

# ZÁRÓDOLGOZAT

BUDAPESTI KOMPLEX SZAKKÉPZÉSI CENTRUM  
WEISS MANFRÉD  
TECHNIKUM, SZAKKÉPZŐ ISKOLA ÉS KOLLÉGIUM

**FOCUSDOG.HU**

**Témavezető :**  
Kovács László Bálint

**Készítette :**  
Novák Noémi,  
Megyeri Réka

BUDAPEST  
2025

# TARTALOM

<b>TARTALOM.....</b>	<b>3</b>
<b>BEVEZETÉS.....</b>	<b>5</b>
Záródolgozatunk témája.....	5
Záródolgozatunk funkciói.....	5
Miért egy kutyaiskola?.....	6
A 'focusdog.hu' domain név és a szlogen.....	6
Kiknek szánjuk a programot?.....	7
Miben más mint a többi kutyaiskola?.....	7
<b>FEJLESZTŐI DOKUMENTÁCIÓ.....</b>	<b>8</b>
Használt szoftverek az elkészítéshez.....	8
Használt programozási nyelvek.....	9
Fejlesztői hardver.....	9
Az adatbázis.....	10
A 'user' tábla.....	11
A 'dog' tábla.....	12
Az 'oktatok' tábla.....	12
A 'login' tábla.....	13
A 'curzus' tábla.....	13
Az 'arak' tábla.....	14
A 'resztvevo' tábla.....	14
A 'tanfolyam' tábla.....	15
Az 'ertekeles' tábla.....	15
A 'naplo' tábla.....	16
Algoritmusok a weboldalon.....	17
Regisztráció.....	17
Bejelentkezés.....	22
Jelentkezés a tanfolyamokra.....	25
Adatmódosítás.....	33
Adminisztrátori felület.....	38
Admin Felület Elrendezése.....	38

Táblák és Adatkezelés.....	39
Jogusultság kezelése.....	40
A weboldal dizájn-ja.....	41
TESZT DOKUMENTÁCIÓ.....	42
Üres mezők kezelése.....	43
Regisztráció.....	43
Adatmódosítás.....	44
Értékelés.....	44
Duplikált felhasználói adatok szűrése.....	45
Regisztráció.....	45
Értékelés korlátozása.....	46
Fejlesztési lehetőségek.....	47
Mobilbarát alkalmazás fejlesztése.....	47
Webshop integrálása.....	47
<b>FELHASZNÁLÓI DOKUMENTÁCIÓ.....</b>	<b>47</b>
A program célja és lényegesebb funkciói.....	48
Szükséges hardvereszközök és szoftverek.....	49
Telepítés és indítás.....	49
Részletes bemutatás.....	50
Hibajelzések.....	53
Regisztráció.....	53
Bejelentkezés.....	54
Tanfolyamra jelentkezés.....	54
Adatmódosítás.....	54
Értékelés.....	55
Adminisztrátori felhasználó lehetőségei.....	56
Információkérés.....	57
<b>IRODALOMJEGYZÉK.....</b>	<b>58</b>
<b>ÖSSZEGZÉS.....</b>	<b>59</b>
<b>KÖSZÖNETNYILVÁNÍTÁS.....</b>	<b>60</b>

# **BEVEZETÉS**

## **Záródolgozatunk témája**

A záródolgozatunk témája egy olyan weboldal fejlesztésére fókuszál, amely egy kutyaiskola online jelenlétét biztosítja. A weboldal célja, hogy segítsen bemutatni a kutyaiskola szolgáltatásait, programjait, valamint lehetőséget biztosítson az érdeklődők számára a különböző képzési lehetőségek megismerésére. Az oldal tartalmát úgy terveztük, hogy átfogó képet adjon a kutyaiskola filozófiájáról, és oktatási módszereiről.

## **Záródolgozatunk funkciói**

A záródolgozatunk egy kutyaiskola weboldalának megvalósítása, amely bemutatja az iskola szolgáltatásait, kurzusait, oktatóit és kutyáikat. Az oldalon szerepelnek az iskola bemutatkozó információi, a különböző tanfolyamok és azok ára, valamint a kutyaiskola galériája. A felhasználók regisztrálhatnak, bejelentkezhetnek, és többféle tanfolyamra is jelentkezhetnek. Emellett lehetőség van véleményt írni a kutyaiskoláról. Az oldal könnyen elérhetővé teszi a kutyaiskola szolgáltatásait és információit.

## Miért egy kutyaiskola?

A kutyák iránti elköteleződés és szeretet mindig is fontos szerepet játszott, ami nagymértékben befolyásolta a témaválasztásunkat is. Sok kutya tulajdonos keres online segítséget kutyája neveléséhez, és egy jól megtervezett weboldal lehetőséget nyújt arra, hogy a gazdák egyszerűen tájékozódhassanak, beiratkozhatnak tanfolyamokra, és minden fontos információt gyorsan elérhessenek. Különösen fontos számunkra, hogy egy olyan online platformot hozzunk létre, amely a kutyaiskola számára nemcsak előnyöket biztosít, hanem a gazdák számára kényelmes és könnyen használható módot kínál a szolgáltatások elérésére.

## A 'focusdog.hu' domain név és a szlogen

A legtöbb kutya iskolánál az alapelv az a kutya nevelése. Természetesen ez is fontos, azonban mi úgy gondoljuk, hogy a tanítás sokkal élvezhetőbb ha főként a kutya és gazdája kapcsolata van előtérbe helyezve, és erősítve. Innen is ered a **focus** név, amely az egymásra való odafigyelést, és az összhangot jelképezi. Szlogenünk is így jött létre, mely így szól : **“Fókuszban a közös összhang”**.

## **Kiknek szánjuk a programot?**

A weboldalt elsősorban azok számára készítettük, akik kutyájuk nevelésében szeretnének segítséget kapni, legyen szó kezdő vagy haladó gazdiktól. Az oldal célja, hogy könnyen elérhető információkat és tanfolyamokat biztosítson mindazoknak, akik szeretnék jobban megérteni és képezni kutyájukat. Legyen szó kölyökkutya neveléséről, vagy a viselkedési problémák kezeléséről, a weboldal minden kutyatulajdonos számára hasznos és praktikus megoldásokat kínál.

## **Miben más mint a többi kutyaiskola?**

A Focus kutyaiskola kiemelkedik a többi kutyaiskola közül, mivel nem csupán a kutyák alapvető engedelmességi készségeire összpontosít, hanem a kutyák és gazdáik közötti kapcsolat erősítésére is. Az egyik legfontosabb különbség, hogy nálunk nem csupán a kutya tanul, hanem a gazdi is aktívan részt vesz a képzésben. Kutyaiskolánk különös figyelmet fordít a korosztályok szerinti oktatásra is, ami egyedülálló. Az egyes korcsoportok számára testreszabott tanfolyamok lehetővé teszik, hogy minden kutya és gazdi az életkoruknak, temperamentumuknak megfelelő módon fejlődhessenek. Ezáltal minden kutya a legjobb fejlődési lehetőséget kapja, a gazdik pedig megtanulják, hogyan támogathatják kutyájukat életük különböző szakaszaiban. Mi nem csak tanítunk, hanem érzelmi támogatást is biztosítunk.

# FEJLESZTŐI DOKUMENTÁCIÓ

## Használt szoftverek az elkészítéshez



### Windows 10

Windows 10 operációs rendszert használtunk a záródolgozatunk, és a vizsgaremekünk elkészítéséhez.



### Visual Studio Code

A kezdetek óta ezt a kódszerkesztőt használjuk programozáshoz, így a vizsgaremekhez is. Sok kiegészítő könnyítette meg a kód írást.



### phpMyAdmin / MySQL

Ezt az adatbázis kezelő szoftvert ismerjük a legjobban, ezért a vizsgaremekhez is ezt használtuk.



### Google Drive

Itt tudtuk megosztani egymással a készítési folyamat alatt álló fájlokat. Emellett sokadik mentési opcióként is szolgált.



### Canva

A logók és az adatbázis grafikai tervezésében is ezt a szoftvert használtuk, illetve sok kisebb ikont is felhasználtunk azokból amiket tartalmaz.



### Github

A Github platformját is használtuk, mivel lehetővé tette a kódmegosztást és az együttműködést.



# Használt programozási nyelvek



## HTML

A weboldalunk szerkezetének és megjelenésének megtervezésére használtuk, mint például szövegek, képek, linkek, űrlapok és egyéb tartalmak.



## CSS

Ezzel a nyelvvel a weboldalunk megjelenését / stílusát, alakítottuk ki, és formáztuk a különböző HTML elemeket.



## JavaScript

A weboldalunk interaktív részeinek, dinamikus műveleteinek megalkotásában használtuk ezt a nyelvet.



## MySQL

A weboldalunk adatainak tárolására és kezelésére, illetve az adatok táblázatokban való tárolására, módosítására használtuk ezt az adatbázis-kezelő rendszert.



## PHP

A weboldalunk adatainak feldolgozására, az adatbázissal való kommunikálásra és a HTML tartalmak dinamikus generálására használtuk ezt a szerveroldali programozási nyelvet.

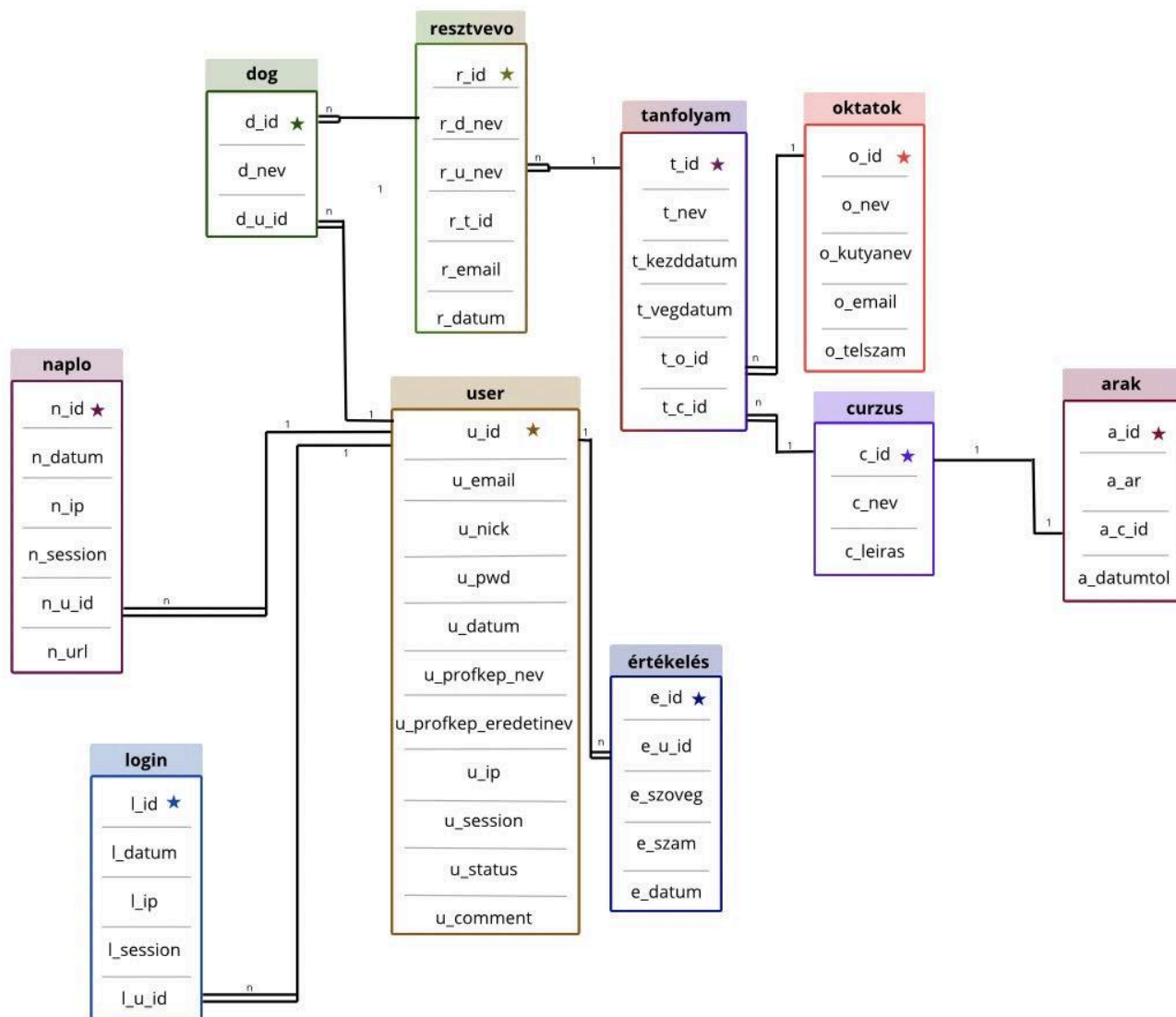
## Fejlesztői hardver

**Processzor :** Intel(R) Core(TM) i7-3517U CPU @ 1.90GHz 2.40 GHz

**Memória mérete :** 8,00 GB

**Rendszer típusa :** 64 bites Windows 10 operációs rendszer, x64-alapú processzor

# Az adatbázis



A focusdog.hu adatbázisa, amely 10 táblából áll.

