hibátlanul fut végig.

Ez után lehetőség van a jelszó vagy a felhasználónév változtatására PHP segítségével.

Ha a PHP feltételeiben hiba történik akkor azoknál létrejön egy SESSION változó egy adott névvel amely szimbolizálja a hibát. Ehhez a változóhoz hozzáfűzünk egy üzenetet.

Ha a hiba bekövetkezik akkor Javascript-el animáljuk és időzítjük hogy az értesítés ne maradjon ott.

Ez a PHP funkció minden esetleges hibára figyel ami lehetséges.



## Automatikus kijelentkezés PHP/JS

Ez a funkció két részből áll.

Az egyik a javascript kód ami elindít egy időzítőt ami fél órára van állítva a másik a kijelentkezés PHP

Az időzítő mindaddig lefrissül amíg mozgás van a képernyőn. Amint ez nem történik meg az időzítő elindul.

Ha fél óra lejár akkor a JavaScript lefuttatja a kijelentkezés PHP-t és visszasdob a főoldalra egy üzenettel, mely azt mondja, hogy inaktivitás miatt kijelentkeztünk.

Természetesen van lehetőség a sima kijelentkezésre is ami egy gombhoz van kötve.

## Területek és táblázatai PHP/JS

Minden területhez hozzáfűztünk 2 gombot, amelyek a növényeket és állatokat jeleníti meg egy táblázatban.

A funkció úgy működik, hogy a JavaScript megnézi milyen id-t kaptak a gombok. A kód ezen felül még azt is figyeli melyik gomb van használatban és kikapcsolja, ha a másik gomb használatba kerül. Ez minden területhez használva van.

A kód miután megkapta az értékét az id-nek akkor lefuttatja a hozzá tartozó PHP kódot. A PHP ekkor lekéri az adatokat az adatbázisból és legenerál egy adott táblázatot majd azt feltölti a begyűjtött adatokkal.

ID	Faj	Megnevezés	Hely	Veszélyeztettség	Virágzás
1	Növények	Gyertyán Magaskőrös	Aggteleki Nemzeti Park	Nem védett	Nyár
2	Növények	Osztrák Sárkányfű	Aggteleki Nemzeti Park	Fokozottan védett	Tavaszi
3	Növények	Leánykökörcsin	Aggteleki Nemzeti Park	Nem védett	Tavaszi
4	Növények	Korai Fehér Szegfű	Aggteleki Nemzeti Park	Védett	Nyár
5	Növények	Kakasmandikó	Aggteleki Nemzeti Park	Védett	Nyár
6	Növények	Molyhos Tölgyes	Aggteleki Nemzeti Park	Nem védett	Tavaszi
7	Növények	Aggteleki Karszt	Aggteleki Nemzeti Park	Nem védett	Tavaszi
8	Növények	Tarka Nőszirm	Aggteleki Nemzeti Park	Nem védett	Tavaszi
9	Növények	Nadálytő	Aggteleki Nemzeti Park	Nem védett	Nyár
10	Növények	Nagy Pacsirtafű	Aggteleki Nemzeti Park	Védett	Tavaszi

### Süti kezelés/Cookie PHP/JS

A süti kezeléséhez szintén JavaScriptet használunk. A kóddal eltárolunk egy email változót amit titkosítunk az oldalon belül. Emellett a kód ügyel arra, hogy mikor jelenjen meg újra a süti ablak.

A kód úgy van beállítva, hogy tizenkettő óránként újra jelenjen meg a felhasználónak. Ha esetlegesen nem fogadja el a sütit akkor az email nem mentődik el és minden alkalommal mikor frissít az oldalon vagy megnyit egy oldat a weblapon belül akkor szintén feldobja a süti ablakot. Elfogadás esetén az email elmentődik ha azon a részen használja email címét a felhasználó ahol ez a funkció be van kapcsolva és a süti ablak nem jelenik meg az adott idő limitig.

Ez az oldal sütiket használ annak érdekében, hogy jobb felhasználói élményben részesüljön.

Elfogadás

Elutasítás



## Tesztelések

### Back to top/Oldal tetejére teszt

```
1 document.addEventListener("DOMContentLoaded", function() {
2   const backToTopButton = document.querySelector("#back-to-top");
3
4   window.addEventListener("scroll", scrollFunction);
5
6   function scrollFunction() {
7     if (window.pageYOffset > 100) { // Show back-to-top button when user scrolls down 300px from the top of the document
8       backToTopButton.classList.add("showbtn");
9       backToTopButton.classList.remove("hidebtn");
10      backToTopButton.style.visibility = "visible"; // show the button
11    } else { // Hide back-to-top button when user scrolls up to the top of the document
12      backToTopButton.classList.add("hidebtn");
13      backToTopButton.classList.remove("showbtn");
14      backToTopButton.style.visibility = "hidden"; // hide the button
15    }
16  }
17
18  backToTopButton.addEventListener("click", smoothScrollBackToTop);
19
20  function smoothScrollBackToTop() {
21    const targetPosition = 0; // scroll to the top of the document
22    const startPosition = window.pageYOffset;
23    const distance = targetPosition - startPosition;
24    const duration = 1; // scroll to the top of the document in 750 milliseconds
25    let start = null;
26
27    window.requestAnimationFrame(step);
28
29    function step(timestamp) {
30      if (!start) start = timestamp;
31      const progress = timestamp - start;
32      window.scrollTo(0, easeInOutCubic(progress, startPosition, distance, duration));
33      if (progress < duration) window.requestAnimationFrame(step);
34    }
35  }
36
37  function easeInOutCubic(t, b, c, d) {
38    t /= d/2;
39    if (t < 1) return c/2*t*t*t + b;
40    t -= 2;
41    return c/2*(t*t*t + 2) + b;
42  }
43
44  // Hide back-to-top button when user is already at the top of the document on page load
45  if (window.pageYOffset <= window.innerHeight) {
46    backToTopButton.classList.add("hide");
47  }
48 });
```

