WEBPROGRAMOZÁS

Cím: Kutya sétáltatása

|  |  |
| --- | --- |
| Készítette: Pécsi Erik  Leckekönyv: 26221008  Szak: Informatika | Tanár: Zlatko Čović |

Tartalomjegyzék

[Projektleírás 3](#_Toc88554483)

[Adatbázis 4](#_Toc88554484)

[Csatlakozás az adatbázishoz 5](#_Toc88554485)

[Auth session 6](#_Toc88554486)

[Adatbevitel 7](#_Toc88554487)

[SQL-lekérdezések 8](#_Toc88554488)

[AJAX, JavaScript 9](#_Toc88554489)

[CSS 10](#_Toc88554490)

# Projektleírás

A **Kutyasétáltatás** egy weboldal, amely lehetőséget kínál a kutyatulajdonosoknak és kutyasétáltatóknak egyaránt. Az oldal célja, hogy összekapcsolja a kutyák tulajdonosait olyan lelkes kutyasétáltatókkal, akik szívesen gondoskodnak a kutyusok rendszeres sétáltatásáról és ellátásáról.

A weboldal további hasznos információkat is kínál, például az aktív sétáltatók értékelését és visszajelzéseit, így a kutyatulajdonosok megbizonyosodhatnak róla, hogy megbízható szolgáltatást kapnak. Emellett lehetőség van a sétáltatók profiljainak böngészésére, ahol részletes leírások és elérhetőségek találhatók.

# Adatbázis

Az adatbázis a Kutyasétáltatás weboldal működésének alapját képezi, ahol tárolódnak és kezelésre kerülnek a felhasználói adatok, sétáltatási kérelmek, kutyákkal kapcsolatos információk és egyéb releváns adatok. Az adatbázis lehetővé teszi a weboldal számára, hogy hatékonyan kezelje és rendszerezze az adatokat, és megfelelően válaszoljon a felhasználók kéréseire. Az adatbázis lehetővé teszi a felhasználók regisztrációját, bejelentkezését, sétáltatási kérelmek feladását és kezelését, sétáltatók értékelését és a kapott visszajelzéseket. Az adatok hatékony kezelésével a weboldal lehetővé teszi a felhasználók számára, hogy megbízható és kényelmes kutyasétáltatási szolgáltatást találjanak vagy nyújtsanak.

# Csatlakozás az adatbázishoz

<?php  
  
$host = "host";  
$username = "username";  
$password = "password";  
$database = "database";  
  
$con = mysqli\_connect($host, $username, $password, $database);  
  
  
if (mysqli\_connect\_errno()) {  
 echo "Failed to connect to MySQL: " . mysqli\_connect\_error();  
}  
?>

Először is definiálódnak a kapcsolódáshoz szükséges változók:

- $host: Az adatbázis szerver neve vagy IP címe.

- $username: A felhasználónév, amellyel az adatbázishoz kapcsolódni szeretnénk.

- $password: A jelszó a megadott felhasználónévhez.

- $database: Az adatbázis neve, amelyhez kapcsolódni szeretnénk.

Ezt követően a mysqli\_connect() függvény segítségével próbálunk kapcsolódni az adatbázishoz a fent megadott információk alapján. A mysqli\_connect() függvény visszaad egy kapcsolatot az adatbázissal, vagy false értéket, ha a kapcsolódás sikertelen.

Ezt követően a mysqli\_connect() függvény segítségével próbálunk kapcsolódni az adatbázishoz a fent megadott információk alapján. A mysqli\_connect() függvény visszaad egy kapcsolatot az adatbázissal, vagy false értéket, ha a kapcsolódás sikertelen.

# Auth session

Csak bejelentkezett felhasználók:

<?php  
session\_start();  
if(!isset($\_SESSION["username"])) {  
 header("Location: login.php");  
 exit();  
}  
?>

Vendégek is engedélyezve:

<?php  
session\_start();  
if(!isset($\_SESSION["username"]) || $\_SESSION["username"] == "Vendég") {  
 header("Location: login.php");  
 exit();  
}  
?>

Az if utasítás vizsgálja meg, hogy van-e értéke a $\_SESSION["username"] változónak. A $\_SESSION munkamenet változóit tartalmazza. Ebben az esetben az ellenőrzés arra irányul, hogy a "username" kulcsú változó létezik-e a munkamenetben. A második esetben is ez történik, annyi kivétellel, hogy a “Vendég” is engedélyezve van az oldalon. Vendégként a bejelentkező felületen található gombbal tudunk belépni.

# Adatbevitel

Az oldalon lehetőségünk van adatokat bevinni az adatbázisba. Egy példa erről:

if (!$walkData['rated']) {  
  
 $updateRatedQuery = "UPDATE walks SET rated = 1 WHERE id = '$walk\_id'";  
 mysqli\_query($con, $updateRatedQuery);  
  
  
 $updateRatingQuery = "UPDATE users SET rating = '$rating', rates = rates + 1 WHERE id = '$walker\_id'";  
 mysqli\_query($con, $updateRatingQuery);  
  
 echo "Sikeresen értékelted a sétáltatást!";  
} else {  
 echo "Ezt a sétáltatást már értékelted korábban.";  
}

Az if utasítás ellenőrzi, hogy a $walkData tömbben lévő 'rated' kulcs értéke false vagy sem. Ez arra utal, hogy a sétáltatás még nincs értékelve. A $walkData tömb valószínűleg előzőleg be lett töltve adatokkal, például az adott sétáltatásról és az értékelésről.