## A 'user' tábla

A 'user' tábla eltárolja az összes, weboldalunkra regisztrált felhasználót, és adatait.

**u\_id** : A regisztrált felhasználók sorszáma, és a tábla elsődleges kulcsa. Ez az azonosító lehetővé teszi a felhasználó azonosítását más táblákban.

**u\_email** : A regisztrált felhasználók e-mail címét tárolja el. Az e-maileket erre az e-mail címre küldjük ki.

**u\_nick** : A regisztrált felhasználók felhasználó nevei, amit a regisztráció során megadtak.

**u\_pwd** : A regisztrált felhasználók beállított jelszavai, md5 titkosítással eltárolva.

**u\_datum** : A felhasználók regisztrációinak dátuma, év-hónap-nap óra:perc:másodperc pontossággal.

**u\_profkep\_nev** : A felhasználó által beállított profilkép neve, mely egy randomstring() függvény segítségével lett generálva.

**u\_profkep eredetinev** : A felhasználó által beállított profilkép eredeti fájl neve.

**u\_ip** : A regisztrált felhasználók IP címét tárolja el.

**u\_session** : Az adott felhasználó munkamenetéhez kapcsolódó adatokat tárolja el.

**u\_status** : A regisztrált felhasználók állapotát jelzi, aktív vagy passzív formában.

**u\_comment** : A regisztrált felhasználókhoz társított megjegyzéseket tárolja el.

user	
u_id	★
u_email	
u_nick	
u_pwd	
u_datum	
u_profkep_nev	
u_profkep eredetinev	
u_ip	
u_session	
u_status	
u_comment	

## A 'dog' tábla

A 'dog' tábla eltárolja az összes, felhasználó által regisztrált kutyát, és adatait.

**d\_id** : A regisztrált kutya sorszáma, és a tábla elsődleges kulcsa. Ez az azonosító lehetővé teszi a kutyák azonosítását más táblákban.

**d\_nev** : A regisztrációkor beírt kutya nevét tárolja el.

**d\_u\_id** : A regisztrált kutyához tartozó felhasználó sorszáma, a felhasználó id-t a 'user' táblában tároljuk.

dog	
d_id	★
d_nev	
d_u_id	

## Az 'oktatok' tábla

Az 'oktatok' tábla eltárolja a kutyaiskola oktatóit, és adataikat.

**o\_id** : Az iskola oktatóinak sorszáma, és a tábla elsődleges kulcsa. Ez az azonosító lehetővé teszi az oktatók azonosítását más táblákban.

**o\_nev** : Az iskola oktatóinak nevét tárolja el.

**o\_kutyanev** : Az iskola oktatóinak kiképző kutyáinak nevét tartalmazza.

**o\_email** : Az oktatók egyik fő elérhetőségét, az e-mail címüket tárolja el.

**o\_telszam** : Az oktatók második elérhetőségét, a telefonszámukat tartalmazza.

oktatok	
o_id	★
o_nev	
o_kutyanev	
o_email	
o_telszam	

## A 'login' tábla

A 'login' tábla eltárolja az összes felhasználó bejelentkezését.

**l\_id** : A bejelentkezések sorszáma, és a tábla elsődleges kulcsa.

**l\_datum** : A felhasználó bejelentkezésének dátuma, év-hónap-nap  
óra:perc:másodperc pontossággal.

**l\_ip** : A bejelentkezések IP címét tárolja el.

**l\_session** : Az adott felhasználó munkamenetéhez kapcsolódó adatokat  
tárolja.

**l\_u\_id** : A bejelentkezett felhasználó sorszáma, a felhasználó id-t a  
'user' táblában tároljuk.

login	
l_id	★
l_datum	
l_ip	
l_session	
l_u_id	

## A 'curzus' tábla

A 'curzus' tábla eltárolja a kutyaiskola 6 darab kurzusát, és azok  
adatait.

**c\_id** : Az iskola kurzusainak sorszáma, és a tábla elsődleges  
kulcsa. Ez az azonosító lehetővé teszi a kurzusok azonosítását  
más táblákban.

**c\_nev** : A kurzusok nevét tárolja el.

**c\_leiras** : A kurzusokról való információkat, tájékoztatásokat  
tartalmazza.

curzus	
c_id	★
c_nev	
c_leiras	

## Az 'arak' tábla

Az 'arak' tábla eltárolja a kutyaiskola 6 darab kurzusának árát, és érvényességi idejüket.

**a\_id** : A kurzusok árainak sorszáma, és a tábla elsődleges kulcsa.

**a\_ar** : A kurzusok árát tárolja el Ft-ban.

**a\_c\_id** : Az árhoz tartozó adott kurzus sorszáma, a kurzus id-t a 'curzus' táblában tároljuk.

**a\_datumtol** : Az árak érvényességének kezdeti dátuma, év-hónap-nap óra:perc:másodperc pontossággal.

arak	
a_id	★
a_ar	
a_c_id	
a_datumtol	

## A 'resztvevo' tábla

A 'resztvevo' tábla eltárolja a kutyaiskola tanfolyamaira jelentkezett résztvevőket.

**r\_id** : A résztvevők sorszáma, és a tábla elsődleges kulcsa. Ez az azonosító lehetővé teszi a tanfolyamokon résztvevők azonosítását más táblákban.

**r\_d\_nev** : A résztvevő kutya nevét tárolja el.

**r\_u\_nev** : A résztvevő felhasználó nevét tárolja el.

**r\_t\_id** : A tanfolyam sorszáma amelyre jelentkeztek a résztvevők , a tanfolyam id-t a 'tanfolyam' táblában tároljuk.

**r\_email** : A tanfolyamra jelentkezett résztvevők email címe.

**r\_datum** : A tanfolyamra jelentkezés időpontja, év-hónap-nap óra:perc:másodperc pontossággal.

resztvevo	
r_id	★
r_d_nev	
r_u_nev	
r_t_id	
r_email	
r_datum	

## A 'tanfolyam' tábla

A 'tanfolyam' tábla eltárolja a kutyaiskola tanfolyamait, és azok adatait.

**t\_id** : A tanfolyamok sorszáma, és a tábla elsődleges kulcsa. Ez az azonosító lehetővé teszi a tanfolyamok azonosítását más táblákban.

**t\_nev** : A tanfolyamok nevét tárolja el.

**t\_szint** : A tanfolyamok szintjét tárolja el, ez lehet kezdő vagy haladó.

**t\_kezddatum** : A tanfolyamok kezdési dátumát tartalmazza, év-hónap-nap óra:perc:másodperc pontossággal.

**t\_vegdatum** : A tanfolyamok befejezési dátumát tartalmazza, év-hónap-nap óra:perc:másodperc pontossággal.

**t\_o\_id** : A tanfolyam oktatójának sorszáma, az oktató id-t az 'oktatók' táblában tároljuk.

**t\_c\_id** : A kurzus tanfolyamának sorszáma, a kurzus id-t a 'kurzus' táblában tároljuk.

tanfolyam	
t_id	★
t_nev	
t_szint	
t_kezddatum	
t_vegdatum	
t_o_id	
t_c_id	

## Az 'ertekeles' tábla

Az 'ertekeles' tábla eltárolja a kutyaiskoláról írt véleményt, és azok értékelését 0-5-ig.

**e\_id** : Az értékelések sorszáma, és a tábla elsődleges kulcsa.

**e\_u\_id** : Az értékelést író felhasználó sorszáma, a felhasználó id-t a 'user' táblában tároljuk.

**e\_szoveg** : Az értékelés szövegét tárolja el.

**e\_szam** : Az értékelés értékét tárolja el számként.

**e\_datum** : Az értékelés elküldésének időpontja, év-hónap-nap óra:perc:másodperc pontossággal.

értékelés	
e_id	★
e_u_id	
e_szoveg	
e_szam	
e_datum	

## A 'naplo' tábla

A 'naplo' tábla eltárolja a weboldalon végrehajtott összes tevékenységet.

**n\_id** : A naplóbejegyzések sorszáma, és a tábla elsődleges kulcsa.

**n\_datum** : A naplóbejegyzések dátuma, év-hónap-nap  
óra:perc:másodperc pontossággal.

**n\_ip** : A naplóbejegyzések ip címét tárolja el.

**n\_session** : A naplóbejegyzésekhez tartozó munkamenetet tárolja el..

**n\_u\_id** : A naplóbejegyzésekhez tartozó felhasználó sorszáma, a  
felhasználó id-t a 'user' táblában tároljuk.

**n\_url** : A naplóbejegyzések url címét tárolja el.

naplo
n_id ★
n_datum
n_ip
n_session
n_u_id
n_url



# Algoritmusok a weboldalon

## Regisztráció

reg\_back.php, login.php

```
if($_POST['u_r_nick']== "")
    die("<script> alert('Nem adtál meg felhasználónevet!')</script>");
if($_POST['u_r_email']== "")
    die("<script> alert('Nem adtál meg e-mail címet!') </script>");
if($_POST['u_r_pwd']== "")
    die("<script> alert('Nem adtál meg jelszót!') </script>");

$usernamequery = mysqli_query($adb, "SELECT u_nick FROM user WHERE u_nick =
'$_POST[u_r_nick]'");
$emailquery = mysqli_query($adb, "SELECT u_email FROM user WHERE u_email
= '$_POST[u_r_email]'");
if(mysqli_num_rows($usernamequery) > 0){
    echo "<script> alert('Ez a felhasználónév már létezik!')</script>";
    print "<script> parent.location.href = './?p=login' </script>";
}
else if(mysqli_num_rows($emailquery) > 0){
    echo "<script> alert('Ez az e-mail cím már létezik!') </script>";
    print "<script> parent.location.href = './?p=login' </script>";
}
```

Az űrlap mezőknél a kód ellenőrzi, hogy a POST metódussal érkező adatokban megtalálhatóak-e a kötelező mezők: felhasználónév (u\_r\_nick), e-mail (u\_r\_email) és jelszó (u\_r\_pwd). Ha bármelyik mező hiányzik vagy üres, a regisztráció nem folytatódik, és egy Javascript alapú figyelmeztető üzenetet kap a felhasználó.