Első sorban meghívunk egy esemény figyelő funkciót, amely figyelembe veszi, hogy az oldal betöltése megtörtént. Ez után begyűjtjük az elemünk id-jét és ezt eltároljuk egy egyszer használatos változóban, majd ismét megfigyeljük az oldalon használt görgő sávot.

Létrehozunk egy metódust, ami figyeli az oldal görgő helyzetkedését. Ha az elem elér egy bizonyos távot akkor generálunk egy showbtn-t és eltávolítjuk a hidebtn osztályt egyaránt. Ezek az osztályok azért felelnek hogy az oldal tetejére a gomb megjelenik-e vagy sem a görgő

helyzetéhez igazodva. Az 'else' másnéven 'vagy' feltétel a 'ha' feltételek ellentétje, ahol a showbtn fog törlődni és a hidebtn fog regenerálódni.

A következő funkcióhoz, amely azért felel hogy gombnyomásra az oldal tetejére kerüljön a felhasználó ahhoz létrehozunk egy esemény figyelőt ismét ami a kattintásért felel.

Ebben a funkcióban készítünk öt darab változót melyek ezek:

- `const targetPosition = 0;` - Célpozíció
- `const startPosition = window.pageYOffset;` - Görgő helyzete
- `const distance = targetPosition - startPosition;` - Megtett távolság - Kezdőpont
- `const duration = 1;` - Időtartam. Ez felel milyen gyorsan ugrik fel az oldal tetejére
- `let start = null;` - Kezdő érték. Ezt akkor használjuk mikor az oldal tetejére érünk gombnyomás után.

Ezeket a változókat egy későbbi funkcióban használjuk fel ahol az animációját készítjük el a gombnak.

Az animációt úgy oldottuk meg, hogy megjelenéskor felugrik csúszás animációval, majd használat vagy helyzet esetén ellentétes animációval eltűnik.

## Táblázatok teszt

```
1 <?php
2 include("connection.php");
3
4 $sql = "SELECT * FROM orsegiallat";
5 $result = mysqli_query($conn, $sql);
6
7 if (mysqli_num_rows($result) > 0) {
8     echo "<table>";
9     echo "<tr>";
10    echo "<th>Faj</th>";
11    echo "<th>Megnevezés</th>";
12    echo "<th>Hely</th>";
13    echo "<th>Létszám</th>";
14    echo "<th>Veszélyeztettség</th>";
15    echo "</tr>";
16
17    // Output data of each row
18    while($row = mysqli_fetch_assoc($result)) {
19        echo "<tr>";
20        echo "<td>" . $row["Faj"]. "</td>";
21        echo "<td>" . $row["Megnevezes"]. "</td>";
22        echo "<td>" . $row["Hely"]. "</td>";
23        echo "<td>" . $row["Letszam"]. "</td>";
24        echo "<td>" . $row["Veszelyezettseg"]. "</td>";
25        echo "</tr>";
26    }
27    echo "</table>";
28 } else {
29     echo "0 results";
30 }
31
32
33 mysqli_close($conn);
34 ?>
35
```

```
1 let currentTable = null;
2
3 function showTable(tableNum) {
4     if (currentTable === tableNum) {
5         return;
6     }
7
8     if (currentTable !== null) {
9         hideTable(currentTable);
10    }
11
12    fetch(`asset/PHP/table${tableNum}.php`)
13        .then(response => response.text())
14        .then(data => {
15            document.getElementById("data").innerHTML = data;
16            document.getElementById("data").style.display = "block";
17            setTimeout(function() {
18                document.getElementById("data").style.opacity = 1;
19            }, 10);
20            document.getElementById(`showTable${tableNum}Btn`).style.display = "none";
21            document.getElementById(`hideTable${tableNum}Btn`).style.display = "inline-block";
22
23            currentTable = tableNum;
24        });
25 }
26
27 function hideTable(tableNum) {
28     document.getElementById("data").style.opacity = 0;
29     document.getElementById("data").style.display = "none";
30     document.getElementById(`showTable${tableNum}Btn`).style.display = "inline-block";
31     document.getElementById(`hideTable${tableNum}Btn`).style.display = "none";
32
33     currentTable = null;
34 }
35
```

A JavaScript kódban (jobb oldal) megnézzük hogy a tábla neve és a php fájlnek adott név megegyezik-e. Ha igen akkor a data nevű id értékét bekérjük és megjelenítjük a felhasználó részére. Viszont ha a folyamat során ez nem történne meg akkor a táblázat nem jelenik meg.