Ha a sétáltatás még nincs értékelve ($walkData['rated'] értéke false), akkor a következő lépések történnek:

Az $updateRatedQuery változóban egy SQL lekérdezés definiálódik, amely az adatbázisban beállítja a walks táblában a rated oszlopot 1-re (azaz értékeltre) az adott id alapján ($walk\_id).

A mysqli\_query() függvény segítségével végrehajtjuk az előző lekérdezést a kapcsolaton keresztül ($con).

Az $updateRatingQuery változóban egy másik SQL lekérdezés van, ami a users táblában frissíti a felhasználó értékelési adatait. A rating oszlopot frissítjük az új értékeléssel ($rating), és növeljük a rates oszlop értékét 1-el az adott felhasználó ($walker\_id) alapján.

A mysqli\_query() függvény segítségével végrehajtjuk az előző lekérdezést a kapcsolaton keresztül.

# SQL-lekérdezések

Az oldal több helyen is használ SQL lekérdezést, adatok kiírására (például: Kutyasétáltatók). Egy példa erről:

$userQuery = "SELECT id, username, email, create\_datetime, adminlevel, banned FROM users";  
$userResult = mysqli\_query($con, $userQuery);  
  
while ($row = mysqli\_fetch\_assoc($userResult)) {  
 echo "<tr>";  
 echo "<td>" . $row['id'] . "</td>";  
 echo "<td>" . $row['username'] . "</td>";  
 echo "<td>" . $row['email'] . "</td>";  
 echo "<td>" . $row['create\_datetime'] . "</td>";  
 echo "<td>" . $row['adminlevel'] . "</td>";  
 echo "<td>" . $row['banned'] . "</td>";  
 echo "</tr>";  
}

Először definiálódik a $userQuery változóban egy SQL lekérdezés, amely kiválasztja az "users" táblából az alábbi oszlopokat: 'id', 'username', 'email', 'create\_datetime', 'adminlevel', 'banned'. Az SQL lekérdezés a teljes "users" táblát lekéri.

A mysqli\_query() függvény segítségével végrehajtjuk az előző lekérdezést a kapcsolaton keresztül ($con). Az eredményt a $userResult változóban tároljuk.

Ezt követően egy while ciklus indul, amely végigiterál az $userResult eredményhalmazon, amíg van még sor az eredmények között. A mysqli\_fetch\_assoc() függvény az eredményhalmaz következő sorát tölti be egy asszociatív tömbbe ($row változóba).

# AJAX, JavaScript

$(***document***).ready(function() {  
 $.ajax({  
 url: 'get\_walkers.php',  
 type: 'GET',  
 dataType: 'json',  
 contentType: 'application/json; charset=utf-8',  
 success: function(data) {  
 var walkerList = $('#walker-list');  
 walkerList.empty();  
  
 for (var i = 0; i < data.length; i++) {  
 var walker = data[i];  
 walkerList.append('<div class="walker">' +  
 '<h2>' + walker.username + '</h2>' +  
 '<p>Átlagos értékelés: ' + walker.averageRating + '</p>' +  
 '<p>Leírás: ' + walker.description + '</p>' +  
 '</div>');  
 }  
 },  
 error: function() {  
 ***console***.log('Hiba történt az AJAX kérés során.');  
 }  
 });  
});

A $(document).ready(function() {...}); blokk azt jelenti, hogy a kód akkor fut le, amikor az egész dokumentum (HTML oldal) betöltődött és készen áll a JavaScript végrehajtására.

A success: function(data) {...} blokkban a siker esetén végrehajtandó kód van:

Először megtisztítjuk (empty()) a walkerList elem tartalmát, hogy a friss adatokat tudjuk hozzáadni.

Egy cikluson keresztül végigmegyünk a data tömbön, amelyet a válasz JSON adatok tartalmazzák.

Minden walker objektumon belül (a tömb egyes elemei) az alábbi HTML elemeket hozzáadjuk a walkerList elemhez:

<div class="walker">: Egy új "walker" div elem kezdete.

<h2>: Sétáltató felhasználóneve.

<p>: Átlagos értékelés.

<p>: Sétáltató leírása.

'</div>': A "walker" div elem lezárása.

# CSS

A Cascading Style Sheets (CSS) egy stíluslap nyelv, amelyet a weboldalak kinézetének és elrendezésének irányítására használnak. A CSS segítségével meg lehet határozni, hogy az elemek hol helyezkedjenek el a weboldalon. A CSS segítségével animációkat és átmeneteket lehet létrehozni. Ezek az animációk lehetnek például gombnyomásra történő mozgások vagy színes változások. A CSS segítségével a weboldalak mérete csökkenthető, ami gyorsabb betöltést eredményezhet. A kódot összevonhatjuk, felesleges ismétlődő szabályokat eliminálhatjuk, és minimálisra csökkenthetjük a képek és egyéb erőforrások mennyiségét. CSS segítségével az oldalt reszponzívvá tehetjük, ami azt eredményezi, hogy minden készülékről könnyen olvasható lesz az oldalunk.

@media (max-width: 768px) {  
 .navbar {  
 width: 100%;  
 position: static;  
 justify-content: center;  
 align-items: center;  
 }  
  
 .content {  
 margin-left: 0;  
 }  
}

Ez a CSS kód egy media lekérdezést definiál, amely a képernyő szélességét vizsgálja. Ha a képernyő szélessége 768 pixeltől kisebb vagy egyenlő, akkor az itt megadott stílusok alkalmazódnak. A media lekérdezések lehetővé teszik a stílusok testreszabását és alkalmazkodását különböző kijelzőméretekhez és eszközökön.