Ezt követően ellenőrzi, hogy a megadott felhasználónév vagy e-mail cím már létezik-e az adatbázisban. Ha a \$usernamequery vagy a \$emailquery lekérdezés eredménye tartalmaz sorokat (mysqli\_num\_rows(\$usernamequery) > 0), akkor figyelmeztető üzenetet kap a felhasználó, hogy az e-mail cím vagy a felhasználónév már létezik, majd visszairányít a regisztrációhoz.

```
$upwd = md5($_POST['u_r_pwd']);
```

```
mysqli_query($adb, "
    INSERT INTO user (u_id, u_email, u_nick, u_pwd, u_datum,
        u_profkep_nev, u_profkep_eredetinev, u_ip, u_session,
        u_status, u_comment)
    VALUES (NULL, '$_POST[u_r_email]', '$_POST[u_r_nick]', '$upwd',
        NOW(), '', '', '', '', '', '');
");
```

```
$d_uid = mysqli_insert_id($adb);
```

```
mysqli_query($adb, "
    INSERT INTO dog (d_id, d_nev, d_u_id)
    VALUES (NULL, '$_POST[u_r_dname]', '$d_uid');
");
```

Először a felhasználó által megadott jelszót az MD5 algoritmussal titkosítja, majd egy SQL **INSERT** lekérdezéssel elmenti az adatbázis **user** táblájába a felhasználó e-mail címét, felhasználónevét, titkosított jelszavát és a regisztráció dátumát. Ezután a `mysqli_insert_id($adb)` segítségével lekéri az újonnan beszúrt felhasználó azonosítóját, majd ezt felhasználva egy másik **INSERT** lekérdezéssel elmenti a **dog** táblába a felhasználó által megadott kutyanévet és a felhasználó azonosítóját, így összekapcsolva őket.

```

$user_name_result = mysqli_query($adb, "SELECT u_nick FROM user ORDER
    BY u_id DESC LIMIT 1");
$user_email_result = mysqli_query($adb, "SELECT u_email FROM user
    ORDER BY u_id DESC LIMIT 1");

$user_name_row = mysqli_fetch_assoc($user_name_result);
$user_email_row = mysqli_fetch_assoc($user_email_result);
$user_name = $user_name_row['u_nick'];
$user_email = $user_email_row['u_email'];

$from = "info@focusdog.hu";
$from_name = "Focus Kutyaiskola";
$subject = "Üdvözlünk a Focus Kutyaiskolánál!";

$message = "Kedves ".$_POST['u_r_nick'].",\n\n" .
    "Üdvözlünk a Focus Kutyaiskola közösségében!\n\n" .
    "Nagyon örülünk, hogy csatlakoztál hozzánk, és izgatottan
    várjuk, hogy együtt tanuljunk és fejlődjünk veled és a
    kutyusoddal.\n\n" .
    "Ha bármilyen kérdésed van, ne habozz írni nekünk!\n\n" .
    "Köszönjük, hogy a Focus Kutyaiskolát választottad!\n\n" .
    "Üdvözlettel,\n" .
    "A Focus Kutyaiskola Csapata";

```

A kód lekérdezi az adatbázisból a legutóbb regisztrált felhasználó felhasználónevét és e-mail címét a `user` táblából. Ezután a `mysqli_fetch_assoc()` függvénnyel kinyeri az eredményeket, majd a `$user_name` és `$user_email` változókba menti őket. Az e-mail küldéséhez beállítja a feladó e-mail címét (`$from`) és a feladó nevét (`$from_name`), valamint az e-mail tárgyát (`$subject`). Ezután létrehozza az e-mail tartalmát a `$message` változóban.

```

headers = "From: ".$from_name." <".$from.">". "\r\n";
$headers .= "Reply-To: ".$from." \r\n";
$headers .= "Content-type: text/plain; charset=UTF-8\r\n";

if(mail($_POST['u_r_email'], $subject, $message, $headers)) {
    echo "<script> alert('Sikeres regisztráció! Kérjük lépjen be.')
        </script>";
}
else {
    echo "<script> alert('Hiba történt az e-mail küldésekor.')
        </script>";
}

mysqli_close($adb);
print "<script> parent.location.href = './?p=login' </script>";

```

A kódban a regisztrációt követő köszöntő e-mail elküldése valósul meg a PHP `mail()` függvény segítségével. Először beállítja az e-mail fejlécét (`$headers`), amely tartalmazza a feladó nevét és e-mail címét, a válaszcímét, valamint a tartalom típusát és karakterkódolását. Ezután a `mail()` függvény segítségével elküldi az e-mailt a felhasználó által megadott e-mail címre, a korábban definiált tárggyal, üzenettel és fejléc információkkal. Ha a küldés sikeres, egy JavaScript `alert()` üzenetet jelenít meg a böngészőben a felhasználónak, amely tájékoztatja a sikeres regisztrációról és belépésre kéri. Ezt követően lezárja az adatbázis kapcsolatot a `mysqli_close($adb)` függvénnyel, majd átirányít a belépéshez.

```

<div class="form-box register">
  <form action="reg_back.php" method="POST">
    <h1>Regisztráció</h1>
    <div class="input-box">
      <input type="text" placeholder="Felhasználónév"
        name="u_r_nick" required>
      <i class='bx bxs-user'></i>
    </div>
    <div class="input-box">
      <input type="email" placeholder="Email"
        name="u_r_email" required>
      <i class='bx bxs-envelope'></i>
    </div>
    <div class="input-box">
      <input type="password" placeholder="Jelszó"
        name="u_r_pwd" required>
      <i class='bx bxs-lock-alt'></i>
    </div>
    <div class="input-box">
      <input type="text" placeholder="Kutyád neve"
        name="u_r_dname" required>
      <i class='bx bxs-dog'></i>
    </div>
    <button type="submit" class="btn">Regisztrálás</button>
  </form>
</div>

```

Ez a HTML kódrészlet egy felhasználói regisztrációs űrlapot jelenít meg. Az űrlap a `reg_back.php` fájlhoz küldi el az adatokat a POST metódussal. Négy kötelező mező van : egy felhasználónév (`u_r_nick`), egy e-mail cím (`u_r_email`), egy jelszó (`u_r_pwd`), valamint egy kutya név (`u_r_dname`). A felhasználó egy „Regisztrálás” feliratú gombbal tudja elküldeni az adatokat, ami elindítja az űrlap feldolgozását a `reg_back.php` oldalon. Az űrlap célja tehát az új felhasználók adatainak gyűjtése és továbbítása a háttérrendszer felé.

## Bejelentkezés

login\_back.php, login.php

```
$unick = $_POST['u_nick'];
$upwd = $_POST['u_pwd'];
$pw = md5($upwd);

$user = mysqli_query($adb, "
    SELECT * FROM user
    WHERE u_nick = '$unick'
    AND u_pwd = '$pw'
");

$userek = mysqli_fetch_assoc($user);

if (!$userek) {
    echo "<script> alert('Hibás belépési adat(ok)! Kérjük
        próbálja meg újra.') </script>";
    echo "<script> window.location.href = './?p=login'
        </script>";
    exit();
}
```

Ez a PHP kódrészlet egy bejelentkezési folyamat adatainak ellenőrzése. Először a `$_POST` tömbből beolvassa a felhasználónevet (`u_nick`) és a jelszót (`u_pwd`), majd a jelszót az `md5()` függvénnyel titkosítja. Ezután egy lekérdezést hajt végre a `user` táblán, amely megkeresi a felhasználót. A lekérdezés eredményét a `mysqli_fetch_assoc()` függvénnyel asszociatív tömbként tárolja. Ha a bejelentkezési adatok hibásak, akkor JavaScript segítségével megjelenít egy figyelmeztető ablakot a hibáról, majd visszairányítja a felhasználót a bejelentkezési oldalra (`./?p=login`). Az `exit()` parancs biztosítja, hogy a kód futása itt megszakadjon, és ne menjen tovább.

```

$_SESSION['u_id'] = $userek['u_id'];
$_SESSION['u_nick'] = $userek['u_nick'];

$ip = $_SERVER['REMOTE_ADDR'];
$sess = substr(session_id(), 0, 8);
if (isset($_SESSION['u_id'])) $uid = $_SESSION['u_id'];
else $uid = 0;

if (!empty($unick) && !empty($upwd) && $userek['u_nick'] == $unick &&
$userek['u_pwd'] == $pw) {
    echo "<script> alert('Sikeres belépés!') </script>";
    print "<script> window.location.href = './?p=adatlapom' </script>";
}
mysqli_query($adb, "
    INSERT INTO login (l_id, l_datum, l_ip, l_session, l_u_id)
    VALUES (NULL, NOW(), '$ip', '$sess', '$uid')
");

```

A kódrészlet egy sikeres bejelentkezés után menti a felhasználó legfontosabb adatait a munkamenetbe. Először a felhasználó azonosítóját (`u_id`) és nevét (`u_nick`) eltárolja a `$_SESSION` tömbben, így ezek az adatok később is elérhetőek lesznek. Ezután lekéri a felhasználó IP-címét a `$_SERVER['REMOTE_ADDR']` segítségével. A `session_id()` függvénnyel lekérdezi az aktuális munkamenet azonosítóját és eltárolja a `$sess` változóba. Ellenőrzi, hogy az aktuális session-ben van-e beállított `u_id` ha igen, akkor azt a `$uid` változóba menti, ellenkező esetben nullára állítja.

Az ezt követő kódrészlet először egy feltétellel ellenőrzi, hogy a felhasználónév (`$unick`) és a jelszó (`$upwd`) nem üresek-e, valamint hogy ezek megegyeznek-e az adatbázisból lekért értékekkel. Ha ezek igazak, akkor egy figyelmeztető üzenetet küld a felhasználónak, hogy a bejelentkezés sikeres volt, majd átirányítja őt az adatlapjára (`./?p=adatlapom`). Ezután beszúr egy sort a `login` adatbázis táblába, rögzítve a bejelentkezés idejét (`NOW()`), a felhasználó IP-címét (`$ip`), a session azonosítót (`$sess`), és a felhasználó azonosítóját (`$uid`).

```
<div class="form-box login">
  <form action="login_back.php" method="POST">
    <h1>Bejelentkezés</h1>
    <div class="input-box">
      <input type="text" placeholder="Felhasználónév" name="u_nick"
        required>
      <i class='bx bxs-user'></i>
    </div>

    <div class="input-box">
      <input type="password" placeholder="Jelszó" name="u_pwd"
        required>
      <i class='bx bxs-lock-alt'></i>
    </div>

    <button type="submit" class="btn">Belépés</button>
  </form>
</div>
```

Ez a HTML kódrészlet egy bejelentkezési űrlapot jelenít meg, amely lehetővé teszi a felhasználók számára, hogy belépjenek a weboldalra. Az űrlap a `login_back.php` fájlhoz küldi el az adatokat a **POST** módszerrel. Két bemeneti mező található: az első a felhasználónév (`u_nick`), a második pedig a jelszó (`u_pwd`). Mindkét mező **required**, tehát kötelező kitölteni őket. Az űrlap végén egy „Belépés” feliratú gomb található, amely elküldi az adatokat a háttérrendszerbe, ahol azokat a `login_back.php` fájl dolgozza fel.