Mindazonáltal ennek a funkciónak van egy eltüntetés része, hiszen egy gomb segítségével jelenítjük meg a táblázatainkat.

A PHP kódban (bal oldal) a táblázat adatainak lekérése megy végbe. Először a szerverhez kapcsolódunk, melyet a connection.php fájlban tárolunk. Ezután lekérjük az adatbázisunkból bizonyos adatokat vagy az egészet, majd megvizsgáljuk, hogy tartalmaz-e adatokat. Ha igen akkor generálunk egy table html elemet és ezeket feltöltjük az adott adatbázishoz tartozó adatokkal

## Inaktivitás mérő

```
1 var inactivityTime = 0;
2 // var logoutTime = 0.1667; // set logout time in minutes
3 var logoutTime = 30;
4 var logoutUrl = "logout2.php"; // set logout URL
5
6 // Start the inactivity timer
7 function startInactivityTimer() {
8     document.addEventListener("mousemove", resetTimer);
9     document.addEventListener("keypress", resetTimer);
10    document.addEventListener("touchmove", resetTimer);
11    resetTimer();
12 }
13
14 // Reset the inactivity timer
15 function resetTimer() {
16     inactivityTime = 0;
17 }
18
19 // Check for inactivity and log out the user if necessary
20 function checkInactivity() {
21     setInterval(function() {
22         inactivityTime++;
23         if (inactivityTime >= logoutTime * 60) { // check if logout time has been reached
24             window.location.href = logoutUrl; // redirect to logout URL
25         }
26     }, 1000);
27 }
28
29 // Check if the user is logged in and start the inactivity timer if necessary
30 if (typeof loggedIn !== 'undefined' && loggedIn) {
31     startInactivityTimer();
32     checkInactivity();
33 }
```

```
1 <?php
2 session_start();
3 session_unset();
4 session_destroy();
5 echo "<script>";
6 echo "window.location = 'index.php#popup'";
7 echo "</script>";
8 ?>
9
```

JavaScript rész(felső) ez a funkció akkor lép érvénybe, amikor a felhasználó bejelentkezve van. A funkció megfigyeli hogy van-e aktivitás jelenpillanatban. Kreálunk 3 változót, az első inaktivitás időzítő, és az érték, amikor a felhasználó ki lesz jelentkezettve, amelyek percben

vannak megadva. A harmadik pedig egy url, ami a kijelentkezés logout2.php-ra hivatkozik. A következő funkció, azért felel, hogyha kattintás, billentyűlenyomás, vagy egérmozgatás történik, akkor meghívja a resetTimer funkciót.

A resetTimer funkció az inaktivitásmérőt lenullázza aktivitás esetén. A checkInactivity funkció, amennyiben inaktivitásidő eléri a félórát, akkor a felhasználót kijelentkezteti.

Ha be van jelentkezve, akkor elindítja az időzítőt.

Php rész(alsó) amikor elindul a folyamat, ami az összes mentett elemet összegyűjti, majd megsemmisíti és egy JavaScript alert funkcióban eltárolt üzenet meghívásra kerül, amelyet a felhasználó felületen megjeleníti, majd a bejelentkezési felületre továbbítjuk.

### Fejlesztési lehetőségek

- Nagyító: A nagyító opció kényelmesebbé tenné az idősebbek, vagy a rosszul látók számára a weboldalt.
- GYK: Az emberekben rengeteg kérdés felmerül egy adott témával kapcsolatban, a GYK opció némelyikre gyors választ adhat.
- Fórum: Egy fórum lehetőséget biztosít arra, hogy ha kérdés merülne fel, akkor azt fel lehessen tenni, és egy a kérdésben jártasabb ember azt megválaszolhassa, de remek lehetőséget ad szimpla kommunikációra is.
- Környezettudatossági tippek: Ez a rész a weboldalon egy útmutatót adna azok számára, akik szeretnének zölden élni, vagy tudatosabban.
- Színpaletta: Ha valaki személyre szeretné szabni a weboldal színösszetételét, akkor ezzel az opcióval megteheti.
- Ökológiai lábnyom Számláló: Ez a számláló egyszerűen megmutathatná, hogy egy adott ember mekkora ökológiai lábnyom szerint él, rendszerezné az étkezési és a vásárlási, fogyasztási szokást, a közlekedési eszköz használatát, és a háztartásunk energia és hulladékfogyasztását.
- Kapcsolatok: Ez a funkció összegezné az ország összes természetvédő csoportját, illetve közösségi terét, ezáltal ha valaki szeretne egy a hozzá közel lévő környezetvédő közösség tagja lenni, akkor itt tud keresni.
- Tudástár: A környezeti neveléssel, illetve életmóddal kapcsolatos praktikumok.