## Jelentkezés a tanfolyamokra

jelentkezes\_back.php, focusdog-fooldal.php, focusdog.js

```
$dname = $_POST['d_j_nev'];
$stanfolyam = $_POST['tanfolyam-indulas'];
$stanfolyamnev = $_POST['tanfolyam-indulas_label'];
$sunev = $_POST['u_j_nev'];
$umail = $_POST['u_j_email'];
$cname = $_POST['kurzus_label'];

$stanf_result = mysqli_query($adb, "SELECT t_id, t_kezddatum,
    t_vegdatum FROM tanfolyamok WHERE t_nev = '$stanfolyam' ");
$stanf_row = mysqli_fetch_assoc($stanf_result);
$stanf_id = $stanf_row['t_id'];
$stanf_kezd = $stanf_row['t_kezddatum'];
$stanf_veg = $stanf_row['t_vegdatum'];

mysqli_query($adb, "
    INSERT INTO résztvevo ( r_id , r_u_nev , r_d_nev ,
        r_t_id, r_email , r_datum)
    VALUES ( NULL , '$sunev' , '$dname', '$stanf_id', '$umail',
        NOW()
    );
```

Ez a PHP kódrészlet egy tanfolyamra történő jelentkezés adatait dolgozza fel és rögzíti az adatbázisban. Először a `$_POST` tömbből lekéri az adatokat: a kutya nevét (`d_j_nev`), a tanfolyam azonosítóját (`tanfolyam-indulas`), a tanfolyam nevét (`tanfolyam-indulas_label`), a jelentkező nevét (`u_j_nev`), e-mail címét (`u_j_email`), valamint a kurzus nevét (`kurzus_label`). Ezután lekéri a `tanfolyamok` adatbázistáblából azt a tanfolyamot, amelyet a felhasználó kiválasztott, és eltárolja a változóba annak azonosítóját (`t_id`), kezdő dátumát (`t_kezddatum`) és záró dátumát (`t_vegdatum`). Végül egy `INSERT INTO` utasítással rögzíti a jelentkezést a `resztvevo` nevű táblában: eltárolja a résztvevő nevét, a kutya nevét, a kiválasztott tanfolyam azonosítóját, az e-mail címet, valamint a jelentkezés időpontját a `NOW()` függvényen.

```

$from = "info@focusdog.hu";
$from_name = "Focus Kutyaiskola";
$subject = "Jelentkezés visszaigazolása";

$message = "Kedves ".$sunev.",\n\n" .
    "A jelentkezésedet rögzítettük, szeretettel várunk a
    tanfolyamon:\n\n" .
    "Kurzus neve: ".$sname."\n" .
    "Dátum: ".str_replace('-', ' ', $tanf_kezd)." -
    ".str_replace('-', ' ', $tanf_veg)."\n" .
    "Tanfolyam szintje és időpontja: ".$stanfolyamnev."\n" .
    "Helyszín: 1033, Május 9. park (Óbudai sziget csúcsa)\n\n" .
    "További információk: Kérjük, tájékozódj a focusdog.hu
    weboldalon található " .
    "Információk menüpontban a szükséges részletekről.\n\n" .
    "Amennyiben bármilyen kérdésed lenne, vagy a tanfolyam előtt
    bármi változna, " .
    "kérjük, lépj kapcsolatba velünk.\n\n" .
    "Köszönjük, hogy minket választottál!\n\n" .
    "Üdvözlettel,\n" .
    "A Focus Kutyaiskola Csapata

";

```

Ez a PHP kódrészlet egy e-mail üzenet amelyet a sikeres jelentkezést követően kap meg a felhasználó. A kód elején beállítja az e-mail küldőjének címét (\$from), a feladó nevét (\$from\_name), valamint az e-mail tárgyát (\$subject). Ezután összeállítja a konkrét e-mail szöveget a \$message változóban. A jelentkező nevét (\$sunev) megszólításként használja. A szöveg részletes információkat ad a tanfolyamról: a kurzus nevét (\$sname), a kezdő és záró dátumát (\$tanf\_kezd, \$tanf\_veg), amelyeket szép formában jelenít meg a str\_replace függvénnyel, valamint a tanfolyam szintjét és időpontját (\$stanfolyamnev).

```

$headers = "From: ".$from_name." <".$from.">". "\r\n";
$headers .= "Reply-To: ".$from." \r\n";
$headers .= "Content-type: text/plain; charset=UTF-8\r\n";

if(mail($umail, $subject, $message, $headers)) {
    echo "<script> alert('Köszönjük hogy jelentkezted tanfolyamunkra!')
        </script>";
}
else {
    echo "<script> alert('Hiba történt az e-mail küldésekor.')
        </script>";
}

mysqli_close($adb);
print "<script> parent.location.href = './' </script>";

```

Ez a PHP kódrészlet egy automatikus e-mail elküldését végzi el. Először a `$headers` változóban beállítja az e-mail fejléceit: megadja a feladó nevét és e-mail címét, a válaszcímként szolgáló e-mail címet, valamint az üzenet formátumát és karakterkódolását. Ezután a `mail()` függvény segítségével elküldi az e-mailt a jelentkezőnek a megadott e-mail címre (`$umail`), a korábban összeállított tárggyal és üzenettel. Ha az e-mail sikeresen elküldésre került, a felhasználó egy felugró ablakban visszajelzést kap róla. Ha hiba történik a küldés során, akkor egy másik felugró ablak jelenik meg, amely figyelmezteti az elküldési hibára. A kód ezután lezárja az adatbáziskapcsolatot a `mysqli_close($adb)` paranccsal, majd átirányítja a felhasználót a főoldalra (`./`).

```

<form class="jelentkezes-form" method="POST" action="jelentkezes_back.php">
  <i class='bx bx-x' id="form-close"></i>
  <div class="jelentkezes-form-container">
    <h1 class="jelentkezes-h1">Jelentkezz a tanfolyamra</h1>

    <div class="jelentkezes-tanfdata">
      <div class="form-group2">
        <label for="kurzus" class="jelentkezes-label">Válaszd ki a
          tanfolyamot:</label>
        <select id="kurzus" name="kurzus" class="jelentkezes-select">
          <option value="kolyok-kurzus">Kölyökkutya kurzus</option>
          <option value="felnott-kurzus">Felnőtt kutya kurzus</option>
          <option value="idos-kurzus">Idős kutya kurzus</option>
          <option value="agility-kurzus">Agility kurzus</option>
          <option value="orzo-vedo-kurzus">Őrző-védő kurzus</option>
          <option value="egyeni-oktatas">Egyéni oktatás</option>
        </select>
        <input type="hidden" name="kurzus_label" value=""
          id="kurzus-label">

```

Ez a HTML kódrészlet a jelentkezés űrlapját valósítja meg. Az űrlap a **POST** metódust használja, a beküldés után a `jelentkezes_back.php` nevű PHP fájl dolgozza fel az adatokat. Az űrlap egyik legfontosabb része egy tanfolyamválasztó legördülő menü (`<select>` tag), amelyben a felhasználó kiválaszthatja, hogy melyik tanfolyamra szeretne jelentkezni. Minden egyes opció rendelkezik egy **value** attribútummal, amelyet az adatküldés során a szerver fog feldolgozni. Emellett szerepel egy rejtett mező (`<input type="hidden">`), amely nem látható a felhasználó számára, de JavaScript segítségével értékek vannak hozzárendelve, mint a kiválasztott tanfolyam szöveges címe, amelyet az e-mail elküldésekor használunk fel.

```
<label for="tanfolyam-indulas" class="jelentkezes-label">Tanfolyam  
szintje és időpontja:</label>
```

```
<select id="tanfolyam-indulas" name="tanfolyam-indulas"  
    class="jelentkezes-select">  
</select>  
<input type="hidden" name="tanfolyam-indulas_label" value=""  
    id="tanfolyamindulas-label">
```

```
<label for="oktato" class="jelentkezes-label">Oktató:</label>  
<select id="oktato" name="oktato" class="jelentkezes-select">  
</select>
```

Ez a HTML kódrészlet egy intelligens, dinamikusan frissülő űrlap része, a kiválasztott tanfolyam pontos részleteiről. Három fontos elemet tartalmaz: a tanfolyam indulásának idejét és szintjét, valamint az oktató személyét. A `label` elemek leíró szöveget biztosítanak az egyes mezőkhöz. Az első `select` mezőben a tanfolyam szintjét és indulási időpontját lehet kiválasztani, míg a második az oktató kiválasztására szolgál. Ezek a mezők jelenleg nem tartalmazzak `option` elemet a HTML-ben – ezek JavaScript segítségével kerülnek dinamikusan beillesztésre. Ez lehetővé teszi, hogy a megjelenő opciók mindig az aktuális választáshoz igazodjanak.

```
<div class="form-group">
  <label for="nev" class="jelentkezes-label">Neved:</label>
  <input type="text" id="jelentkezes-nev" name="u_j_nev"
    class="jelentkezes-input" required>

  <label for="email" class="jelentkezes-label">Email címed:</label>
  <input type="email" id="jelentkezes-email" name="u_j_email"
    class="jelentkezes-input" required>

  <label for="kutya-neve" class="jelentkezes-label">A kutyád
    neve:</label>
  <input type="text" id="jelentkezes-kutya-neve" name="d_j_nev"
    class="jelentkezes-input" required>
</div>

<button type="submit" class="jelentkezes-btn">Elküldöm a
  jelentkezést</button>
```

Ez a kódrészlet a tanfolyami jelentkezési űrlap azon részét tartalmazza, ahol a felhasználónak meg kell adnia a saját és kutyája alapvető adatait. A `form-group` nevű div-blokk csoportosítja az adatbekérő mezőket, amelyek mindegyike egy `label`-ből és egy hozzá tartozó `input` mezőből áll. Elsőként a jelentkező nevét (`u_j_nev`) kéri be a rendszer egy szövegmezőn keresztül. Ez a mező „required”, vagyis kötelezően ki kell tölteni. Ezt követi az email cím megadása (`u_j_email`) egy email típusú mezővel, amely automatikus szintaxis ellenőrzést is végez, azaz csak helyesen formázott email címet enged megadni. Ez a mező szintén kötelezően kitöltendő. Végül a kutya neve adható meg (`d_j_nev`) egy újabb szövegmezőn keresztül, amely szintén „required”, tehát enélkül sem lehet elküldeni az űrlapot. Az egész űrlap utolsó eleme egy beküldő gomb (`<button type="submit">`). Ha minden kötelező mező ki van töltve, a gomb megnyomásával az űrlap elküldi az összes adatot, amit a `jelentkezes_back.php` fájl feldolgoz.

```

const kurzusData = {
  "kolyok-kurzus": {
    indulas: [
      { value: "kolyok-kezdő", label: "Kezdő (2025.07.07. 10:00)" },
      { value: "kolyok-haladó", label: "Haladó (2025.07.14. 8:00)" }
    ],
    oktato: "Takács Boldizsár"
  },
  "felnott-kurzus": {
    indulas: [
      { value: "felnott-kezdő", label: "Kezdő (2025.07.28. 10:00)" },
      { value: "felnott-haladó", label: "Haladó (2025.07.21. 8:00)" }
    ],
    oktato: "Molnár Bernadett"
  },
  "idos-kurzus": {
    indulas: [
      { value: "idos-kezdő", label: "Kezdő (2025.08.04. 10:00)" },
      { value: "idos-haladó", label: "Haladó (2025.08.11. 8:00)" }
    ],
    oktato: "Nagy Ákos"
  },
  "agility-kurzus": {
    indulas: [
      { value: "agility", label: "Agility (2025.08.18. 12:00)" }
    ],
    oktato: "Tóth Kata"
  },
  "orzo-vedo-kurzus": {
    indulas: [
      { value: "orzo-vedo", label: "Őrző-védő (2025.08.04. 17:00)" }
    ],
    oktato: "Kovács Tünde"
  },
  "egyeni-oktatas": {
    indulas: [
      { value: "egyeni", label: "Egyéni oktatás (Időpontért keresse fel oktatóinkat)" }
    ],
    oktato: "Kiss Erik"
  }
};

```

```

kursusSelect.addEventListener('change', function() {
  const selectedKurzus = this.value;
  const selectedData = kurzusData[selectedKurzus];
  indulasSelect.innerHTML = '';
  selectedData.indulas.forEach(indulas => {
    const option = document.createElement('option');
    option.value = indulas.value;
    option.textContent = indulas.label;
    indulasSelect.appendChild(option);
  });
  oktatoSelect.innerHTML = '';
  const option = document.createElement('option');
  option.value = selectedData.oktato.toLowerCase().replace(/\s+/g, '-');
  option.textContent = selectedData.oktato;
  oktatoSelect.appendChild(option);
});
kursusSelect.dispatchEvent(new Event('change'));

```

Ez a két JavaScript kódrészlet együtt egy dinamikus jelentkezési űrlap működését valósítja meg. Az első kódrészlet egy JavaScript objektumot (**kurzusData**) definiál, amely strukturált módon tárolja a különböző tanfolyamokhoz tartozó indulási időpontokat (**indulas**) és az oktatókat (**oktato**). A második rész eseményfigyelőt kapcsol a tanfolyamválasztó legördülő listához (**kursusSelect**), amely minden választáskor kiolvassa a megfelelő adatokat, majd frissíti a tanfolyamindulási időpontokat (**indulasSelect**) és az oktató nevét (**oktatoSelect**) tartalmazó mezőket. A végén egy programozott eseményindítás (**dispatchEvent**) gondoskodik arról, hogy az űrlap betöltésekor az első tanfolyam adatai azonnal megjelenjenek. Így a felhasználó mindig csak az adott kurzushoz tartozó lehetőségekből választhat, ami egyszerűbbé és hibamentessé teszi a jelentkezést.



## Adatmódosítás

adatlap\_back.php, adatlap.php

```
if(!empty($_POST['felhasznev']) && !empty($_POST['felhaszmail'])) ){
    $u_id = mysqli_real_escape_string($adb, $_SESSION['u_id']);
    $sql = "SELECT * FROM user WHERE u_id = '$u_id'";
    $felhasznalok = mysqli_fetch_array(mysqli_query($adb, $sql));
    $sql_update_user = "
        UPDATE user
        SET u_email = '"' . mysqli_real_escape_string($adb,
            $_POST['felhaszmail']) . "',
            u_nick = '"' . mysqli_real_escape_string($adb,
            $_POST['felhasznev']) . "'
        WHERE u_id = '$u_id'
    ";
    mysqli_query($adb, $sql_update_user);
    $_SESSION['u_nick'] = $_POST['felhasznev'];
    echo "<script>alert('Adataidat sikeresen módosítottuk');</script>";
}
else{
    echo "<script>alert('Nem lehet üres a felhasználónév és az e-mail
        cím!');</script>";
    print "<script> parent.location.href = './?p=adatlapom' </script>";
}
```

Ez a kódrészlet a felhasználó adatainak (felhasználónév és e-mail cím) frissítését végzi el, miközben biztosítja, hogy a megadott adatok ne legyenek üresek. Először ellenőrzi, hogy mindkét mező ki van-e töltve (`!empty()`), és ha bármelyik üres, figyelmeztetést ad. Az eddig regisztrált felhasználók adatait a `$felhasznalok` változóba kéri le. Ezután egy új SQL lekérdezést generál, amely frissíti a felhasználó e-mail címét és felhasználónevét az adatbázisban az új értékekkel. Miután az adatbázis sikeresen frissült, az új felhasználónév (a `$_POST['felhasznev']` változó értéke) elmentésre kerül a session változóban.

```

$oldpwd = md5($_POST['oldpwd']);
if(!empty($_POST['oldpwd'])) {
    if($oldpwd == $felhasznalok['u_pwd']) {
        $sql_update_pw = "
            UPDATE user
            SET u_pwd = '"' . mysqli_real_escape_string($adb,
                md5($_POST['newpwd'])) . '"'
            WHERE u_id = '$u_id'
        ";
        mysqli_query($adb, $sql_update_pw);
        echo "<script>alert('Adataidat sikeresen módosítottuk');
            </script>";
    }
    else{
        echo "<script>alert('A megadott régi jelszó nem
megfelelő!');
            </script>" ;
        print "<script> parent.location.href = './?p=adatlapon'
            </script>";
    }
}
}

```

Ez a kódrészlet a felhasználó jelszavának módosítását végzi el. Először ellenőrzi, hogy a régi jelszó (oldpwd) mező ki van-e töltve a felhasználó által. Ha a mező tartalmaz adatot, a felhasználó által megadott régi jelszó (amit MD5 hash-el titkosítanak) összehasonlítódik az adatbázisban tárolt jelszóval (\$felhasznalok['u\_pwd']). Ha a két jelszó megegyezik, akkor az új jelszó (newpwd) szintén MD5 hash-el titkosítva frissíti a felhasználó jelszavát az adatbázisban. A sikeres módosítás után a kód egy JavaScript alert üzenetet jelenít meg, amely tájékoztatja a felhasználót a sikeres módosításról. Ha a régi jelszó nem megfelelő, a rendszer figyelmezteti a felhasználót egy hibaüzenettel, majd vissza irányítja az adatlapra, hogy kijavítsa a hibát.

```

if (isset($_POST['felhaszdog']) && !empty($_POST['felhaszdog'])) {
    $kutya_nev = mysqli_real_escape_string($adb,
        $_POST['felhaszdog']);
    $sql_update_dog = "
        UPDATE dog
        SET d_nev = '$kutya_nev'
        WHERE d_u_id = '$u_id'
    ";

    mysqli_query($adb, $sql_update_dog);
    $_SESSION['d_nev'] = $_POST['felhaszdog'];

    echo "<script>alert('Adataidat sikeresen módosítottuk');
        </script>";
}

print "<script> parent.location.href = './?p=adatlapom' </script>";
mysqli_close($adb);

```

Ez a kódrészlet a felhasználó kutyájának nevét módosítja az adatbázisban. Először ellenőrzi, hogy a `felhaszdog` nevű űrlapmező be van-e töltve. Ha igen, a kutya új neve (`$_POST['felhaszdog']`) biztonságosan kerül mentésre az adatbázisba, miután a `mysqli_real_escape_string()` biztosítja, hogy ne legyenek benne biztonsági rések. Ezt követően egy SQL lekérdezést futtat, amely frissíti a `dog` táblában az adott felhasználó kutyájának nevét a `$kutya_nev` változóval. Miután a módosítás megtörtént, az új kutyanevet elmenti a session-be, hogy a későbbiekben is elérhető legyen. Végül egy JavaScript alert üzenet jelenik meg, amely tájékoztatja a felhasználót a sikeres módosításról. A kód az adatbázis művelet befejeztével visszairányítja a felhasználót az adatlapjára (`parent.location.href = './?p=adatlapom'`). A végén bezárja az adatbázis kapcsolatot a `mysqli_close($adb)` függvénnyel.

```

<div class="info-item">
    <label for="felhasznnev">Felhasználónév:</label>
    <input type="text" id="felhasznnev" name="felhasznnev"
value="<?php
        print $felhasznalok['u_nick']; ?>" >
</div>
<div class="info-item">
    <label for="emailcim">E-mail cím:</label>
    <input type="email" id="emailcim" name="felhaszmail"
value="<?php
        print $felhasznalok['u_email']; ?>" >
</div>
<div class="info-item">
    <label for="dogname">Kutyája neve:</label>
    <input type="text" id="dogname" name="felhaszdog" value="<?php
print
        $dogs['d_nev']; ?>" >
</div>
<div class="info-item regijelszo">
    <label for="regijelszo">Régi jelszó:</label>
    <input type="password" id="regijelszo" name="oldpwd" >
</div>
<div class="info-item ujjelszo">
    <label for="ujjelszo">Új jelszó:</label>
    <input type="password" id="ujjelszo" name="newpwd" >
</div>

```

Ez a kódrészlet az adاتمódosítás HTML része. Az űrlap mezői előre kitöltődnek a korábban mentett adatokkal, amelyeket a PHP segítségével dinamikusan tölt be az adatbázisból lekért `$felhasznalok` és `$dogs` tömbökből. Az első mező a felhasználónév, a második mező az e-mail cím, a harmadik mező a kutya neve, amelyek mindegyike az adatbázistáblák mezőiből vannak kiírva. A következő két mező jelszavakhoz kapcsolódik: az egyik a régi jelszó (`oldpwd`), a másik pedig az új jelszó (`newpwd`), amit a felhasználó beállítani szeretne. A PHP-kód beágyazásával a mezők automatikusan kitöltődnek a meglévő adatokkal, így egyszerűbbé válik a módosítás.

```

<div class="button-container">

    <a href="/"><button type="button" class="back-button"> <i
class='bx
        bx-left-arrow'></i> Főoldal </button></a>

    <button type="submit" class="modify-button">Mentés</button>

    <?php
        $userek = mysqli_fetch_array(mysqli_query($adb, " SELECT
u_comment

            FROM user WHERE u_id = $_SESSION[u_id]"));

        if(isset($userek['u_comment']) &&
trim($userek['u_comment']) ===
            'admin'){
            echo"<a href='?p=admin'><button type='button'
                class='admin-button'>Admin
                felület</button><a/>";
        }
    ?>

    <a class="exit-button" href='?p=kilepes'>Kilépés<a/>
</div>

```

Ez a kódrészlet az adatlap HTML-gombjainak megvalósítása, amely különböző navigációs lehetőségeket biztosít. A bal oldali gomb egy visszalépési lehetőség a főoldalra, a középső gomb egy „Mentés” típusú gomb, amely elküldi az aktuális űrlapot (formot), míg a jobb oldali részen a kijelentkezés és — bizonyos esetben — az admin felület elérése jelenik meg. A PHP-rész egy adatbázis-lekérdezést hajt végre, amellyel a bejelentkezett felhasználóhoz tartozó `u_comment` mezőt kérdezi le. Ez a mező tartalmazza a jogosultságjelölést (`admin`). Ha az értéke pontosan „admin” (és nem üres vagy szóközös), akkor megjelenik egy extra gomb is, amellyel az admin felületre lehet navigálni. Végül mindig látható egy kilépés gomb („Kilépés”), amely kijelentkezteti a felhasználót.

## Adminisztrátori felület

A weboldal adminisztrátori felületének célja, hogy átlátható és könnyen kezelhető módot kínáljon az adminisztrátorok számára a kutyaiskola online platformjának irányításához. A felület lehetővé teszi a különböző adatbázis-táblák egyszerű kezelését és a fontos információk gyors elérését, beleértve a felhasználói adatokat, kutyákat, kurzusokat, tanfolyamokat és értékeléseket.

### Admin Felület Elrendezése

Az admin felület a **controlpanel.php** fájlban található, amely egy összesített nézetet biztosít az összes fontosabb adatbázis-táblához. Az egyes táblák külön-külön vannak beágyazva az `include()` függvény segítségével.

Az admin felületen az egyes táblák listázása egyetlen oldalra összpontosul, amely minden táblát külön `div` konténerbe helyez, és mindegyik táblát egy-egy új szakaszként jeleníti meg. Az adminisztrátorok egyszerűen navigálhatnak a különböző táblák között, és végezhetnek módosításokat az adatbázisban.

```
<div class="tablesdiv">
    <?php
        include("userstable.php");
    ?>
</div>

<div class="tablesdiv">
    <?php
        include("loginstable.php");
    ?>
</div>
```

## Táblák és Adatkezelés

Minden táblázat a szükséges adatokat jeleníti meg, valamint eszközöket biztosít az adatok módosítására vagy törlésére. Az adminisztrátorok egyszerűen kezelhetik a rendszer információit, gyorsan végezhetnek módosításokat, anélkül, hogy bonyolult navigációval kellene szembenézniük.

**Kurzusok :** A `curzustable.php` fájlban található táblázat a kurzusok kezelését teszi lehetővé.

**Felhasználók:** A `userstable.php` fájlban található táblázat a felhasználók kezelését teszi lehetővé.

**Értékelések:** Az `ratingstable.php` fájlban található táblázat az értékelések kezelését teszi lehetővé.

**Tanfolyamok:** A `programstable.php` fájlban található táblázatban a tanfolyamok kezelését teszi lehetővé.

**Kutyák:** A `dogstable.php` fájlban található táblázatban a kutyák kezelését teszi lehetővé.

**Árak:** A `pricestable.php` fájlban található táblázatban az árak kezelését teszi lehetővé.

A kódrészlet lekérdezi az adatokat az adatbázisból, és egy táblázatban jeleníti meg őket. A felsorolt táblák mindegyike ezen a kódrészleten alapszik.

```
if ($kurzusok->num_rows > 0) {
    while ($row = $kurzusok->fetch_assoc()) {
        echo "<tr>
            <td>".(!empty($row['c_id']) ? $row['c_id'] : '-')."</td>
            <td>".(!empty($row['c_nev']) ? $row['c_nev'] : '-')."</td>
            <td>".(!empty($row['c_leiras']) ? $row['c_leiras'] :
            '-')."</td>
            <td><i class='bx bxs-pencil' style='color: green;'></i></td>
            <td><i class='bx bxs-trash-alt' style='color: red;'></i></td>
        </tr>";
    }
}
```

## Jogosultság kezelése

Az admin felület jogosultságkezelése biztosítja, hogy csak azok a felhasználók férhessenek hozzá az adminisztrátori funkciókhoz, akik rendelkeznek megfelelő jogosultságokkal.

A kódban ellenőrzés történik a felhasználói adatokban található "u\_comment" mező értékére, amely azt jelzi, hogy a felhasználó adminisztrátori jogokkal rendelkezik-e. Amennyiben a felhasználó nem rendelkezik admin jogosultsággal, és próbál belépni az adminisztrátori felületre, a rendszer egy "404-es" hibaoldalra irányítja őt, így megakadályozva a jogosulatlan hozzáférést.

Ha a felhasználó rendelkezik admin jogosultsággal, akkor hozzáférést kap az admin felülethez, és betöltődik a `controlpanel.php` oldal. Ez a jogosultságkezelés biztosítja, hogy csak a megfelelő felhasználók férhessenek hozzá az érzékeny adminisztratív funkciókhoz, és védelmet nyújt a rendszer biztonsága érdekében.

Ha a felhasználó rendelkezik admin jogosultsággal, akkor egy admin felülethez vezető gomb jelenik meg számára.

```
$isAdmin = (isset($userek['u_comment']) && trim($userek['u_comment']) === 'admin');  
if ($p == "admin" && !$isAdmin) {  
    include("404es.php");  
}  
  
if ($isAdmin) {  
    include("kezelopult/controlpanel.php");  
}  
else {  
    include("404es.php");  
}  
  
if(isset($userek['u_comment']) && trim($userek['u_comment']) === 'admin'){  
    echo"<a      href='?p=admin'><button      type='button'  
class='admin-button'>Admin  
        felület</button><a/>";  
}  
}
```



## A weboldal dizájn-ja

A weboldal dizájnja fontos szerepet játszik a felhasználói élmény megteremtésében és a weboldal hatékonyságának növelésében. A projekt során a cél, hogy egy olyan modern, felhasználóbarát és esztétikailag vonzó felületet hozzunk létre, amely egyszerre figyel a vizuális élményre és a könnyű navigálhatóságra. A dizájn során figyelembe vettük az interakciókat is, és a reszponzív (mobilbarát) megoldásokat is.

A weboldal navigációs struktúrája egyszerű, mégis hatékony. A navbar, amely a fő navigációs elemeket tartalmazza, rugalmas elrendezést alkalmaz, így bármilyen eszközön jól működik. A gombok és linkek stílusa, valamint a felhasználói ikonok, amelyek a weboldal funkcióit kommunikálják, egyértelműek és könnyen érthetők. A hamburger menü biztosítja, hogy a kisebb képernyőkön is egyszerűen használható maradjon a navigáció.

A bejelentkezési és regisztrációs felület tervezése során arra összpontosítottunk, hogy a felhasználók könnyen navigálhassanak. A formázás egyszerűségével és a könnyen kezelhető mezőkkel javítjuk a felhasználói élményt. A színskombinációk – mint a #895159 árnyalat – barátságos és vonzó megjelenést biztosítanak.

A reszponzív dizájn kiemelt szempont volt, hiszen fontos volt, hogy a weboldal minden eszközön – legyen az asztali számítógép vagy okostelefon – megfelelően jelenjen meg. A média lekérdezések (@media query) alkalmazásával biztosítottuk, hogy a weboldal különböző képernyőméretekhez alkalmazkodjon.

A színválasztás az egész weboldalon átgondolt és tudatos döntés volt. A meleg, földszínek és pasztell tónusok, mint a #895159 és a #ffff5, barátságos légkört teremtenek. A figyelemfelkeltő hover színek kiemelik a fontos elemeket, elősegítve ezzel a felhasználói interakciót. A szövegek, címek és gombok szintén kellően kiemelve, így a felhasználók számára egyértelművé válik a weboldal navigációja és struktúrája.

# TESZT DOKUMENTÁCIÓ

A weboldal tesztelése során a rendszer különböző platformokon és eszközökön került kipróbálásra annak érdekében, hogy biztosítsuk a funkcionalitás, a felhasználói élmény és a teljesítmény minden környezetben történő helyes működését.

A tesztelés a következő eszközök és környezetek felhasználásával zajlott:

**Asztali számítógép** : A tesztelés egyik környezete egy Windows 10 operációs rendszert futtató asztali számítógép volt, amely biztosította a weboldal teljes funkcionalitásának tesztelését nagyobb képernyőméreten.

**Hordozható számítógép** : A tesztelés másik környezete egy szintén Windows 10 operációs rendszert futtató laptop volt, amely biztosította a weboldal mobilitásának és alkalmazkodásának tesztelését.

**Böngésző** : A tesztelés során a legfrissebb stabil verziójú Google Chrome böngészőt használtuk, amely széleskörű támogatást nyújt a legújabb webes technológiákhoz. A böngésző biztosította az oldal megjelenésének és interaktivitásának zökkenőmentes működését.

**Mobiltelefon** : A weboldal mobil kompatibilitását egy IOS alapú mobiltelefon tesztelte, amelyen a Google mobilböngészője futott. A tesztelés célja annak biztosítása volt, hogy a weboldal megfelelően alkalmazkodjon a kisebb képernyőméretekhez is.

A teszteléshez létrehozott tesztfelhasználó adatai a következők (minden jogosultság megadva):

**Felhasználónév** : focus

**Jelszó** : Focusdog

## Üres mezők kezelése

A weboldal tesztelése során több helyen is felmerült, hogy a rendszer nem kezeli megfelelően az üresen hagyott mezőket, így ezek ellenőrzésére külön figyelmet fordítottunk.

### Regisztráció

Ha a felhasználónév (`u_r_nick`), az e-mail cím (`u_r_email`) vagy a jelszó (`u_r_pwd`) mező üresen marad, akkor egy JavaScript-es felugró ablakban figyelmeztető üzenetet jelenít meg. Ez biztosítja, hogy a regisztráció csak megfelelően kitöltött adatokkal történhessen meg, így megelőzhetők az adatbázisba kerülő hiányos bejegyzések.

```
if($_POST['u_r_nick']== "") {  
    echo "<script> alert('Nem adtál meg felhasználónevet!') </script>";  
    print "<script> parent.location.href = './?p=login' </script>";  
}  
if($_POST['u_r_email']== "") {  
    echo"<script> alert('Nem adtál meg e-mail címet!') </script>";  
    print "<script> parent.location.href = './?p=login' </script>";  
}  
if($_POST['u_r_pwd']== "") {  
    echo"<script> alert('Nem adtál meg jelszót!') </script>";  
    print "<script> parent.location.href = './?p=login' </script>";  
}
```

**A(z) focusdog.hu közlendője**

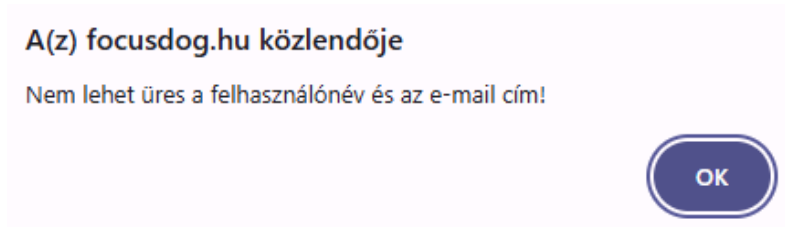
Nem adtál meg felhasználónevet!

OK

## Adatmódosítás

A feltételben a “ha nem üres” esetet adtuk meg, de ha a felhasználónév (**felhasznnev**), vagy az e-mail cím (**felhaszmail**) mező üresen marad, akkor az egy else ágban kezelve van. Szintén egy JavaScript-es felugró ablakban figyelmeztető üzenetet jelenít meg.

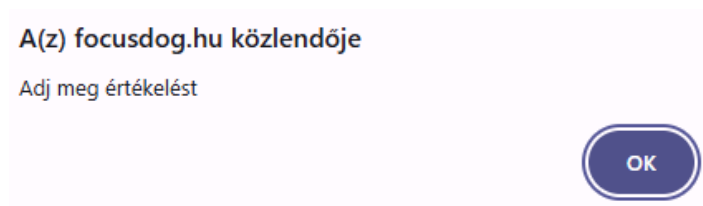
```
if(!empty($_POST['felhasznnev']) && !empty($_POST['felhaszmail'])) {...}
else{
    echo "<script>alert('Nem lehet üres a felhasználónév és az e-mail cím!');
    </script>";
    print "<script> parent.location.href = './?p=adatlapom' </script>";
}
```



## Értékelés

A feltételben ha a szöveges értékelés mező (**opinion**), és a csillagos értékelés (**rating**) mező üresen marad, akkor egy JavaScript-es felugró ablakban figyelmeztető üzenetet jelenít meg.

```
if($_POST['opinion']=="" && $_POST['rating']==""){
    echo "<script> alert('Adj meg értékelést') </script>";
}
```



## Duplikált felhasználói adatok szűrése

A weboldal tesztelése során felmerült, hogy a rendszer nem szűri ki megfelelően a már létező felhasználóneveket vagy e-mail címeket, így ezek ellenőrzésére is figyelmet fordítottunk.

### Regisztráció

A regisztrációs folyamat során ellenőriztük, hogy a lekérdezett felhasználónév (`unamequery`), és e-mail cím (`umailquery`) már szerepel-e az adatbázisban. Ha igen akkor egy JavaScript-es felugró ablakban figyelmeztető üzenetet jelenít meg.

```
if(mysqli_num_rows($unamequery) > 0){  
    echo "<script> alert('Ez a felhasználónév már létezik!') </script>";  
    print "<script> parent.location.href = './?p=login' </script>";  
}  
else if(mysqli_num_rows($umailquery) > 0){  
    echo "<script> alert('Ez az e-mail cím már létezik!') </script>";  
    print "<script> parent.location.href = './?p=login' </script>";  
}
```

**A(z) focusdog.hu közlendője**

Ez a felhasználónév már létezik!

OK

## Értékelés korlátozása

Weboldalakon gyakran előfordul, hogy a felhasználók próbálnak a rendszer által meghatározott határokon kívül eső értékeléseket megadni, például negatív vagy rendkívül magas értékeket választva. A rendszer tesztelése során ennek ellenőrzésére is figyelmet fordítottunk.

A kódrészletben a rendszer ellenőrzi, hogy az értékelés értéke nem nulla, továbbá, ha az érték kisebb, mint 0 vagy nagyobb, mint 5, akkor a rendszer egy JavaScript alapú felugró ablakban figyelmeztető üzenetet jelenít meg.

```
$ratingstar = $_POST['rating'];

if($ratingstar != 0){
    if( $ratingstar < 0 || $ratingstar > 5){
        echo "<script> alert('Az értékelés csak 1 és 5 között  
lehetséges!')
        </script>";
        print "<script> parent.location.href = './' </script>";
    }
}
```

### A(z) focusdog.hu közlendője

Az értékelés csak 1 és 5 között lehetséges!

OK

## Fejlesztési lehetőségek

A kutyaiskola weboldalának megvalósítása során célunk egy könnyen navigálható, átlátható és felhasználóbarát online platform létrehozása volt, amely nemcsak bemutatja az iskola szolgáltatásait és tanfolyamait, hanem lehetőséget biztosít a felhasználók számára, hogy gyorsan és kényelmesen regisztráljanak, jelentkezzenek a képzésekre, és véleményt írjanak az iskola szolgáltatásairól. A projekt sikeresen megvalósította a kutyaiskola alapvető funkcióit, de a jövőben számos fejlesztési lehetőség van a rendszer bővítésére.

Az alábbiakban bemutatjuk a weboldal további fejlesztési irányait, amelyek növelhetik a felhasználói élményt és a weboldal hatékonyságát.

### Mobilbarát alkalmazás fejlesztése

A weboldal mobilbarát verziója már elérhető, de egy mobilalkalmazás fejlesztése tovább növelhetné a felhasználói élményt. Az alkalmazás segítségével a felhasználók könnyedén hozzáférhetnek a kutyaiskola tanfolyamaikhoz, nyomon követhetik a kutyáik előrehaladását, és értesítéseket kaphatnak az új programokkal és ajánlatokkal kapcsolatban. A mobilalkalmazás előnye, hogy offline elérhetőséget biztosíthat, így a felhasználók bármikor és bárhol hozzáférhetnek az információkhoz.

### Webshop integrálása

A kutyaiskola weboldalának további fejlesztési lehetősége egy webshop bevezetése, amely lehetőséget biztosít a felhasználók számára, hogy kutyáik számára szükséges eszközöket, játékokat, felszereléseket és oktatási anyagokat vásároljanak. A webshop kínálata bővíthető lenne kutyákkal kapcsolatos termékekkel, mint például nyakörvek, pórázok, kutyaeledel, tréning eszközök, és kutyaiskolai oktatási anyagok. Ez a funkció nemcsak a felhasználók kényelmét szolgálná, hanem új bevételi forrást is biztosíthatna az iskola számára.

# FELHASZNÁLÓI DOKUMENTÁCIÓ

## A program célja és lényegesebb funkciói

Ez az internet kapcsolattal működő weboldal azért készült, hogy segítsen neked, mint kutyatulajdonosnak eligazodni a kutyaiskolával kapcsolatos dolgokban. Ha szeretnéd, hogy a kutyád jól viselkedjen, figyeljen rád, és boldogabb legyen a közös élet, akkor ez az oldal neked szól.

A cél az, hogy minden fontos információ egy helyen elérhető legyen, és ne kelljen külön e-maileket írogatni, telefonálni vagy papírokat töltögetni. A weboldalon keresztül egyszerűen, gyorsan és érthetően lehet intézni a kutyaiskolával kapcsolatos dolgokat.

Amikre fel tudod használni az oldalt :

**Megnézheted, milyen tanfolyamok vannak** – például kölyökkutya tanfolyam, felnőtt kutya tanfolyam, idős kutya tanfolyam, akár kezdő és haladó szinten is, illetve különböző ügyességi tanfolyamok is elérhetőek. Minden tanfolyamnál részletes leírást találsz arról, hogy mit tanul meg a kutya, mennyi ideig tart a képzés, és kiknek ajánlott.

**Egyszerűen tudsz jelentkezni** – kiválasztod a tanfolyamot, megadod az adataidat, és már kész is a jelentkezés. Rendszerünk küld neked egy visszaigazoló e-mailt is.

A weboldalt úgy alakítottuk ki, hogy akkor is tudd használni, ha nem vagy gyakorlott internethasználó. Minden egyértelműen van felépítve, a szövegek közérthetőek, és a lépések logikus sorrendben követik egymást.

Ez az oldal azért van, hogy megkönnyítse a dolgod gazdiként, és hogy a kutyaiskolai ügyintézés gyors, egyszerű és stresszmentes legyen.



## Szükséges hardvereszközök és szoftverek

**Hardver** : ez alatt a kézzel megfogható informatikai eszközt értjük, mint például egy okostelefon, vagy egy laptop, vagy akár egy asztali számítógép. A weboldalunk úgy lett kialakítva, hogy ezek közül bármelyik eszközön megnyitható, és használható.

**Szoftver** : szoftver alatt pedig a kézzel nem megfogható alkalmazásokat értjük. Csak vezetéknélküli Wifi, vagy Mobilinternet kapcsolatra lesz szükség. Természetesen egy böngészőre is szükség van, mit például a Mozilla Firefox, a Google Chrome, a Microsoft Edge, az iOS Safari böngészője, vagy a telefonos Google böngésző.

## Telepítés és indítás

Ez a kutyaiskola weboldal nem igényel semmilyen telepítést, vagyis nem kell letöltened vagy feltelepítened semmit a számítógépedre vagy a telefonodra. Az oldal egy úgynevezett webalkalmazás, amit egyszerűen a böngésződön keresztül érhetsz el, pont úgy, mint bármelyik másik internetes oldalt.

1. Nyisd meg a böngészőt – ez lehet például Google Chrome, Firefox, Safari vagy Edge.
2. Írd be a weboldal címét a címsorba – [focusdog.hu](http://focusdog.hu)
3. Nyomj Entert – és már meg is jelenik az oldal főoldala.

Innen pedig egyszerűen követheted a menüpontokat : „Kurzusaink”, „Oktatóink”, „Árlistánk”, „Kapcsolat”.

## Részletes bemutatás

Az alábbiakban lépésről lépésre bemutatjuk, milyen menüpontok vannak az oldalon, és mire használhatók.

### Főoldal

Amikor először megnyitod a weboldalt, ide érkezel.

Itt egy rövid, barátságos bemutatkozás fogad: ki vagyunk, mit csinálunk, és hogyan tud neked segíteni a kutyaiskola. Ha csak ismerkedni szeretnél az iskolával, ez a legjobb kiindulópont.

### Kurzusok

Ezt a részt szintén a főoldalon találod lejjebb görgetve. Ebben a menüpontban van az összes elérhető tanfolyam, például: Kölyök kutya kurzus kezdő és haladó szinten, Felnőtt kutya kurzus kezdő és haladó szinten, Idős kutya kurzus kezdő és haladó szinten, Agility kurzus, Őrző-védő kurzus, és még az Egyéni oktatás is. Minden tanfolyamhoz tartozik egy részletes leírás, amelyből megtudhatod hogy milyen kutyának ajánlott, hogy mit fog megtanulni a kutya, hogy mennyi ideig tart, hogy mikor indul, és hogy mennyibe kerül.

### Jelentkezés

Minden kurzus alatt van egy jelentkezem gomb. A megnyomásával megnyílik egy ablak ahol lehetőséged van kiválasztani a kurzus nevét, a tanfolyam szintjét (kezdő vagy haladó), és az oktatót. Emellett ha jelentkezelsz meg kell adnod a nevedet, az e-mail címed, és a kutyád nevét akivel részt vennél. A mezők karakter hossza vagy ékezetes betűkre nem érzékenyek, de az e-mail cím beírásakor kötelező érvényes e-mail címet megadni amiben szerepel a @. Miután elküldted a jelentkezésedet érkezik hozzád egy visszaigazoló e-mail, amelyben szintén megtalálhatsz minden információt.

### Oktatók

Szintén a főoldalon találod a kurzusok alatt az oktatóinkat, akiknek elolvashatod bemutatkozásukat, illetve a feltüntetett telefonszámon, vagy e-mail címen keresztül kapcsolatba is tudsz lépni velük.

## Információk

Az oktatóink alatt találsz az információk szakaszt, ahol egy tanfolyam előtt, vagy akár általános szükséges információkat el tudod olvasni.

## Gyakori kérdések

Ismét a főoldalon találsz a galériánk után. A leggyakoribb kérdéseket gyűjtöttük össze, amik egy gazdiban felmerülhetnek. A kérdésre kattintással lenyílik egy szöveg ami tartalmazza a választ.

### Milyen tanúsítványt kapunk a tanfolyam elvégzése után?

A tanfolyamok sikeres elvégzése után igazolást állítunk ki, amely tartalmazza a kutya és a gazdi által elsajátított készségeket.

## Értékelés

Közvetlen a gyakori kérdések alatt találsz. Ha már jártál nálunk egy tanfolyamon, vagy csak véleményed van kutyaiskolánkról, itt le tudod írni azt, sőt még csillagos értékelést is adhatsz. Ahhoz hogy értékelést tudj írni szükséges regisztrálni / bejelentkezni. A szöveges mező karakter hosszra vagy ékezetes betűkre nem érzékeny.

## Kapcsolat

Ez a főoldal legalja. Itt felvehet velünk a kapcsolatot a feltüntetett telefonszámon vagy e-mail címen, megtalálhatod az adatkezelési tájékoztatót, és az általános szerződési feltételeket is. Emellett fel van tüntetve a helyszín, és a nyitvatartási időnk is.

## Regisztráció

A navigációs sávban a "user" ikonra rákattintva megnyílik a felület ahol tudsz regisztrálni az oldalra. A mezőkben fel van tüntetve milyen adatokat kell beírni. A mezők ékezetes betűkre nem érzékenyek, de az e-mail cím beírásakor kötelező érvényes e-mail címet megadni amiben szerepel a @, illetve a jelszónak minimum 8 karakter hosszúnak kell lennie. Ha regisztráltál az oldal automatikusan átirányít és be kell jelentkezni. A regisztrációt követően a rendszerünk küld egy visszaigazoló e-mailet a sikeres regisztrációról.



## Bejelentkezés

Akár regisztráció után, vagy a már meglévő fiókodba szeretnél bejelentkezni azt ugyanott megteheted ahol a regisztrációt is. A mezőkben fel van tüntetve, hogy milyen adataid kell beírd. A mezők karakter hossza vagy ékezetes betűkre nem érzékenyek, de az e-mail cím beírásakor kötelező érvényes e-mail címet megadni amiben szerepel a @. Ha sikeresen bejelentkeztél az oldal átirányít az adatlapodra, és a kis "user" ikon kap egy pipát. Ez jelzi, hogy be vagy lépve. Ahhoz hogy be tudj lépni szükséges előtte regisztrálni.



## Adatlap / Adatmódosítás

A bejelentkezés után rögtön az adatlapodra irányít az oldal. Az itt lévő mezőkben láthatod az adataidat (felhasználónév, e-mail cím, kutya neve) amelyekkel regisztráltál. Ezeket az adatokat módosítani is tudod ha rájuk kattintasz. Alatta találsz a jelszómódosítást ahova be kell írd a régi jelszavadat, és az újat amire szeretnéd cserélni. A mezők ékezetes betűkre nem érzékenyek, de az e-mail cím beírásakor kötelező érvényes e-mail címet megadni amiben szerepel a @, illetve a jelszavaknak minimum 8 karakter hosszúnak kell lennie. A mezők alatt találsz 3 gombot. Az egyikkel a főoldalra tudsz visszalépni, a másikkal le tudod menteni a módosított adatokat, a harmadikkal pedig ki tudsz jelentkezni. Ahhoz hogy az adatlapodat meg tudd nézni szükséges regisztrálni / bejelentkezni.

## Kijelentkezés

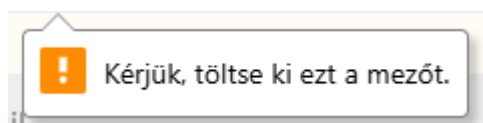
Ha be vagy lépve az említett "pipás user" ikon mellett a navigációs sávban egy kilépés gomb is megjelenik. Erre kattintva a rendszer kiléptet az oldalról. A másik lehetőség a kilépésre az adatlap alján lévő kilépés gomb, amely pirossal ki van emelve. Ahhoz hogy ki tudj lépni az oldalról előtte szükséges regisztrálnod / bejelentkezned.



# Hibajelzések

## Regisztráció

A regisztráció során meg kell adni a felhasználónevet, az e-mail címet, a jelszót, és a kutyád nevét. Ezek közül mindegyik kötelezően kitöltendő, kivéve a kutya neve. Az e-mail címnél érvényes e-mail címet kell megadni ami tartalmazza a @ karaktert. A jelszónak minimum 8 karakter hosszúnak kell lennie. Ha bármelyik mezőt üresen hagyod, két féle hiba üzenet jelenhet meg : az egyik egy “Kérjük, töltsd ki ezt a mezőt” kis hiba üzenet, a másik egy “Nem adtál meg felhasználónevet” vagy “Nem adtál meg e-mail címet” feliratú hibaüzenet. Ha olyan felhasználónevet vagy e-mail címet próbálsz megadni amivel már más regisztrált akkor szintén két hibaüzenet jelenhet meg : az egyik az “Ez a felhasználónév már létezik”, a másik az “Ez az e-mail cím már létezik” feliratú hibaüzenet.



A(z) focusdog.hu közlendője

Nem adtál meg felhasználónevet!

OK

A(z) focusdog.hu közlendője

Ez az e-mail cím már létezik!

OK

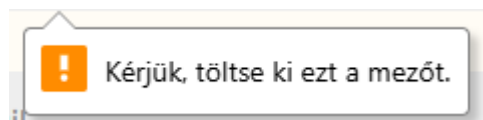
A(z) focusdog.hu közlendője

Ez a felhasználónév már létezik!

OK

## Bejelentkezés

A bejelentkezés során meg kell adni a felhasználónevet, és a jelszót. Ezek közül mindegyik kötelezően kitöltendő. A jelszónak minimum 8 karakter hosszúnak kell lennie. Ha bármelyik mezőt üresen hagyod, egy “Kérjük, töltsd ki ezt a mezőt” kis hibaüzenet jelenik meg. Ha az adataid nem egyeznek a regisztrált adataiddal, akkor egy “Hibás belépési adat(ok)! Kérjük próbálja meg újra.” feliratú hibaüzenet jelenik meg.



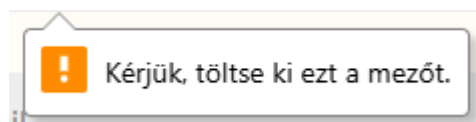
### A(z) focusdog.hu közlendője

Hibás belépési adat(ok)! Kérjük próbálja meg újra.

OK

## Tanfolyamra jelentkezés

A jelentkezés során meg kell adni nevedet, az e-mail címedet, és a kutyád nevét. Ezek közül mindegyik kötelezően kitöltendő. Az e-mail címnél érvényes e-mail címet kell megadni ami tartalmazza a @ karaktert. Ha bármelyik mezőt üresen hagyod, egy “Kérjük, töltsd ki ezt a mezőt” kis hibaüzenet jelenik meg.



## Adatmódosítás

A adatmódosítás során tudod módosítani felhasználónevedet, az e-mail címedet, a kutyád nevét, és jelszavadat. Ezek közül mindegyik kötelezően kitöltendő ha módosítod őket. Az e-mail címnél érvényes e-mail címet kell megadni ami tartalmazza a @ karaktert, és jelszónak minimum 8 karakter hosszúnak kell lennie. Ha bármelyik mezőt üresen hagyod, egy “Nem lehet üres a felhasználónév és az e-mail cím” hibaüzenet jelenik meg.

### A(z) focusdog.hu közlendője

Nem lehet üres a felhasználónév és az e-mail cím!

OK

## Értékelés

A értékelés során meg tudsz adni szöveges és csillagos értékelést is. A szöveges értékelésnek nincsen karakter hosszra vonatkozó maximuma. Ha a csillagos értékelésnél negatív számot, vagy 5-nél magasabb számot próbálsz megadni akkor megjelenik egy “Az értékelés csak 1 és 5 között lehetséges” feliratú hibaüzenet. Ha mindkét mezőt üresen hagyod, egy “Adj meg értékelést” hibaüzenet jelenik meg. Mivel az értékelést csak regisztrált és bejelentkezett felhasználó adhat, így ha nem vagy belépve egy “Kérjük lépjen be az értékelés megadásához!” feliratú hibaüzenet jelenik meg.

A(z) focusdog.hu közlendője

Kérjük lépjen be az értékelés megadásához!

OK

A(z) focusdog.hu közlendője

Az értékelés csak 1 és 5 között lehetséges!

OK

A(z) focusdog.hu közlendője

Adj meg értékelést

OK

## Adminisztrátori felhasználó lehetőségei

A kutyaiskola weboldalán nemcsak hétköznapi felhasználók, azaz érdeklődők és gazdik regisztrálhatnak, hanem van egy különleges jogosultságú felhasználói szint is: az adminisztrátor. Az adminisztrátor (vagy röviden: admin) olyan felhasználó, aki nemcsak használja a rendszert, hanem teljes hozzáféréssel rendelkezik annak háttérrendszeréhez is, vagyis ő kezeli a tanfolyamokat, a jelentkezéseket, a felhasználókat és a kapcsolatfelvételeket is. Az adminisztrátori jogosultság kizárólag azoknak jár, akik az oldal működtetésében aktívan részt vesznek.













A bejelentkezés után az admin számára az adatlapján megjelenik egy "Admin felület" feliratú gomb, ami átviszi őket az adminisztratív felületre. Az admin felület célja, hogy minél egyszerűbben és gyorsabban tudja kezelni a rendszer legfontosabb tartalmait.

Az admin felhasználó hozzáfér a regisztrált felhasználók listájához, ahol szükség esetén módosíthatja az adatokat, vagy akár új adminokat is hozzáadhat, illetve meglévő jogosultságokat visszavonhat. Ezzel szemben egy „sima” (általános) felhasználó csak a saját fiókjához fér hozzá. Ő tud regisztrálni, bejelentkezni, megtekinteni az elérhető tanfolyamokat, jelentkezni rájuk, és módosítani a saját adatait. De nem látja más felhasználók adatait, és nem fér hozzá a háttérkezelő funkciókhoz.

Az admin jogosultsággal rendelkező tesztfelhasználó :

**felhasználónév** : focus     **jelszó** : Focusdog

### Oktatók

ID	Név	Kutyanév	Email	Telefonszám	Módosít	Töröl
110	Takács Boldizsár	Zara	takacsboldizsar@focusdog.hu	99/111-2222		
220	Molnár Bernadett	Marley	molnarbernadett@focusdog.hu	99/876-5432		
330	Nagy Ákos	Colin	nagyakos@focusdog.hu	99/123-4567		
440	Tóth Kata	Luna	tothkata@focusdog.hu	99/987-6543		
550	Kovács Tünde	Roxy	kovacsTunde@focusdog.hu	99/234-5678		
660	Kiss Erik	Benő	kisserik@focusdog.hu	99/555-1234		



## Információkérés

A navigációs sávban elérhető a “Kapcsolat” menüpont, amely az oldal aljára irányít, ahol fel van tüntetve a Focus Kutyaiskola telefonszáma, és e-mail címe. Ha bármilyen kérdésed felmerülne ezeken az elérhetőségeken keresztül kapcsolatba tudsz lépni velünk, a nyitvatartási időn belül, ami közvetlen az elérhetőségek mellett található.

Ha közvetlen az oktatókkal szeretnél kapcsolatba lépni, az ő e-mail címüket és telefonszámukat az oktatók szekcióban találod, minden oktatónk alatt a sajátját. Emellett a navigációs bárból is eléred az oktatók menüpontot.

Ha pedig személyesen szeretnél érdeklődni, az Óbudai sziget csúcsán, a Május 9. parkban megtalálasz minket, szintén nyitvatartási időben.

A Focus Kutyaiskola telefonszáma : +36 99 999 9999

A Focus Kutyaiskola e-mail címe : [info@focusdog.hu](mailto:info@focusdog.hu)

A Focus Kutyaiskola helyszíne : 1033, Május 9. park (Óbudai sziget csúcsa)

Oktatóink és elérhetőségeik :

Nagy Ákos	: <a href="mailto:nagyakos@focusdog.hu">nagyakos@focusdog.hu</a>	+36 99 123 4567
Tóth Kata	: <a href="mailto:tothkata@focusdog.hu">tothkata@focusdog.hu</a>	+36 99 987 6543
Kiss Erik	: <a href="mailto:kisserik@focusdog.hu">kisserik@focusdog.hu</a>	+36 99 555 1234
Kovács Tünde	: <a href="mailto:kovacstunde@focusdog.hu">kovacstunde@focusdog.hu</a>	+36 99 234 5678
Molnár Bernadett	: <a href="mailto:molnarbernadett@focusdog.hu">molnarbernadett@focusdog.hu</a>	+36 99 876 5432
Takács Boldizsár	: <a href="mailto:takacsboldizsar@focusdog.hu">takacsboldizsar@focusdog.hu</a>	+36 99 111 2222

# IRODALOMJEGYZÉK

Szakmai könyvek amelyek segítettek a weboldalunk struktúrájának, és interaktivitásának kialakításában :

Richard Blum - PHP, MySql & Javascript 7 könyv 1-ben

Virginia DeBolt - HTML és CSS - Webszerkesztés stílusosan

Weboldalak amik szintén sokban hozzájárultak projektünk elkészítéséhez :

<https://www.w3schools.com/> - több nyelvben is hasznát vettük

<https://infojegyzet.hu/webszerkesztes/php/> - legjobb segítségünk a php nyelvben

<https://infojegyzet.hu/webszerkesztes/html/> - a dinamikus elemek elkészítése

<https://infojegyzet.hu/webszerkesztes/dokumentacio/> - a záródolgozat szempontjai

<https://hu.pinterest.com/> - weboldal ötletek, képek a galériához

# ÖSSZEGZÉS

A projektünk célja egy modern, felhasználóbarát kutyaiskola weboldal megtervezése és megvalósítása volt. A projekt során törekedtünk arra, hogy mind a felhasználói, mind az adminisztratív funkciók könnyen használhatók legyenek. A kezdeti célkitűzéseinket – miszerint egy interaktív és esztétikus felületet hozunk létre, ahol a látogatók jelentkezhetnek tanfolyamokra, értékelhetnek, kapcsolatba léphetnek az iskolával – sikerült megvalósítani.

A legnagyobb kihívást a rendszer komplexitása jelentette: több adatbázistábla és összetett kapcsolatok kezelésére volt szükség, amelyek logikáját precízen kellett megtervezni. Emellett a jogosultságkezelés és az adminfelület kialakítása is összetett feladat volt, de sokat tanultunk belőle, különösen a backend fejlesztés és adatbázis-kezelés terén.

A projekt során sokat fejlődhattunk szakmailag, jobban megérthettük a Backend fejlesztés működését, és gyakorlatot szereztünk abban, hogyan lehet egy valós igényre szabott, működőképes weboldalt készíteni.

A projekt különlegessége abban rejlett, hogy nem csupán technikai megoldásokat kellett alkalmaznunk, hanem a felhasználói élményt is szem előtt kellett tartanunk. Nagy hangsúlyt fektettünk arra, hogy az oldal vizuálisan is vonzó legyen, hiszen egy kutyaiskola célközönsége számára a barátságos megjelenés és a könnyű kezelhetőség alapvető elvárás. A reszponzív design révén a weboldal mobil eszközökön is kényelmesen használható, így a felhasználók bármikor és bárhol elérhetik a szolgáltatásokat.

A weboldal jövőjét illetően úgy látjuk, hogy a további fejlesztések segíthetnék abban, hogy a program hosszú távon is használható és fejleszthető maradjon.

Összességében büszkék vagyunk arra, amit létrehoztunk, és úgy érezzük, hogy ez a projekt valódi értéket képviselhet a kutyaiskolák és a gazdák számára.

# KÖSZÖNETNYILVÁNÍTÁS

Ezúton szeretnénk megköszönni mindazoknak, akik bármilyen formában hozzájárultak a vizsgaremekünk elkészítéséhez.

Először is hálásak vagyunk Kovács László tanár úrnak a szakmai iránymutatásért, támogatásért és türelemért, amivel végigkíserte a munkánkat. Észrevételei és javaslatai nélkül a projekt nem lehetett volna ilyen alapos és jól strukturált.

Külön szeretném megköszönni valamennyi szakmai tanárunknak a tanulmányaink során nyújtott folyamatos támogatást és elkötelezett munkájukat. Az általuk közvetített naprakész ismeretek és a rendszerszintű gondolkodásmód kialakításához nyújtott segítség elengedhetetlen alapot biztosított számunkra ahhoz, hogy a projektünket meg tudjuk valósítani. Az órákon átadott szakmai tudás és az építő jellegű visszajelzések nem csupán a vizsgaremek elkészítéséhez járultak hozzá, hanem hosszú távon is meghatározóak lesznek a szakmai fejlődésünk szempontjából.

Végül, de nem utolsósorban köszönettel tartozunk a családjainknak, akik támogatásukkal segítették tanulmányainkat. Megértésük és türelmük sokat jelentett. A barátainknak is hálásak vagyunk a folyamatos biztatásért és az építő visszajelzésekért, amikor tesztelték az oldalt vagy véleményt mondtak a terveinkről.

Ez a projekt számunkra nem csupán egy kötelező feladat, hanem egy tanulási folyamat és szakmai mérföldkő volt, melyért minden érintettnek köszönettel tartozunk.